Mining for meaning: from vision to language through multiple networks consensus

Iulia Duţă*[1][2]
iduta@bitdefender.com
Andrei Liviu Nicolicioiu*[1][3]
anicolicioiu@bitdefender.com
Simion-Vlad Bogolin*[4]
vladbogolin@gmail.com
Marius Leordeanu[3][4]
marius.leordeanu@imar.ro

[1] Bitdefender, Romania
[2] University of Bucharest, Romania
[3] University Politehnica of Bucharest, Romania
[4] Institute of Mathematics of the Romanian Academy

Abstract

Describing visual data into natural language is a very challenging task, at the intersection of computer vision, natural language processing and machine learning. Language goes well beyond the description of physical objects and their interactions and can convey the same abstract idea in many ways. It is both about content at the highest semantic level as well as about fluent form. Here we propose an approach to describe videos in natural language by reaching a consensus among multiple encoder-decoder networks. Finding such a consensual linguistic description, which shares common properties with a larger group, has a better chance to convey the correct meaning. We propose and train several network architectures and use different types of image, audio and video features. Each model produces its own description of the input video and the best one is chosen through an efficient, two-phase consensus process. We demonstrate the strength of our approach by obtaining state of the art results on the challenging MSR-VTT dataset.

1 Introduction

The task of describing videos into natural language is one of the most exciting and still unsolved problems in artificial intelligence today. Solving this task would help decode many important questions about how the mind works, how we perceive the world, how we think and then communicate to one another. Efficient methods for vision to language translation would also have an immense practical value, with applications in many areas ranging from technology to medicine and entertainment.

The problem is hard to formulate in the traditional supervised machine learning paradigm. For every video sequence, there is, in principle, an infinite number of correct descriptions in natural language. Many leading cognitive scientists, such as Noam Chomsky [7] and Steven Pinker [25] among others, observed that every human utterance is unique. Thus, it is not reasonable to enforce an exact rigid form on a video description in natural language. Vision and language are deeply linked and evolve naturally during early age [25]. Given sufficient

*The first three authors contributed equally.

training data, one could expect an end-to-end deep learning approach to be able to translate vision into language. However, there is a lot of work to do in extracting meaning from language and being able to evaluate linguistic descriptions based on both meaning and form.

In this paper, we present an approach to address these challenges based on finding the consensual linguistic description among multiple vision to language translation models. While each model individually is able to generate well formed sentences that generally obey grammatical rules, it is the consensus among many models that best captures the hidden meaningful content and significantly outperforms the individual models on the tested evaluation metrics.

Related work. Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) [14], which are successful on text generation, are the basis for current models in vision to language translation. Initial works [31] on video captioning using RNNs perform average feature pooling over the video and bring the task closer to image captioning. The strategy works well for short videos, in which a single major event takes place [38]. For longer videos, different video encoding schemes are proposed. These schemes use either a recurrent encoder [8, 32] or an attention model [36]. In [38] authors use a hierarchical RNN model, with a sentence generator and a separate paragraph generator. The sentence decoder has an attention mechanism to focus on video features while exploiting spatial attention.

Methods for selecting captions from multiple models have been proposed in [10, 29]. Unlike our work, they learn a compatibility score between a single sentence and a given video, without taking in consideration the whole group of output sentences. The authors of [5] use latent topics to guide the sentence generation process. They mine a number of K topics and implicitly learn an ensemble of K decoders, one for each topic. The number of parameters is reduced by a 3-way factorization [19] of the mixture of all topic parameters.

External data can be used to enlarge the linguistic knowledge [33]. In [24] the authors use additional tasks for improving the learning process: an unsupervised video prediction and a language entailment generation task. The usual way of predicting the next word given the previous correct one using standard LSTM decoders creates a difference between the distributions at training vs. testing time, an issue called exposure bias. To tackle it, reinforcement learning approaches have been studied in the context of image captioning [6, 21, 27]. An already trained model is improved by a policy gradient method that works on whole output sentences, guided by a non-differentiable reward, given by the language metrics. Recently this approach has been applied also for video captioning [23, 34].

Main contributions. The main contributions of our approach are: 1) We describe videos in sentences by finding a consensus among multiple encoder-decoder networks. While the individual encoder-decoder networks are able to produce well-formed, fluent sentences it is the consensus among many models that improves the content. The consensus process has two stages. Firstly, we choose a select group of sentences that score well when are evaluated against the others. The next stage we use an Oracle network to pick the final best sentence. The proposed approach achieves state of the art results on the MSR-VTT benchmark.

2) We propose two novel architectures and perform extensive tests with many others adapted from the literature. We also study how different kinds of image, audio or video features influence the final result. We conclude that features that are pretrained on different but related tasks, such as word label prediction or action classification, could impact performance more than the individual architectures. Thus simpler yet higher level tasks such as

action or word prediction, could be an effective intermediary between vision and language.

2 Network architectures

We perform tests with different network architectures and types of features as explained in this section. We also propose two novel encoder-decoder networks for video to language translation. All tested models are based on the encoder-decoder paradigm, and they all have the same LSTM decoder structure. They differ only in the way video content is encoded and in the types of features used.

Seq2Seq model: Describing a video in language could be formulated as a machine translation [2] problem, but it is much more difficult in practice. Instead of translating sentences into a foreign language, now we have to translate visual features into language. Since videos consist of sequences of frames, it is natural to use a recurrent net such as LSTM to produce the encoding. Thus, for every frame the encoder LSTM receives visual features extracted from that particular frame, together with the previous hidden state. The LSTM output at the last step represents the encoding. The encoding could be augmented with extra contextual information by concatenating different visual or audio features, which could be pretrained for different, but related tasks. Such features encode additional knowledge that brings significant improvement in performances (Section 4). We use LSTM cells with one layer in our experiments. The encoder has 512 hidden dimension and the decoder has hidden dimension 512 plus the size of the additional features. Moreover, the encoder part could include an attention mechanism similar to [2] to weight differently the encoder hidden state from each time step, before linearly combining them for the final encoding. In experiments, the attention mechanism brought a marginal improvement.

The decoder, common to all models, works as follows: it starts to output one word at a time. While training, the model receives the previous hidden state and the ground truth word from the previous time step to generate the next word in the sentence. At test time, the ground truth word is replaced by the generated output from the previous state. Our models are trained using softmax cross-entropy loss, unless otherwise specified.

Two-Wings network with sentence reconstruction: The seq2seq model tends to produce, in experiments, simple sentences with very limited vocabulary. We want a stronger decoder, able to capture more realistic, complex sentences. We aim to accomplish this by a model which we term the **Two-Wings network** due to its dual language and vision encoder. Besides the video to language pathway the Two-Wings net has a second encoder-decoder branch for language reconstruction (Figure 1 b). The decoders of the two networks are shared and the second branch is only used at training time for a stronger decoder with more generalization power. The two branches (wings) are trained alternatively, with the decoder having shared parameters. The second, language reconstruction wing is trained for a few iterations and learns to reconstruct broken sentences or to create fluent sentences from sets of words. For a given sentence, we randomly remove half of its words and then shuffle them. We have chosen this approach to make the model more robust since without introducing this noise, the model would simply copy the input. In this way the model can benefit from learning how a correct sentence looks like from a huge amount of text data.

The first wing for vision to language translation is trained for a few iterations while using the same decoder as the other one. By forcing a common decoding part we hope

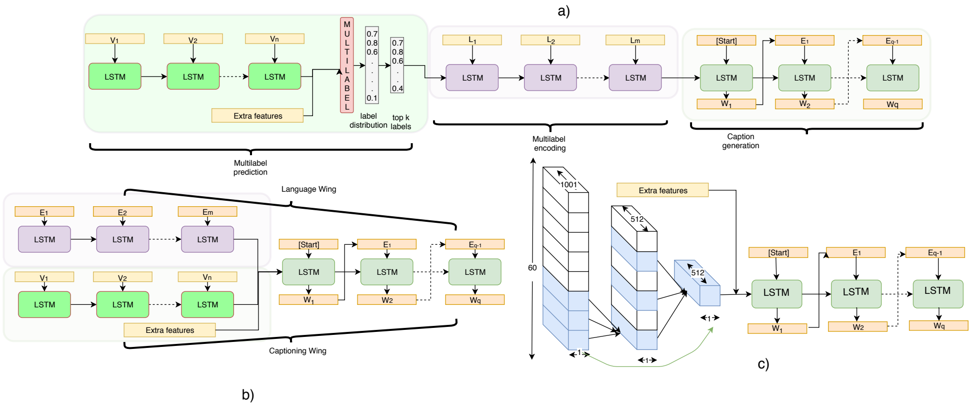


Figure 1: Our main architectures: a) Two-Stage: vision to words to sentences, b) Two-Wings network, c) Temporal Convolutional Network (TCN). The architectures differ in structure significantly and generally output different sentences, but have a similar overall performance.

to learn a common embedding between language and vision with stronger generalization power, better able to capture meaningful content across vision and language. One advantage of the language reconstruction path is that it can be trained on any freely available dataset of texts, unlike the second path, for which there is limited training data available.

We train the language reconstruction wing using a set of 10M sentences (of maximum 20 words) extracted from Wikipedia along with training video captions. Note that the language reconstruction model is used only during training in order to learn a more powerful decoder. During testing, only the second video to sentence network is used.

Two-Stage Network, from video to words to sentences: The Two-Wings network uses the language reconstruction encoder only to train a stronger sentence decoder. The second model we propose, the Two-Stage net, puts two encoder-decoder nets one after the other (Figure 1). The first stage net learns to output words from videos. The second stage net learns to produce sentences from the sets of words given by the first. Thus, word labels provide an intermediate semantic interpretation standing between video data and the final sentence. This idea could increase the generalization power, by focusing on content first (as it is captured by individual words), before learning to produce fluent sentences. Note that the words generation is treated as a multi-label classification problem, with no order imposed.

For the first stage net, we keep the same video encoding scheme as in the Two-Wings model. We replace the language decoder by a model for multi-label prediction, consisting of three fully connected layers that predict the probability of each word label. Given the predicted word labels, we then output a sentence at the second stage, using the same encoder-decoder net used in the language branch of the Two-Wings net.

To form our words labels vocabulary we select the most frequent nouns and verbs from the captioning dataset, resulting in 3059 labels. During training this part, for every sentence we extract the labels and with a given probability we randomly remove some labels and add other ones for robustness. As in the Two-Wings case, we augment the training set by extracting sentences from Wikipedia that contain some words from the 3059 set of labels.

Initially, the two encoder-decoder stages are learned separately and then finetuned end-to-end on the captioning dataset. To make end-to-end training possible, we keep the whole path, from video input to final sentence differentiable, as follows. For each j in the top K predicted labels, we multiply the label embedding $E(j)$ with its predicted soft probability and obtain a differentiable latent representation $L_t = p_{t,j} * E(j)$. Thus, the gradients could be propagated through $p_{t,j}$ back to the video encoder.

Temporal convolutional network In experiments, many of the generated sentences, although fluent, do not reflect the actual video content. This indicates poor encoding of the video. Inspired by [3], we adapt their idea of a temporal convolution network architecture (TCN) to replace the recurrent neural network encoder. The TCN approach was used for sequence to sequence generation on tasks where input consists of long sequences of action segmentation or copy memory. We adapt the TCN model to generate a single output - the embedding for the entire video - in order to provide full information to the decoder before it starts to generate sentences. The decoder structure is the same as before.

The idea behind TCN (Figure 1 c) is to capture how features change over time by using one dimensional temporal filters. By employing a hierarchy of convolutions with increasing dilation rate, the amount of information combined increases exponentially, over different time scales, until it reduces the temporal dimension to one, to capture global content.

For each one of the N_t time steps, we have N_{fe} features, resulting in a tensor of $1 \times N_t \times N_{fe}$ dimension. The network is composed of several blocks of convolution, without padding, in order to reduce the temporal dimension of input from N_t to 1. Each block has 2 dilated convolutional layers [37]. Each applies several 1x3 filters with Relu nonlinearity and batch normalization [15]. The dilation rate is increased with the depth, in order to compute over different scales. Between 2 successive levels, there are residual connections [12]. Batch normalization is applied to ease the optimization process. Based on this architecture, we trained several models, varying the network depth, the size of the filters and the dilation rate.

3 Multiple networks consensus

While our models reach a level of accuracy that stands well against published literature, there is a relatively high degree of variation in their output sentences due to the different ways we encode the video content. Some models tend to have complex, descriptive results with a richer vocabulary, while others generate simple, concise sentences. There is also variation in terms of content vs. form. Some sentences are more fluent and complex (e.g. "a man in a suit and a woman talking about the history of the world"), while others are simpler, but better rooted in actual video content (e.g. "a man is talking about a historical topic").

We noticed that the group of sentences very often contains correct sentences. In order to validate this observation, we selected for each video, from all sentences generated, the one with the best CIDEr metric [30] with respect to ground truth sentences. It turned out that the best selected sentence per video gave on average, over the whole test video set results that are well above state of the art (Table 3). Then we made another observation: models generally produce sentences that gravitate around the correct meaning. Thus, noisy sentence variations could be eliminated if the ensemble of networks could work jointly, as a whole.

Here we propose an efficient **consensus algorithm** for selecting the best sentence in the group, composed of two stages - a first consensus stage using simple agreements between sentences and a second stage that involves training an Oracle network.

First consensus stage. If the group of sentences generated by the models pool contains a strong cluster united around the correct meaning, then we could find the best sentence as the one which agrees most with the others. Thus, for each sentence we compute its CIDEr score against the others and select the one with the highest score. This idea is relatively simple, yet statistically powerful. If the better sentences form a strong cluster and the weaker ones depart from human annotations in random noisy ways, then we could find the good ones by measuring the level of agreement between each sentence and the remaining ones. Accidental agreements are very rare, while agreements based on good content are more likely. This is the basis of our approach. By selecting the sentence with highest agreement score we maximize our chances of selecting a good sentence. In experiments, selecting the top scored sentence significantly improves the results (Table 1 and Figure 2).

Second consensus stage - Oracle network. Often the best quality sentence is within the top C ($C = 3$) according to the consensus score at phase 1. We added a second level of selection by training an Oracle network to help picking the better sentence at the top. We train the Oracle Net to pick the better of two sentences, given a reference video. The video encoding consist of the average over frame features. The two sentences are encoded by LSTMs with shared parameters. All features are then concatenated and passed through 3 fully connected layers to obtain the final output. At inference time, we compare each sentence in the top C, selected through phase one consensus as described previously to all of the others. Each sentence is scored based on the number of pairwise victories. We rank all the sentences according to this score and pick the top one.

We train this model on pairs of sentences generated by our models on videos from training set. Pairs could include two sentences of the reference video, or one for the reference and the other randomly chosen from other videos. The ground truth label is picked according to the CIDEr score w.r.t the corresponding human annotations on the training set.

The consensus algorithm is now complete and proceeds as follows: 1) For each sentence in the group, compute its CIDEr score against the others; 2) Keep the top-C scored sentences. 3) Re-rank the top C using the Oracle Net and output the top scored sentence.

4 Experimental analysis

We trained our models on the challenging MSR-VTT 2016 video captioning dataset and benchmark [35]. This is the main dataset used for experimental testing in recent literature. It contains 10k videos with diverse visual content. Each clip is 10 to 30 seconds long and is annotated with 20 sentences from different people. For comparison with state-of-the-art we use four of the evaluation metrics most often used in the current literature for natural language tasks, shown here in inverse chronological order of their publication date: CIDEr [30], METEOR [4], ROUGE [20] and BLEU [22].

Models in the pool: Our final consensus network works over a pool of 16 models based on the 4 main architectures, differing in the visual, audio and video features used and the number of layers of depth. We observed that the more models we added to the pool the better the performance. Thus, we have 1 basic seq2seq model, 2 seq2seq models with attention, 2 Two-Wings models, 1 Two-Stage network, 4 TCN models, 4 seq2seq models with different groups of extra features added to the encoding, 1 seq2seq model with inception features extracted from small patches on a grid, and the last 1 model uses 2 convolutional layers, one

	Group A		Group A+B		Group A+B+C	
Model	**Cider**	**Meteor**	**Cider**	**Meteor**	**Cider**	**Meteor**
Seq2Seq	36.0	25.5	44.0	27.4	46.1	28.3
Two-Wings	32.2	25.2	42.2	27.3	46.2	28.8
Two-Stage	34.9	25.2	43.3	27.4	45.7	28.4
TCN	36.80	25.5	43.9	27.4	46.1	28.4
Attention	41.0	26.6	44.2	27.5	46.4	28.5
MEAN	36.0 ±2.5	25.6 ±0.5	43.9 ±0.6	27.4 ±0.2	46.0 ±0.9	28.4 ±0.3

Best individual model	46.2	28.8
Consensus	52.1	29.6
Consensus + OracleNet	**53.8**	**29.7**

Table 1: Performance of our models using different image, video and audio features added during three experimental phases: Group A - Inception features; Group B - C3D + MFCC audio features; Group C - VGG audio + Y8M word labels features. In each phase we report the average results of each type of models and the average of all the models. Note how additional features pretrained on different tasks significantly improve performance. Also note the very large performance gain obtained through consensus.

over time dimension and the other over feature dimension as an encoder. Our models were trained using Adam optimizer algorithm with a learning rate set to 0.001 and decayed with a factor of 10. The video embedding dimension is set to 512 units and the decoder hidden dimension varies according to additional features provided.

4.1 Features vs. Network architectures

Here we present in detail the features we bring in, by concatenating them to the video encoding. We added features incrementally, in 3 phases (1).

In the **initial phase** we use object category features from Inception v3 (type A features) to encode the video frames, with no additional features concatenated to the encoding. In the **second phase**, we add extra C3D-Resnet [11] features trained for Kinetics [18] action recognition in video and audio features computed from the means and standard deviations of MFCC feature signals extracted from temporal audio segments (type B features). In the **third phase** of our experiments we bring in stronger higher-level audio features. These are 128-dimension VGG-style deep audio features [13] trained on Youtube-70M. Audio features are constructed concatenating averages over five overlapping segments of video. We also added visual features that we trained for multi-words prediction on a large dataset that combines a subset of Youtube8M [1] and the MSR-VTT videos using as word labels the intersection between their vocabularies. The actual features used are those from the layer preceding the final output. The features added in the third phase are of the type C features.

Our experiments (Table 1 and Figure 2) show that the additional, complementary high level information brought in by features pre-trained on different tasks have an impact on performance comparable to the consensus procedure. This fact strongly suggests that the intermediate level of semantics captured by these features is important for better bridging the gap between vision and language. At the same time, the results suggest that, while a

	Cider	Meteor	Rouge	Bleu 4
VideoLAB [26]	44.1	27.7	60.6	39.1
v2t navig [16]	44.8	28.2	60.9	40.8
Aalto [29]	45.7	26.9	59.8	39.8
ruc-uva [9]	45.9	26.9	58.7	38.7
MT-Ent [24]	47.1	28.8	60.2	40.8
HRL [34]	48.0	28.7	61.7	41.3
dense [28]	48.9	28.3	61.1	41.4
CIDEnt-RL [23]	51.7	28.4	61.4	40.5
TGM [17]	52.9	**29.7**	-	**45.4**
Ours	**53.8**	**29.7**	**63.0**	44.2

Table 2: Comparison with the top models on MSR-VTT 2016 test dataset. We obtain state of the art results on three evaluation metrics.

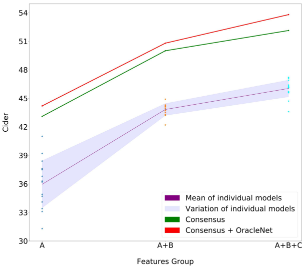


Figure 2: Mean and std of CIDEr score for all our individual models over the 3 features phases along with consensus networks performance.

great effort has been put into creating video captioning datasets, they are still limited for learning such a challenging task. As we discuss in Section 4.3, another potential limitation comes from the current evaluation metrics used in the literature that seem better at evaluating good sentence form and fluency than at capturing the more profound meaning of sentences.

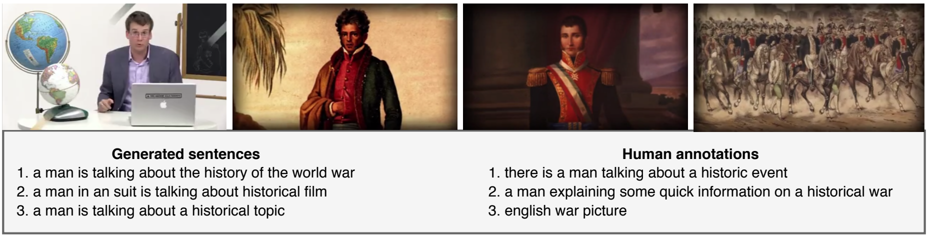
4.2 Comparison with the top models

In Table 2 we compare our method against the top submissions from the MSR-VTT 2016 competition, but also against top models published after the competition on that dataset. While our individual models are very competitive in comparison to the top published methods (1), the consensus between all models significantly improves the performance achieving state of the art results on several metrics. For qualitative results of our system please see Figure 3 and the supplementary material.

Given the multiple architectures in our system, our encodings are diverse and also are more constrained toward the language space by forcing the additional tasks of language reconstruction or multi-label word prediction. In contrast, other methods use a single encoding scheme such as seq2seq or simply mean pooling the features. The authors of [24] use a model similar to our Two-Wings model but they need additional annotated data to learn the language encoding. In [17], they use an implicit ensemble by building a decoder defined by a linear mix of parameters conditioned on multiple latent topics. Because of the non-linearity of the language we argue that it is non-trivial to combine words predictions at the model level, and a simple selection between generated sentence improves results.

4.3 A short discussion on language understanding and evaluation

On close inspection, we found that very small changes in the sentence structure or at the level of words, without changing the overall meaning, may strongly change the metric scores. Consequently, we measure how humans perform against each other on MSR-VTT (Table 3)(by computing the metrics between one human annotation versus the rest) and observed the same instability in the metric scores. The human agreement in terms of the differ-



Generated sentences
1. a man is talking about the history of the world war
2. a man in an suit is talking about historical film
3. a man is talking about a historical topic

Human annotations
1. there is a man talking about a historic event
2. a man explaining some quick information on a historical war
3. english war picture

Figure 3: Qualitative results showing 3 generated sentences from our models along with a few relevant human annotations. The generated sentences are fluent and related in content to the human annotations. Also note how diverse the human annotations are, especially in form, while being highly meaningful. For more qualitative results please see our project page: `https://sites.google.com/view/mining-for-meaning`

ent metrics is quite low and often below the performance of our system (Table 3). Since it is evident that our system does not *speak* better than a human, it must be the metrics that are not quite appropriate. Current methods for sentence generation might be in fact very close to a certain saturation of these metrics.

The truth is that designing good evaluation metrics is, in this case, almost as hard as the research task itself. How could we automatically evaluate the hidden meaning of a sentence if we have not learned yet how to encode this meaning? This seems like a chicken and egg problem. The "meaning" is usually hidden, so we are almost forced to evaluate form, which is explicit. However, there are infinitely many correct sentence forms for a given video sequence. One could expect that a common, higher level representation for understanding the story of what happens in the scene is needed, which sits above vision and language form. Our work suggests that such a representation could be learned indirectly, in a distributed way through intermediate high level but simpler tasks, such as the detection of actions, interactions or more abstract entities (word-labels). They could sit above the physical objects but still below full language expression.

	Cider	**Meteor**	**Rouge**	**Bleu**
Human avg	50.2 ±6.5	29.8 ±3.5	73.1 ±5.4	34.5 ±8.2
Human worst	4.0	15.2	31.9	7.4
Human best	108.1	48.1	84.0	76.1
Ours worst	18.8	22.0	51.1	24.0
Ours best	75.3	34.6	69.7	55.3
Ours	53.8	29.7	63.0	44.2

Table 3: Human performance for MSR-VTT test dataset. Human worst/best are computed by selecting for every video the worst/best sentence with respect to the rest. Ours worst/best are computed by selecting for every video the worst/best scores with respect to ground truth.

In this work we argue that it could be possible to select more meaningful sentences if we look for the consensus of many networks which have learned to output fluent linguistic descriptions that are rooted in powerful visual and audio features pretrained on simpler-to-evaluate, yet high level semantic tasks (such as action or words prediction). Populations of such networks might reach a shared implicit *meaning* through multiple networks consensus.

5 Conclusions

In this work, we presented different architectural approaches to encode content of a video for the purpose of learning to generate captions. We also studied the impact of external features pre-trained on other intermediate tasks and concluded that such features have a strong impact on performance. Describing the dynamic world as it changes through space and time is a very exciting but still extremely challenging problem, which is not well understood. In order to cope with various limitations and reduce the noise of each individual model, we propose a novel approach for finding high quality sentences using multiple encoder-decoder networks. We argue that while the evaluation metrics might not be perfect for individual sentence evaluation, we could still reach correct meaningful sentences through the statistically powerful multiple networks consensus algorithm. We clearly demonstrate the value of our approach by achieving state of the art results on the challenging MSR-VTT benchmark.

Acknowledgements: The authors would like to thank Elena Burceanu and Ioana Croitoru for helpful comments and discussions. This work was partially supported by UEFISCDI, under project PN-III-P4-ID-ERC-2016-0007 and PN-III-P2-2.1-PED-2016-1842.

References

[1] Sami Abu-El-Haija, Nisarg Kothari, Joonseok Lee, Paul Natsev, George Toderici, Balakrishnan Varadarajan, and Sudheendra Vijayanarasimhan. Youtube-8m: A large-scale video classification benchmark. *CoRR*, abs/1609.08675, 2016.

[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[3] Shaojie Bai, J Zico Kolter, and Vladlen Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*, 2018.

[4] Satanjeev Banerjee and Alon Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. 01 2005.

[5] Shizhe Chen, Jia Chen, Qin Jin, and Alexander Hauptmann. Video captioning with guidance of multimodal latent topics. In *Proceedings of the 2017 ACM on Multimedia Conference*, pages 1838–1846. ACM, 2017.

[6] Tseng-Hung Chen, Yuan-Hong Liao, Ching-Yao Chuang, Wan-Ting Hsu, Jianlong Fu, and Min Sun. Show, adapt and tell: Adversarial training of cross-domain image captioner. In *The IEEE International Conference on Computer Vision (ICCV)*, volume 2, 2017.

[7] Noam Chomsky. *Syntactic structures*. Walter de Gruyter, 2002.

[8] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2625–2634, 2015.

[9] Jianfeng Dong, Xirong Li, Weiyu Lan, Yujia Huo, and Cees GM Snoek. Early embedding and late reranking for video captioning. In *Proceedings of the 2016 ACM on Multimedia Conference*, pages 1082–1086. ACM, 2016.

[10] Jianfeng Dong, Xirong Li, Weiyu Lan, Yujia Huo, and Cees GM Snoek. Early embedding and late reranking for video captioning. In *Proceedings of the 2016 ACM on Multimedia Conference*, pages 1082–1086. ACM, 2016.

[11] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? *arXiv preprint*, arXiv:1711.09577, 2017.

[12] 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.

[13] Shawn Hershey, Sourish Chaudhuri, Daniel P. W. Ellis, Jort F. Gemmeke, Aren Jansen, Channing Moore, Manoj Plakal, Devin Platt, Rif A. Saurous, Bryan Seybold, Malcolm Slaney, Ron Weiss, and Kevin Wilson. Cnn architectures for large-scale audio classification. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2017.

[14] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[15] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pages 448–456, 2015.

[16] Qin Jin, Jia Chen, Shizhe Chen, Yifan Xiong, and Alexander Hauptmann. Describing videos using multi-modal fusion. In *Proceedings of the 2016 ACM on Multimedia Conference*, pages 1087–1091. ACM, 2016.

[17] Qin Jin, Shizhe Chen, Jia Chen, and Alexander Hauptmann. Knowing yourself: Improving video caption via in-depth recap. In ACM MM, 2017.

[18] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

[19] Alex Krizhevsky, Geoffrey Hinton, et al. Factored 3-way restricted boltzmann machines for modeling natural images. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 621–628, 2010.

[20] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Proc. ACL workshop on Text Summarization Branches Out*, page 10, 2004.

[21] Siqi Liu, Zhenhai Zhu, Ning Ye, Sergio Guadarrama, and Kevin Murphy. Improved image captioning via policy gradient optimization of spider. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[22] 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 on Association for Computational Linguistics*, ACL '02, pages 311–318, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics. doi: 10.3115/1073083.1073135.

[23] Ramakanth Pasunuru and Mohit Bansal. Reinforced video captioning with entailment rewards. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 979–985. Association for Computational Linguistics, 2017.

[24] Ramakanth Pasunuru and Mohit Bansal. Multi-task video captioning with video and entailment generation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1273–1283, Vancouver, Canada, July 2017. Association for Computational Linguistics.

[25] Steven Pinker. *The language instinct: How the mind creates language*. Penguin UK, 2003.

[26] Vasili Ramanishka, Abir Das, Dong Huk Park, Subhashini Venugopalan, Lisa Anne Hendricks, Marcus Rohrbach, and Kate Saenko. Multimodal video description. In *Proceedings of the 2016 ACM on Multimedia Conference*, pages 1092–1096. ACM, 2016.

[27] Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. Sequence level training with recurrent neural networks. *arXiv preprint arXiv:1511.06732*, 2015.

[28] Zhiqiang Shen, Jianguo Li, Zhou Su, Minjun Li, Yurong Chen, Yu-Gang Jiang, and Xiangyang Xue. Weakly supervised dense video captioning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, 2017.

[29] Rakshith Shetty and Jorma Laaksonen. Frame-and segment-level features and candidate pool evaluation for video caption generation. In *Proceedings of the 2016 ACM on Multimedia Conference*, pages 1073–1076. ACM, 2016.

[30] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4566–4575, 2015.

[31] Subhashini Venugopalan, Huijuan Xu, Jeff Donahue, Marcus Rohrbach, Raymond Mooney, and Kate Saenko. Translating videos to natural language using deep recurrent neural networks. *arXiv preprint arXiv:1412.4729*, 2014.

[32] Subhashini Venugopalan, Marcus Rohrbach, Jeffrey Donahue, Raymond Mooney, Trevor Darrell, and Kate Saenko. Sequence to sequence-video to text. In *Proceedings of the IEEE international conference on computer vision*, pages 4534–4542, 2015.

[33] Subhashini Venugopalan, Lisa Anne Hendricks, Raymond Mooney, and Kate Saenko. Improving lstm-based video description with linguistic knowledge mined from text. *arXiv preprint arXiv:1604.01729*, 2016.

[34] Xin Wang, Wenhu Chen, Jiawei Wu, Yuan-Fang Wang, and William Yang Wang. Video captioning via hierarchical reinforcement learning. *CVPR*, 2018.

[35] 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*, pages 5288–5296. IEEE, 2016.

[36] Li Yao, Atousa Torabi, Kyunghyun Cho, Nicolas Ballas, Christopher Pal, Hugo Larochelle, and Aaron Courville. Describing videos by exploiting temporal structure. In *Proceedings of the IEEE international conference on computer vision*, pages 4507–4515, 2015.

[37] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015.

[38] Haonan Yu, Jiang Wang, Zhiheng Huang, Yi Yang, and Wei Xu. Video paragraph captioning using hierarchical recurrent neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4584–4593, 2016.