OSCAR and ActivityNet: an Image Captioning model can effectively learn a Video Captioning dataset

Emmanuel Byrd*
Tecnologico de Monterrey
México
a01166339@itesm.mx

Miguel Gonzalez-Mendoza
Tecnologico de Monterrey
México
mgonza@tec.mx

Leonardo Chang
Tecnologico de Monterrey
México
lchang@tec.mx

Abstract

Activity Recognition and Classification in video sequences is an area of research that has received attention recently. However, video processing is computationally expensive, and its advances have not been as extraordinary compared to those of Image Captioning. This work, created by Latinx individuals from Mexico, uses a computationally limited environment and transforms the Video Captioning dataset of ActivityNet into an Image Captioning. Generating features with Bottom-Up attention and training an OSCAR Image Captioning model, and using different NLP Data Augmentation techniques, we show a viable and promising approach to simplify the Video Captioning task.

1. Introduction

Activity Recognition and Classification in video sequences is a Vision and Language task (V+L) that has caught attention in the past years. Its applications are vast and range from security surveillance [2] to Visual Question Answering (VQA).

Video processing is computationally expensive, and the search for more efficient methods and models requires new ideas. The available Video datasets are not as vast as those of Image Captioning, and working with them requires the researcher to have significant computational resources at hand. The present work explores an approach to solving these problems: eliminate the temporal dimension and attempt to tackle the Video Captioning task as an Image Captioning one.

Recent projects on Image Captioning [1, 12] have made massive progress towards faster training times [26], more accurate features, and pre-trained models that can be fine-tuned for specific V+L tasks. Transforming the ActivityNet [6] video dataset to an Image Captioning dataset, generating image features with Bottom-Up attention [1] and training an Image Captioning model using *OSCAR* [12], we show that it is possible to generate accurate descriptions from single frames of the videos. We experiment with NLP data augmentation techniques [16] to increase the model's generalization capabilities.

Working entirely in the environment of *Google Colab Pro*, this work shows different experiments made, the execution time they took, and promising results for video captioning following this approach.

2. Related Background

Previous studies have explored pre-training models on vision-language tasks with large datasets of image-text pairs, learning generic representations that can later be fine-tuned for specific tasks [1, 10, 11, 15, 3, 24, 25, 28].

Specifically, this work is based on Bottom-Up attention [1], and OSCAR [12], two contemporary architectures and models with high results on different V+L tasks. Anderson's *et al*. Bottom-Up and Top-Down model won first place in the 2017 VQA Challenge with *70.3%* overall accuracy [1]. Furthermore, OSCAR created a new SoTA of six vision-language understanding and generation tasks, with its different finetuned models [12].

2.1. Bottom-Up attention

Bottom-Up is the encoder phase of Anderson's *et al*. model. It generates the features that can later be used on different V+L tasks, defined in bounding boxes, class tags, probability scores, and the relationship between objects. It was pre-trained first by initializing *Faster R-CNN* [20] with *ResNet-101* [7] pre-trained for classification on ImageNet [23], then trained on Visual Genome [9] data. Top-Down, the decoder phase, was trained first with Cross-Entropy Loss and then with CIDEr optimization using Self-Critical Sequence Training (SCST) [21]. This work uses the features generated by the Bottom-Up encoder.

*We thank the support of CONACYT in the development of this project.

2.2. OSCAR

OSCAR: ***O**bject-**S**emantic**s** **A**ligned **Pr**e-training* is a new cross-modal pre-training method. It leverages anchor points to facilitate the learning of image-text alignments. The essence of OSCAR is motivated by observing that salient objects in an image can be accurately detected, and are often mentioned in its annotations (sentences). For example, on the MS COCO dataset [14], the percentages that an image and its paired text share at least 1, 2, 3 objects are 49.7%, 22.2% and 12.9% respectively.

OSCAR is an enormous model, pre-trained with 6.5 million text-image pairs gathered from different public V+L datasets [8, 17, 14, 22, 27], and further finetuned for IC with COCO. As will be shown further in this work, OSCAR quickly overfits our relatively small dataset. Different techniques are required for the model to learn and generalize properly.

3. ActivityNet Video Dataset

ActivityNet: A Large-Scale Video Benchmark for Human Activity Understanding [6] is a dataset built for video classification, trimmed activity classification and activity detection. It covers a wide range of complex human activities. The videos are mainly between 5 and 10 minutes long and contain tags for the activity class.

The *ActivityNet-Captions* contains 20k videos amounting to 849 video hours with 100k full descriptions, each with its unique start and end time. On average, each of the 20k videos in ActivityNet contains 3.65 temporally localized sentences [6].

To encourage participation in the *ActivityNet Dense Captioning Events in Videos challenge*[1], the creators of ActivityNet team up with other researchers [7, 5, 19] to make available other resources: RBG frames extracted at 5FPS (200GB) and frame-level features for them (89GB). The frames for all videos were extracted with FFmpeg at 5 FPS and uniformly scaled to 320x240. We make use of these extracted and scaled frames as a representation of ActivityNet, but we extracted the features using Anderson's *et al.* Bottom-Up mechanism [1].

3.1. ActivityNet transformation into Image Captioning

The starting point of the used ActivityNet dataset [7, 5, 19], although represented by frames instead of video, is still a Video Captioning dataset. It was transformed into an Image Captioning dataset by pairing each caption with the frame exactly in the middle between its starting and ending time-lapse. The splits provided by [7, 5, 19] consist of a training set with 10k captions and two validation sets with almost 5k captions each.

3.2. Feature extraction

We generated the features using Anderson's *et al.* Bottom-Up attention[1] trained model with the configuration provided in the public GitHub project's repository[2]. Google Colab Pro was the working environment, with an allocated *Tesla P100-PCIE-16GB* or *Tesla V100-SXM2-16GB* GPU and a high-RAM runtime. The pre-trained model corresponds to the Faster R-CNN, ResNet-101, and end2end combination.

The two validation sets provided by [7, 5, 19] are set aside for the final testing phase. After generating features and data cleaning, 88 captions were eliminated from the original 37421 in the training set and split into the final sizes of *mini_train* and *mini_val* with 31733 and 5600 image-sentence pairs each, following an 85%-15% distribution. The datasets used throughout this work are *mini_train* and *mini_val*.

4. Methodology

The final objective is to train an IC model to generate captions for the ActivityNet dataset that can later be fused to describe each video in a story-like fashion. We executed different experiments in the OSCAR training phase.

4.1. Training OSCAR

OSCAR parameters were initialized with the released checkpoint trained from $BERT_{BASE}$ and further trained for 30 epochs on COCO[3]. We trained this checkpoint with the generated features of ActivityNet, with a training batch size of 16 and an eval bach size of 64 (larger batch sizes caused a memory crash in Google Colab Pro), saving and evaluating the model every 5 epochs. The environment used was the same as when extracting the features: Google Colab Pro with an allocated *Tesla P100-PCIE-16GB* or *Tesla V100-SXM2-16GB* as GPU hardware acceleration and a high-RAM runtime.

4.2. NLP Data Augmentation

Different techniques are available to augment sentences [16]. The selected methods include **Back Translation** (translating to a different language and back), **Random Word Insertion** and **Random Word Substitution** using BERT [4] as a language model to select only contextually correct words.

One dataset was created using Back Translation, generating one sentence through German and another through

[1] http://activity-net.org/challenges/2020/tasks/anet_captioning.html

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

[3] https://github.com/microsoft/Oscar

Russian for every human annotation (**3x** the original size). Moreover, we created a different dataset with four new sentences generated by Random Insertion and four more with Random Substitution, both using BERT (**9x** the original size). Back Translation ensured fluency and consistency in the new sentences, while Random Insertion and Substitution not always generated natural sentences. We applied data augmentation only in the training processes; we used the original datasets to obtain both the training and evaluation scores.

5. Results

All graphs and images can be found in Section A: Supplementary Material. The metrics include BLEU-1, BLEU-2, BLEU-3, BLEU-4 (**B4**), ROUGE_L (**R**), CIDEr (**C**) and SPICE (**S**). The BLEU scores have similar values, and for illustration purposes we show a condensed version of only BLEU-4.

We trained the pristine dataset for 200 epochs, the Back Translation dataset for 40 epochs, and the Random Insertion and Substitution dataset for 50 epochs. As shown in the score graphs, datasets with augmented data require less epochs to reach similar overfitting scores, as they contain more data than the original. A more fair comparison can be made using the **global optimization steps**, which correspond to the number of times the model's parameters were updated. In the score graphs, epoch 0 corresponds to the initialization model without any training. The required time to train a single epoch of the pristine dataset, the 3x augmented dataset, and the 9x augmented dataset were: 7 minutes (23.3 hours total), 25 minutes (16.6 hours total), and 47 minutes (39.1 hours total), respectively. Evaluating a single model with *mini_val* and *mini_train* took 20 and 150 minutes respectively. The total time used to evaluate *mini_val* across all experiments was of 19.6 hours, while the time invested to evaluate *mini_train* was of 6.1 *days*.

The training scores of the pristine dataset are shown in Fig 1. All metrics follow a very similar pattern. These different metrics following a very similar pattern allow us to simplify the general behavior showing only a single metric (SPICE) for illustration purposes. The SPICE training and evaluation scores of this dataset are shown in Fig 2, which starts overfitting after epoch 80 (global step 158), with a training score of 55.24 and a validation score of 8.32.

The model is learning correctly, but the metrics do not show an appropriate generalization (validation scores), where all validation scores start at a SPICE score of 2.11, then rise to an average of The training and evaluation results of the **3x Back Translation** dataset are shown in Fig 3. Applying these NLP Data Augmentation techniques allowed the model to overfit with more time but fewer epochs, achieving in epoch 40 what the pristine dataset achieved until epoch 70; however, the validation scores remained similarly low.

The training and evaluation results of the **9x Random Insertion and Substitution** dataset are available in Fig 4. The training curve of this experiment is similar to that of the pristine dataset but a little smoother. However, it needed much more Optimization Steps to obtain similar training scores, meaning that the usage of this configuration is slowing the learning process. An example of a single training frame with its ground truth, multiple data augmented sentences, and its prediction is available in Fig 6.

6. Discussion

Although the evaluation scores do not show a high correlation between predicted output and reference captions, we can see that the generated captions accurately portray the visual information in Fig 5. The same effect occurs even through different checkpoints, as in Fig 7.

More experiments are needed to obtain a model that both learns and generalizes correctly. One of the reasons why the evaluation scores are kept low may be tightly related to the fact that there is only one reference caption per image, where other IC datasets contain five or more [14], and evaluation metrics can make use of these multiple references [13, 18].

More experiments will be done by increasing the model's dropout value and initializing with other models (*e.g.* without pre-training on COCO). The OSCAR dropout value used in these experiments was the default of 0.1. Increasing it could result in better generalization by reducing the model complexity.

Current experiments do not take advantage of the activity's class information. Further work will also include transforming the embedded representation generated by Bottom-Up attention to another one with the same size but trained using the tag of the activity class to which the frame belongs. Feeding this new representation into OSCAR will allow it to use the information of the class, which we have not exploited yet. Another approach to using the Video class is to add a new bounding box encompassing the entire frame, tagged with the activity's class. Once we achieve a stable model, we can further finetune it for the CIDEr metric using Self-Critical Sequence Training (SCST) [21].

Overfitting is the first stage towards a robust model. This work shows that an Image Captioning model can also process data created for Video Captioning. We also explore different methods to tackle the problems that this transformation creates. We are confident that this exploration will be helpful when creating custom Image or Video captioning datasets, providing users with tools and techniques to understand better how to create a functional model and the possible solutions to challenges they may face.

References

[1] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, and L. Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *CVPR*, 2018. 1, 2

[2] J. Chen, K. Li, Q. Deng, K. Li, and P. S. Yu. Distributed deep learning model for intelligent video surveillance systems with edge computing. *IEEE Transactions on Industrial Informatics*, pages 1–1, 2019. 1

[3] Y.-C. Chen, L. Li, L. Yu, A. El Kholy, F. Ahmed, Z. Gan, Y. Cheng, and J. J. Liu. Uniter: Universal image-text representation learning. In *16th European Conference Computer Vision (ECCV 2020)*, August 2020. 1

[4] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019. 2

[5] V. Escorcia, M. Soldan, J. Sivic, B. Ghanem, and B. Russell. Temporal localization of moments in video collections with natural language, 2019. 2

[6] B. G. Fabian Caba Heilbron, Victor Escorcia and J. C. Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 961–970, 2015. 1, 2

[7] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 1, 2

[8] D. A. Hudson and C. D. Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6693–6702, 2019. 2

[9] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L. Li, D. A. Shamma, M. S. Bernstein, and F. Li. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *CoRR*, abs/1602.07332, 2016. 1

[10] G. Li, N. Duan, Y. Fang, M. Gong, D. Jiang, and M. Zhou. Unicoder-vl: A universal encoder for vision and language by cross-modal pre-training, 2019. 1

[11] L. H. Li, M. Yatskar, D. Yin, C.-J. Hsieh, and K.-W. Chang. Visualbert: A simple and performant baseline for vision and language, 2019. 1

[12] X. Li, X. Yin, C. Li, X. Hu, P. Zhang, L. Zhang, L. Wang, H. Hu, L. Dong, F. Wei, Y. Choi, and J. Gao. Oscar: Object-semantics aligned pre-training for vision-language tasks. *ECCV 2020*, 2020. 1

[13] C.-Y. Lin. Rouge: a package for automatic evaluation of summaries. In *Workshop on Text Summarization Branches Out, Post-Conference Workshop of ACL 2004, Barcelona, Spain*, pages 74–81, July 2004. 3

[14] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar, and L. Zitnick. Microsoft coco: Common objects in context. In *ECCV*. European Conference on Computer Vision, September 2014. 2, 3

[15] J. Lu, D. Batra, D. Parikh, and S. Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks, 2019. 1

[16] E. Ma. Nlp augmentation. https://github.com/makcedward/nlpaug, 2019. 1, 2

[17] V. Ordonez, G. Kulkarni, and T. Berg. Im2text: Describing images using 1 million captioned photographs. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc., 2011. 2

[18] K. Papineni, S. Roukos, T. Ward, and W.-J. 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, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics. 3

[19] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in pytorch. In *NIPS-W*, 2017. 2

[20] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1137–1149, 2017. 1

[21] S. J. Rennie, E. Marcheret, Y. Mroueh, J. Ross, and V. Goel. Self-critical sequence training for image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 1, 3

[22] S. J. Rennie, E. Marcheret, Y. Mroueh, J. Ross, and V. Goel. Self-critical sequence training for image captioning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1179–1195, 2017. 2

[23] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. 1

[24] W. Su, X. Zhu, Y. Cao, B. Li, L. Lu, F. Wei, and J. Dai. Vl-bert: Pre-training of generic visual-linguistic representations, 2020. 1

[25] H. Tan and M. Bansal. LXMERT: Learning Cross-Modality Encoder Representations from Transformers. *arXiv e-prints*, page arXiv:1908.07490, Aug. 2019. 1

[26] Y. You, J. Li, S. Reddi, J. Hseu, S. Kumar, S. Bhojanapalli, X. Song, J. Demmel, K. Keutzer, and C.-J. Hsieh. Large batch optimization for deep learning: Training bert in 76 minutes, 2020. 1

[27] 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*, 2:67–78, 02 2014. 2

[28] L. Zhou, H. Palangi, L. Zhang, H. Hu, J. J. Corso, and J. Gao. Unified vision-language pre-training for image captioning and vqa. ArXiv, September 2019. 1

A. Supplementary Material

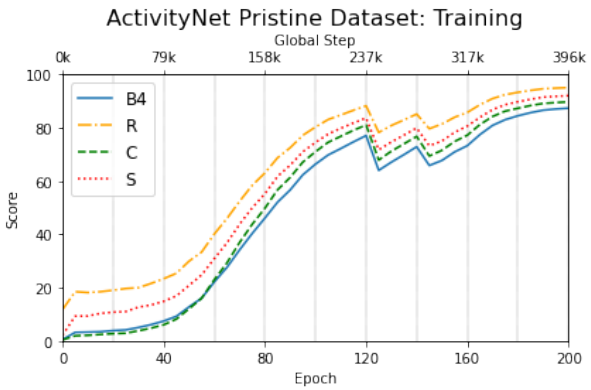

Figure 1. Training and validation scores of OSCAR-ActivityNet

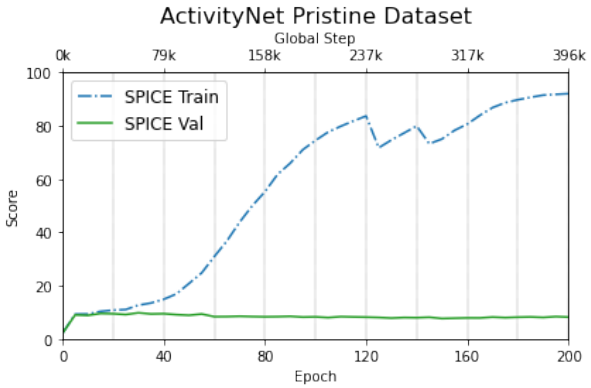

Figure 2. SPICE scores for training and validation without Data Augmentation

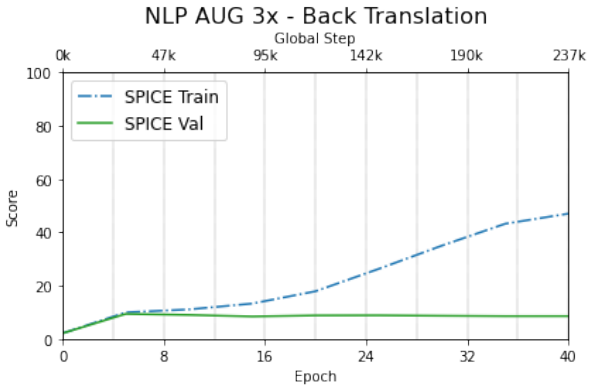

Figure 3. SPICE scores with NLP 3-fold Data Augmentation using Back Translation

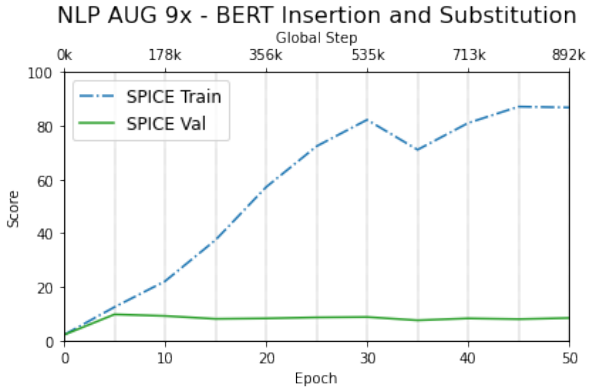

Figure 4. SPICE scores with NLP 9-fold Data Augmentation using BERT Insertion and Substitution

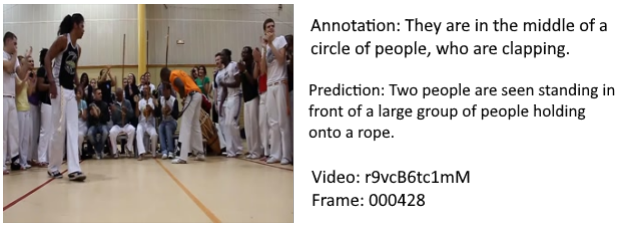

Figure 5. Example of a caption prediction from the validation set

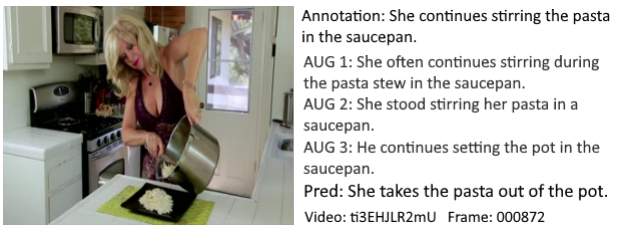

Figure 6. Example of a caption prediction from the training set with some augmented sentences.

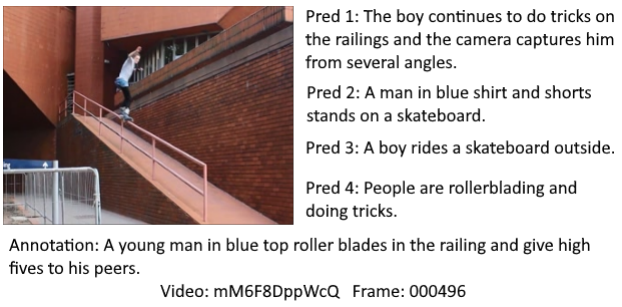

Figure 7. Different predictions for a validation frame. Note how even though the sentences are not similar, they describe the image correctly.