11-775 Project: Localizing Moments in Video with Natural Language

Anusha Prakash
anushap@andrew.cmu.edu

Bhavya Karki
bkarki@andrew.cmu.edu

Pravalika Avvaru
pavvaru@andrew.cmu.edu

Anuva Agarwal
anuvaa@andrew.cmu.edu

Abstract

We propose a system that retrieves specific temporal segments from a video given a natural language text description which can be applied to a wide range of events like retrieving unforgettable memories from a long personal holiday video, searching for educational videos from a library of videos, etc. This project implements as baseline the video-language model in (Lisa Hendricks et al, 2017) [1]. The work aims to improve upon the existing baselines of 41.08% mean Intersection over Union (IoU), average rank@1 of 28.10% and average rank@5 of 78.21% which was obtained using an LSTM based architecture with glove embedding for language model and RGB and optical flow (local and global) video features. We experiment with multiple approaches in terms network architecture, embeddings, different visual and audio features, fusion techniques etc. The project also focuses on elaborate ablation study discussing the effect of each of these features and also the impact of applying models trained on related activity recognition datasets to aid in achieving our target.

Keywords: Moments, LSTM, language model, Kinetics, Resnet, activity recognition, image captioning, multi-modal, embedding space

1 Introduction

Retrieving relevant videos given a natural language has been long studied. Whereas, retrieving a specific temporal segment of a video given a natural language description of the segment is quite challenging and new. Requiring to understand both language and video makes the task more difficult. Consider an example: A baby stands up, falls down and again stands up to start walking. If we want to infer to a particular temporal segment - "when the baby resiliently started walking again after falling down", it's not enough to use action, object or attribute keyword. Instead, moment must be defined by when and how specific actions are affected and dependent to other actions in the video. Such moments will identify the start and end points within the video which captures the temporal moment.

Most of the existing methods (Torabi,Tandon, et al)(Yingwei et al)[2, 3] retrieve videos using natural language but they don't identify the moment within the video. Also, most of the related work deal with action recognition in a themed set of videos such as a particular sport, or baking, or any particular activity. To localize moments in a generic set of videos, (Lisa Hendricks et al, 2017) [1] develop a novel video-language model which uses global video features, specific moment features along with language features. The baseline uses a Moment Context Network (MCN) to include the global video feature to provide temporal context and a temporal endpoint feature to indicate the

moment in the video. The work is carried out on the Distinct Describable Moments (DiDeMo) dataset which consists of over 40,000 pairs of localized video moments and corresponding natural language.

From a broader perspective, the work aims to correctly answer questions of the kind: 'When does a particular moment occur in a video?'. Therefore, the problem we try to solve is to retrieve a specific temporal segment or moment from a video, given a natural language text description. We explore using a different set of visual features apart from the features used in the baseline. We also experiment with different network architectures, changing the vector representations of the language modality, extracting audio features using SoundNet[4] architecture, extracting I3D features [5] and model trained on the Kinetics dataset [6] with the aim of studying the impact of these constructs on the mentioned task and improving the retrieval performance.

The sections following the introduction give deeper insights into the work done in this project. Section 3 talks about the related literature and prior work that has been implemented in this field. The next set of sections explain in detail the different experiments and ablation studies and an in depth analysis and interpretation of our experiments and their corresponding results.

2 Related Work

Over recent years, there has been an increased interest in jointly modeling images/videos and natural language sentences. The challenge lies in encoding temporal aspects of the video, to find contextual relationship between frames and having to process sheer volume of the video data. [7] implemented joint models taking a combination of the simple average or weighted average over video FC-7 features, and Language LSTM or Word2Vec features on MS-COCO and LSMDC16 Movies dataset. [8] tackle the challenge of generating unambiguous descriptions of a specific object or region in an image, which can comprehend such an expression to infer which object is being described. For generating the descriptions, they use RNN to generate descriptive sentences and a bean search to find the most probable sentence that distinguishes the input region from other candidate regions. For the description comprehension part, they adopt a ranking-based approach to generate a set of region proposals and then ask the system to rank these by probability. [9] propose a Compositional Modular Networks (CMNs) - an end-to-end trained model that learns language representation and image region localization jointly. The model localizes a referential expression by grounding the components in the expressions and exploiting their interactions such that the meaning of a complex expression is determined by the meanings of its constituent expressions and the rules used to combine them. The score of a region is determined by simultaneously looking at whether it matches the description of the subject entity and whether it matches the relationship with another interacting object entity mentioned in the expression.

[10] developed a model that works on a finer level and embeds fragments of images and fragments of sentences (typed dependency tree relations) into a common space. [11] introduce a neural network-based embedding models for video, sentence, and image inputs whose parameters can be learned jointly. They utilize web image search in a sentence embedding process to disambiguate fine-grained visual concepts. Another related work [12] encode temporal data using a new concept called rank pooling where they learn a ranking machine that captures the temporal evolution of the data and uses the parameters of the latter as a representation. Another popular work in the domain of action recognition was by (Christoph et al) [13] who introduce an architecture that generalizes ResNets for the spatiotemporal domain by introducing residual connections between the appearance and motion pathways of a two-stream architecture and train the model end-to-end to allow hierarchical learning of complex spatiotemporal features. Much of our work is influenced by the literature review above.

3 DiDeMo Dataset

A major challenge when designing algorithms to localize moments with natural language is that there is a dearth of large-scale datasets which consist of referring expressions and localized video moments. To mitigate this issue, the authors of [1] introduce the Distinct Describable Moments (DiDeMo) dataset which includes over 10,000 25-30 second long personal videos with over 40,000 localized

text descriptions. They randomly selected over 14,000 videos from YFCC100M [14] which contains over 100,000 Flickr videos with a Creative Commons License.

Videos in DiDeMo represent a diverse set of real-world videos, which include interesting, distinct moments, as well as uneventful segments which might be excluded from edited videos. Because videos are curated from Flickr, DiDeMo reflects the type of content people are interested in recording and sharing. Figure 1 depicts the example video moments and referring expressions associated with them in the DiDeMo dataset. Consequently, DiDeMo is human-centric with words like "baby", "woman", and "man" appearing frequently. Since videos are randomly sampled, DiDeMo has a long tail with words like "parachute" and "violin", appearing infrequently (28 and 38 times). Important, distinct moments in a video often coincide with specific camera movements. For example, "the camera pans to a group of friends" or "zooms in on the baby" can describe distinct moments. Many moments in personal videos are easiest to describe in reference to the viewer (e.g., "the little boy runs towards the camera"). DiDeMo contains more sentences with temporal indicators than natural language object retrieval and video description datasets, as well as a large number of spatial indicators. DiDeMo has a higher percentage of verbs than natural language object retrieval datasets, suggesting understanding action is important for moment localization in video.

Figure 1: Sample Video Moments along with corresponding referring expressions

Their annotations include descriptions which are temporally grounded in videos. Each video is split into 5-second temporal chunks. The first temporal chunk corresponds to seconds 0-5 in the video, the second temporal chunk corresponds to seconds 5-10, etc. The dataset can be accessed through json files which contain the following fields: annotation_id (annotation ID for description), description (description for a specific video segment), video (video name), download_link (a download link for the video), num_segments (some videos are a little shorter than 25 seconds, so were split into five temporal chunks instead of six) and times (ground truth time points marked by annotators).

We will split the DiDeMo videos into training (8,395), validation (1,065), and testing (1,004) sets. Videos from a specific Flickr user only appear in one set. Candidate moments come from the temporal segments defined by the gifs used to collect annotations. A 30 second video is to be broken into six five-second gifs. Moments can include any contiguous set of gifs, so a 30-second video contains 21 possible moments. Figure 2 shows the distribution of segments in DiDeMo ground truth labels. It can be observed that moments tend to be short and occur towards the beginning of videos.

4 Evaluation Metrics

We will use the same metrics to evaluate our work as the DiDeMo paper. They measure the performance of each model with Rank@1 (R@1), Rank@5 (R@5), and mean intersection over union (mIoU). They compute the score for a prediction and each human annotation for a particular

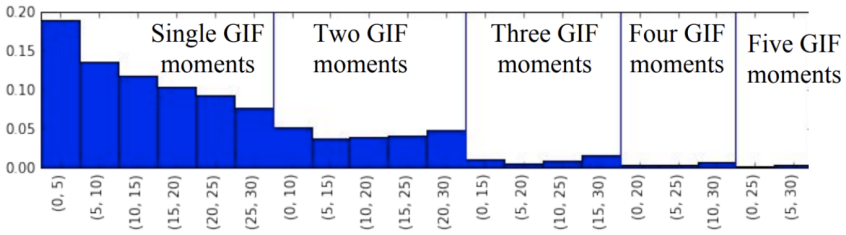

Figure 2: Distribution of segments marked in DiDeMo

description/moment pair instead of consolidating all human annotations into one ground truth.

The baseline we relay on for these evaluation metrics is the Moment Frequency Prior and is reported in Table 1. This has a tendency to select short moments towards the beginning of videos. It selects moments which correspond to gifs most frequently described by annotators. Based on Figure 2, it would mean (0,5) moment most of the times as it's the most frequent.

Model	Avg IoU	Avg Rank@ 1	Avg Rank@ 5
Moment Frequency Prior	26.65	19.40	66.38

Table 1: Moment Frequency Prior Baseline

5 Baseline or Initial Experiments

For the baseline, we implemented the Moment Context Network (MCN) described in [1]. Figure 3 depicts the architecture of MCN. It relies on local and global features. We extracted the visual temporal context features which encode the video moment by integrating both local features, which depicts what occurs within a specific moment, global video context, which provide context for a video moment and temporal endpoint features, which indicate when a moment occurs within the video.

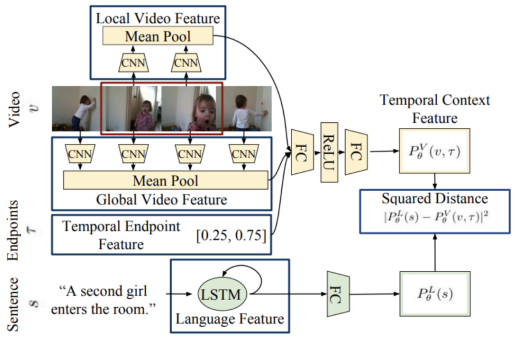

Figure 3: Moment Context Network Architecture

We employed the pre-trained VGG19 model to extract the high-level RGB video features for each video keyframe and then used this to extract the local and global features. We used the Temporal Segment Network to extract the optical flow features. RGB frames and optical flow frames are the two sources of visual input modalities we used. For a given natural language description, we extract the language features using an LSTM network. The language embedding used here is the Glove6B of 300 dimension. The Joint Video and Language Model is the sum of squared distances between embedded appearance, flow, and language features. A late fusion technique is used to fuse the RGB

and optical flow results.

We employed the Triplet Ranking loss over the positive and negative inter and intra video segments to train our model and the Distinct Describable Moments (DiDeMo) dataset the paper proposes to train and evaluate our model. The baseline and reproduced baseline results when the model is trained for 30,000 epochs with late fusion weight of 0.5 and features LSTM-Fusion + global + tef is documented in Table 2. It can be observed that the two results closely match.

Model	Avg IoU	Avg Rank@ 1	Avg Rank@ 5
Baseline	0.4108	0.2810	0.7821
Reproduced Baseline	0.405315	0.270828	0.785377

Table 2: Baseline and Reproduced Baseline Results

6 Experiments, Results and Analysis

We carry out a number of experiments to in order to (i) study the impact of using different high and low level visual features (ii) see how incorporating audio features will change the baseline (iii) draw on novel multi-modal fusion techniques (iv) study the importance of incorporating pre-trained models from related tasks to solve our problem (v) modify the network architecture - improve upon baseline scores. Accordingly, we divide the approaches based on the space in which the methodologies are employed.

6.1 Language Embedding

The baseline experiments use a 300 dimension dense vector representation of the sentences, i.e Glove6B [15] embedding 300D. Though our dataset contains over 40,000 sentences, the vocabulary size is considerably small and it is also still very small in comparison to datasets used for natural language object retrieval. Therefore, we find that representing words with denser word embeddings should be the right approach. Hence we used the 200D Glove embedding instead. A plot of variation of accuracy versus glove embedding dimension can be seen in Figure 4 and we see that the accuracy does not vary largely after 200 dimension.

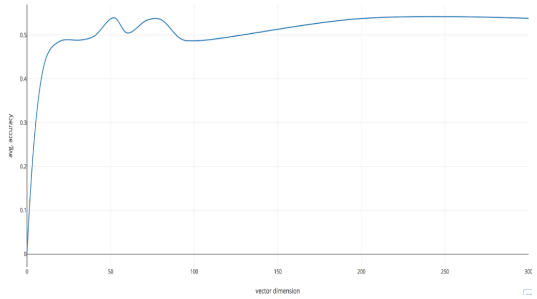

Figure 4: Glove embedding dimensions versus accuracy

Model	Avg IoU	Avg Rank@ 1	Avg Rank@ 5
200 D	**0.389062**	**0.26280**	**0.762746**
300 D	0.386642	0.261378	0.772196

Table 3: Results of Glove 200D versus 300 D

The results of our experiment with 200 D in contrast to 300D is tabulated in Table 3. We find that it did not make much difference to the results but increased it by a very small margin in Avg IoU and AvgRank@1.

6.2 Language Model

The referring expression features were captured by initializing the words with a Glove embedding and passing it through a neural network to obtain the language features. The baseline used a LSTM layer for this task. We experimented with BiLSTM and RNN as well. Since the referring expressions are short, we felt that may be an RNN itself would be enough to capture the language features and capturing long term dependencies isn't very necessary. In general for language, a BiLSTM is known to give a superior performance, and it captures information from both directions. Hence we also decided to experiment with BiLSTM. The results of these can be found in Table 4.

Language Model	Avg IoU	Avg Rank@ 1	Avg Rank@ 5
RNN	0.253292	0.191246	0.262373
BiLSTM	0.238544	0.285571	0.723228
LSTM	**0.386642**	**0.261378**	**0.772196**

Table 4: Results on Different Language Models

Based on the results, we observed that LSTM itself performed the best. RNN gave very low performance, and we believe that this may eb due to the vanishing gradients problem and it's inability to capture sequences well. BiLSTM also did not perform as expected and this could because reversing of inputs to LSTM layers hinders sequence of moments and leads loss of temporal context. The LSTM which performs best handles both these shortcomings.

6.3 Joint Embedding Distance Metrics

The joint embedding distance is calculated between embedded appearance, flow, and language features. This distance tries to calculate the distance between these features to calculate the loss which is propagated through the network. This join embedding distance metric learns a joint video-language model in which referring expressions and video features from corresponding moments are close in a shared embedding space. This closeness is measured using either Euclidean distance, Dot product / Cosine distance, Element-wise distance.

Distance Metric	Avg IoU	Avg Rank@ 1	Avg Rank@ 5
Dot Product	**0.39201**	**0.26719**	**0.78121**
Element-wise	0.373324	0.26013	0.73450
Euclidean	0.386642	0.261378	0.772196

Table 5: Results on Different Distance Metrics

The results of our experiment with different Joint Embedding Distance Metrics can be seen in Table 5. We observe that dot product gives us better results than Euclidean and Element-wise distance.

6.4 Experiments in the Visual Feature Space

We decided to carry out an extensive study to understand how different features would contribute in the task of identifying video segments from natural language queries. We also performed different fusion techniques on these new features in order to obtain an improved performance.

ResNet-50 : We extracted ResNet-50 features pretrained on ImageNet dataset. We extracted the features from the last fully connected layer. We performed both a mean and max pooling over these features and performed an early, late and double fusion with optical flow features to enhance the model performance. The features were extracted on video frames sampled at a rate of 3 frames per second, which essentially gave us 90 frames for each 30 second video clip, which was richer than the VGG19 features used in baseline.

The results of our fusion experiment with ResNet Max which performed better than ResNet Mean features are documented in Table 6. It can be concluded that for our particular dataset and problem statement, late fusion performed much better than early fusion. We also were able to beat the original paper baseline with this.

Resnet Features	Avg IoU	Avg Rank@ 1	Avg Rank@ 5
Early Fusion (ResNet + Flow)	0.384305	0.259886	0.768466
Late Fusion (ResNet + Flow)	**0.418452**	**0.285252**	**0.79806**
Late Fusion (ResNet + VGG + Flow)	0.412504	0.281592	0.791791

Table 6: Results on Visual Feature Fusion

6.5 Fusion Techniques

Early fusion fuses modalities in feature space whereas Late Fusion fuses modalities in semantic space. In early fusion, unimodal features are first extracted and then combined into a single representation, after which classifiers are trained on top of this representation. Early fusion yields a truly multimedia feature representation, since the features are integrated from the start. However, it is not able to combine features into a common representation. In contrast, in late fusion, features are combined into a joined multimodal representation and semantic concepts are learned directly. The scores of classifiers trained on individual modalities are combined to give final scores, which are able to capture the best of both worlds. Thus, we see that even in our case, late fusion is able to give better results than early fusion as it is able to take into account the different aspects of each modality.

In addition to performing a late fusion on Resnet-max and optical flow features, we also included the VGG features in the late fusion model to see if these features capture any additional information that might have been missed by the earlier features. This triple fusion did not prove to be very beneficial as can be seen in Table 6

BiDNN: (Vukotic et al) [16] introduce a Bidirectional (Symmetrical) Deep Neural Networks which are similar to (or a variation of) multimodal autoencoders. They offer cross-modal translation and superior multimodal embedding created in a common representation. Learning is performed in both directions: one modality is presented as an input and the other as the expected output while at the same time the second one is presented as input and the first one as expected output. This is equivalent to using two separate deep neural networks and tying them (sharing specific weight variables) to make them symmetrical. Learning of the two crossmodal mappings is then performed simultaneously and they are forced to be as close as possible to each other's inverses by the symmetric architecture in the middle. A joint representation in the middle of the two crossmodal mappings is also formed while learning. Architecture of BiDNN can be seen in Figure 5

This is an early fusion technique to learn a joint cross modal embedding. We implemented a BiDNN and trained it end to end to obtain the joint representation of ResNet-50 Max and Optical Flow modalities. ResNet features essentially capture object identification features whereas the optical flow features capture the activities occurring across the frames, thus we thought it would be a good idea to fuse these two features to get a joint representation in order to capture complimentary information.

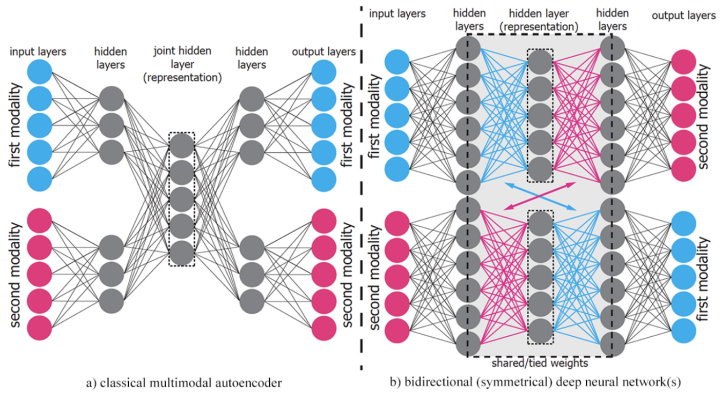

Figure 5: BiDNN - Multimodal Fusion Technique

The results of our experiment with BiDNN features are in Table 7. We see that we do gain some performance boost but it is comparable to that obtained by late fusion of features alone. Since this is an early fusion technique, we thought of trying double fusion as well, in order to see if combining the two techniques (early and late fusion) would enhance the performance. We assigned different weights to each of the features in order to conduct a deeper analysis. However, we were surprised to see that it did not help much in boosting the performance. We think this is because late fusion and early fusion individually captured all the useful information and hence double fusion did not do much to boost the performance further.

Therefore a mixed level fusion of BiDNN with equally weighted ResNet max and Flow features were used and the results showed some improvement over the double fusion. Future work could head in the direction of implementing the BiDNN across multiple modalities.

Features Used	Avg IoU	Avg Rank@ 1	Avg Rank@ 5
BIDNN fusion - ResNet + Flow	0.418332	0.285572	0.795025
Mixed fusion - (l/2)ResNet-max + (l/2)Flow + (1-l)BiDNN	**0.419301**	**0.286567**	**0.797761**
Double fusion - (l/2)ResNet-max + (1-l)Flow + (l/2)BiDNN	0.416826	0.285572	0.798756

Table 7: Results on BiDNN Features - Double and Mixed fusion

6.6 Audio (SoundNet) Features

Most of videos contain audio and many queries revolve around extracting moments involving sound activities like "The baby laughing" or "The man singing in front of the family". Hence we thought it might be interesting to extract sound features and see if they enhance performance of the model. We extracted high level latent features from the audio signals using - Soundnet introduced in [17] which learn rich natural sound representations by capitalizing on large amounts of unlabeled sound data collected in the wild. They leverage the natural synchronization between vision and sound to learn an acoustic representation using two-million unlabeled videos. We extracted features from the Maxpool of the 5th Conv layer of their model and trained it stand alone as well as applied fusion methods.

With respect to the stand alone model, for 410 queries SoundNet performed much better than visual features, with the total number of queries being 4029. These were the queries where the moment as well as the referring expression contained audio and sound words respectively, for instance in Figure 6. Here the query was to retrieve the moment before the person started playing the instrument. This was captured well by Soundnet features as they aided in clearly recognizing the exact moment before the sound started playing, which was not possible to achieve by visual features alone.

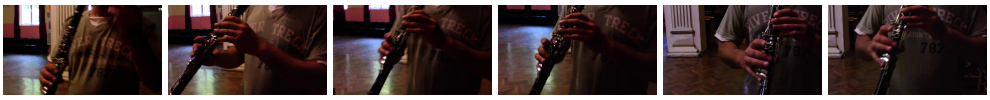

Figure 6: Improvement obtained by SoundNet

Figure 7: No improvement obtained by SoundNet

The reasons for the suboptimal performance of SoundNet on most other queries though was simply because very few queries referred to extracting moments concerning sound related activities and most of the queries focused on concepts such as camera panning (as in 8), and spatial and temporal words.

Figure 8: No improvement obtained by SoundNet

Another reason why the queries didn't perform well can be seen from 7. Here the query was to extract the moment where the man started singing in front of the family. The sound features didn't help here as the man is singing throughout the video and hence the sound features couldn't help in this. Another example would be a query, "Camera focuses on sign saying **zombie disco disaster**", here although the word *saying* is a sound word, it refers to the text on the sign and not any audio related to it.

The results of our fusion experiment with SoundNet features can be seen in Table 8.

Features Used	Avg IoU	Avg Rank@ 1	Avg Rank@ 5
Late fusion (l/2) resnet max + (l/2) soundnet + (1-l)flow	0.409814	0.274402	0.809809
Late fusion (l/2) resnet max + (1-l) soundnet + (l/2)flow	**0.419371**	**0.278469**	**0.814354**

Table 8: Results on SoundNet Features

6.7 I3D Features

We also extracted Inflated 3D ConvNet (I3D) features pretrained on Kinetics Human Action Video dataset with 400 human action classes and over 400 clips per class, which is collected from realistic, challenging YouTube videos. I3D features are extracted by expanding filters and pooling kernels of very deep image classification ConvNets into 3D, making it possible to learn seamless spatio-temporal feature extractors from video while leveraging successful ImageNet architecture designs and even their parameters. We decided to use these features for two reasons. First, our dataset contains realistic videos containing human activities and most of the queries involve retrieving moments specific to certain activities. We found a good mapping of the activities in Kinetics dataset to out DiDeMo dataset and thus thought these features would help better capture the representation of activities. Second, since the I3D model captures temporal features and since our task is essentially to retrieve temporal segments from the video, we felt these features would be a good representation of each candidate temporal segment in the videos. [Since the extraction for 10,000 videos takes very long, we decided to consider only 6 possible segments rather than 21 for this case.]

The results of our experiment with I3D features are documents in Table 9. It can be observed that the higher dimension (1024d) features perform better and this is intuitive too as it captures more information than 400d features.

Features Used	Avg IoU	Avg Rank@ 1	Avg Rank@ 5
I3D (1024d)	**0.390741**	**0.361905**	**0.944974**
I3D (400d)	0.3806	0.352381	0.939153
Baseline (6 moments)	0.389771	0.359259	0.945767

Table 9: Results on I3D Features

6.8 Activity Recognition

As described above, the Kinetics Human Action Video dataset has 400 human action classes and over 400 clips per class. This is leveraged in Activity Recognition. The I3D video features are sent through the above mentioned network to classify it into one of the human classes. Segment wise features are used to predict the classification of that segment into that video. The model outputs a probability distribution across the 400 classes. Top 10 classes into which the segment falls is taken into consideration for this task. [Since the extraction for 10,000 videos takes very long, we decided to consider only 6 possible segments rather than 21 for this case.]

The Natural Language query and the human classes predicted by the model for each segment are used to calculate the similarity between class per segment and the NL query. Sentence similarity is used to calculate the correlation. The segment with the highest correlation with the NL query is given a high score. These scores for each query are ensembled with the scores form the main baseline system to obtain cumulative results.

The results of our experiment with Activity Recognition features are in Table 10. We obtained some performance improvement in videos where the activities mapped to the activities in the kinetics dataset. For example, the video in Figure 9 maps to the activity class of "climbing ladder" and hence for the query "boy starts climbing ladder", this model was able to retrieve the correct moment.

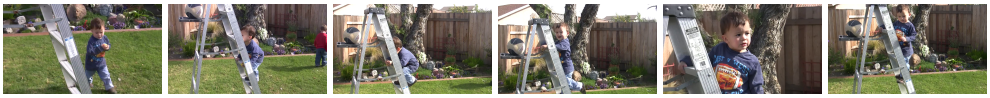

Figure 9: Improvement obtained by I3D based Activity model

Features Used	Avg IoU	Avg Rank@ 1	Avg Rank@ 5
I3D (sent2vec)	**0.390251**	**0.35978**	**0.94576**
Baseline (6 moments)	0.389771	0.359259	0.945767

Table 10: Results on Activity Recognition Features

From the above table it is seen that using I3D with sent2vec embedding directly produced comparable results to our fusion scores as seen in the previous experiment and better results than the 6 moments baseline. This implies that using the class prediction directly to obtain the correlation with the query is a good idea.

6.9 Image Captioning

The activity recognition performed in the previous experiment recognized activities into 400 classes of activities and the query similarity is measured against the class it belongs to. The class names are inherently short phrases ranging from 1 to 4 words, whereas the query is often much longer. Due to this we might not get a very good similarity score with the query. Another drawback that we found in this approach is that the action recognition dataset identifies only human activities. Movements of animals or non-living entities may not be correctly identified. For instance, a cat sleeping by is an activity of the cat, but there is an uncertainty in which class it is going to be classified into. To overcome these issues, we try to incorporate the concept of image captioning to generate longer descriptions. The captions also identify the entities more accurately since they are not constricted to human activities only.

Hence we generate captions for each temporal segment of the video and then calculate the cosine similarity between the generated caption and the natural language query to find the segment of the video with the maximum correlation with the NL query. The captions are generated using a model[18] whose network is combination of Convnets and LSTMs. The model was pre-trained on Flickr 8K, Flickr 30K, and MS COCO datasets. This is then ensembled with the output of the main baseline system to calculate the final scores.

Since the time to generate captions over multiple frames for the 10k odd videos was too long, we built a proof of concept system for the test videos only using only 6 moments instead of 21 moments. The results of the image captioning experiments are documented in Table 11 and 12.

Features Used	Accuracy
Image Captioning	52.2%

Table 11: Standalone Results on Image Captioning Features (Single GIFs)

Features Used	Avg IoU	Avg Rank@ 1	Avg Rank@ 5
VGG+flow+caption generated features	0.381233	0.36475	0.95172

Table 12: Results on Image Captions generated

Though we performed the experiment over a small set as a proof of concept, our results indicate that image captioning has the potential to do better than the activity recognition because the captions generated are longer and match the query better.

7 Conclusion and Future Work

This work explores the usage of natural language to retrieve specific temporal segments from a video. The project established the baseline set by the original works of [1] and tries to improve upon it. We study the baseline architecture and experientially agree with the author that triplet loss is the best measure of loss for this question. We summarize the original work done in terms of the approaches used. Our first approach plays with the embedding space, trying to use a denser representation given the smaller vocabulary. Next we try to use variations of the sequential language model such as using bidirectional LSTM in place of LSTM to learn better, but it learns otherwise. We attribute this drop in scores to the fact that BiLSTM over the description loses the context captured by segments in the moments. Next we explore using combinations and varieties of visual features such as I3D features, ResNet50 features etc. Using ResNet substantially increased the retrieval scores and helped us beat the baseline. Further, the project also studies the impact of using audio features on this dataset. The audio feature used was SoundNet features and we see that it performs well only for a small subset of videos that have distinguishable sound signals and the audio signals correlate to the actions in the moment. The next set of experiments which deals with Bidirectional Deep Neural Networks investigates the use of neural networks as a multi-modal fusion technique. Transfer learning is an extremely useful concept when dealing with related tasks. We employ the concept to leverage the 400 classes from Kinetics Human Action Video dataset to boost our performance. Cosine similarity measure between natural language description and class labels generated from I3D features were used to upweight scores of certain segments of a video. Similarly, scores calculated between the image captions generated per segment and the NL description were used to bias the predictions to improve the performance. All the experiments present an in-depth analysis into various modalities and their effect on moment retrieval from a video.

As for furthering this work, the first step would be to use the BiDNN architecture to fuse different modalities such as language + visual or audio + visual. Another direction in which the work can be extended is to use attention over object detection in video frames. Since the architecture used here is simple, we posit that deeper and more intricate architectures can be designed to enhance the scores. Additionally, the methods described in this work can be applied across other datasets and those performances can be studied.

8 Milestones

Keeping in mind the time complexity and the number of experiments, we ensured we were ahead of our initial milestones so that we could explore more interesting techniques. Shown below are the phase timelines that we followed:

- Proposal - Ready for review : March 10th
- Baseline Implementation : March 17th
- Ablation Study - Embedding Space : March 20th
- Ablation Study - Visual Features : March 31st
- BiLSTM Approach : April 8th
- BiDNN and other Fusion Techniques: April 16th
- Audio Features : April 20th
- Activity Recognition : April 25th

- I3D Kinetics Feature : April 29th
- Analysis and Presentation : May 2nd
- Image Captioning (Proof of concept) : May 9th
- Report and Future work : May 11th

References

[1] Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan C. Russell. Localizing moments in video with natural language. *CoRR*, abs/1708.01641, 2017.

[2] Atousa Torabi, Niket Tandon, and Leonid Sigal. Learning language-visual embedding for movie understanding with natural-language. *CoRR*, abs/1609.08124, 2016.

[3] Yingwei Pan, Tao Mei, Ting Yao, Houqiang Li, and Yong Rui. Jointly modeling embedding and translation to bridge video and language. *CoRR*, abs/1505.01861, 2015.

[4] Yusuf Aytar, Carl Vondrick, and Antonio Torralba. Soundnet: Learning sound representations from unlabeled video. *CoRR*, abs/1610.09001, 2016.

[5] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? A new model and the kinetics dataset. *CoRR*, abs/1705.07750, 2017.

[6] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, Mustafa Suleyman, and Andrew Zisserman. The kinetics human action video dataset. *CoRR*, abs/1705.06950, 2017.

[7] Atousa Torabi, Niket Tandon, and Leonid Sigal. Learning language-visual embedding for movie understanding with natural-language. *CoRR*, abs/1609.08124, 2016.

[8] Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L. Yuille, and Kevin Murphy. Generation and comprehension of unambiguous object descriptions. *CoRR*, abs/1511.02283, 2015.

[9] Ronghang Hu, Marcus Rohrbach, Jacob Andreas, Trevor Darrell, and Kate Saenko. Modeling relationships in referential expressions with compositional modular networks. *CoRR*, abs/1611.09978, 2016.

[10] Andrej Karpathy, Armand Joulin, and Fei-Fei Li. Deep fragment embeddings for bidirectional image sentence mapping. *CoRR*, abs/1406.5679, 2014.

[11] Mayu Otani, Yuta Nakashima, Esa Rahtu, Janne Heikkilä, and Naokazu Yokoya. Learning joint representations of videos and sentences with web image search. *CoRR*, abs/1608.02367, 2016.

[12] Hakan Bilen, Basura Fernando, Efstratios Gavves, and Andrea Vedaldi. Action recognition with dynamic image networks. *CoRR*, abs/1612.00738, 2016.

[13] Christoph Feichtenhofer, Axel Pinz, and Richard P. Wildes. Spatiotemporal residual networks for video action recognition. *CoRR*, abs/1611.02155, 2016.

[14] Bart Thomee, David A Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. The new data and new challenges in multimedia research. *arXiv preprint arXiv:1503.01817*, 1(8), 2015.

[15] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. 2014.

[16] Vedran Vukotić, Christian Raymond, and Guillaume Gravier. Bidirectional joint representation learning with symmetrical deep neural networks for multimodal and crossmodal applications. In *Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval*, pages 343–346. ACM, 2016.

[17] Yusuf Aytar, Carl Vondrick, and Antonio Torralba. Soundnet: Learning sound representations from unlabeled video. In *Advances in Neural Information Processing Systems*, pages 892–900, 2016.

[18] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(4):664–676, April 2017.