Video captioning and sign language interpreting

Jiuge Ren
Yeshiva University
jren@mail.yu.edu

Abstract

Approximately 11.5 million Americans currently experience some form of hearing impairment, which can range from trouble understanding speech to severe hearing loss. That represents around 3.5% of the total population. In 2022, the World Health Organization estimates that there are 70 million deaf people worldwide. The visual aspect of signed languages is the primary emphasis of this study. In recent years, deep learning has had considerable success addressing particular AI issues. Google's speech recognition software is used in the majority of procedures for captioning YouTube videos. However, Google cannot process ASL video. This paper is building up the model that doing the video caption of the ASL videos and translating and then finish the captioning. The data videos are without captions. And the translations are stored as labels for the model to learn.

1. Introduction

The natural language of American Sign Language (ASL). In the USA and Canada, it is the main sign language used by the deaf and those who have hearing loss. This paper is aiming to captioning the ASL video images. Nowadays the captioning technology is evolving with the mass data. Translation of text to sign language is an important feature for people who need to get information from ASL. Implementing the feature to social media software, video website and other interacting platform may be very helpful for the people who in need. The goal of the computer science subfield of computer vision is to derive a more complex understanding from images and videos. The face authenticator on your mobile device, amusing video chat filters, and self-driving automobiles are all powered by this.

Recent advances in image-to-text creation, particularly the work by Donahue et al., have served as inspiration for our methodology, according to various research groups (2014). They used a subset of their model to generate text from video, but they refrained from suggesting a single network that ran end to end, opting to use an intermediary role representation instead. Additionally, they only displayed results for cooking movies, a limited range of predefined performers, and a narrow range of cooking-related objects. Video captioning is similar with the image captioning method, we need to adapt it for video sequences. Creating a natural language description from an input image is the aim of image captioning. For this objective, the encoder-decoder structure is frequently employed. Using a convolutional neural network, the image encoder (CNN). Our proposed approach has several important advantages over existing video description work. Real-world videos frequently have complicated dynamics, so techniques for creating open-domain video descriptions should be cognizant of temporal structure and allow for input in the form of a sequence of frames as well as output in the form of a sequence of words that can be of varying length. We suggest a novel end-to-end sequence-to-sequence paradigm to create subtitles for films as a solution to this issue. For this,

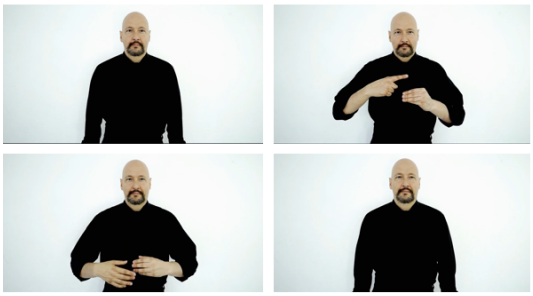

Figure 1. input videos

we use recurrent neural networks, more especially LSTMs, which have shown cutting-edge performance in the production of image captions. In order to produce a description of the event in the video clip, our LSTM model is trained on video-sentence pairs and learns to correlate a sequence of video frames to a sequence of words.

2. Related Work

The foundation of video understanding is the learning of video representation, which typically entails both feature extraction and aggregation. In order to create a compact representation, the ultimate goal is to extract information from many modalities and then combine them spatially and temporally. translate videos directly to sentences using a unified deep neural network with both convolutional and recurrent structure. [4]

The two most commonly used databases in recent years. MSR-VTT dataset: This dataset is the Microsoft Research - Video to Text (MSR-VTT) Challenge of ACM Multimedia 2016. The address is MSR-VTT dataset. The data set contains 10,000 video clips (video clips), which are divided into three parts: training, validation and test sets. Each video clip is annotated with about 20 English sentences. In addition, MSR-VTT also provides category information for each video (a total of 20 categories), which is a priori and known in the test set. At the same time, video contains audio information. The database uses a total of four machine translation evaluation indicators, namely: METEOR, BLEU@1-4, ROUGE-L, CIDEr. YouTube2Text dataset(or called MSVD dataset): This dataset is also provided by Microsoft Research, the address is MSVD dataset. The data set contains 1970 YouTube video clips (between 10-25s in length), and each video is labeled with about 40 English sentences. It can be seen that both databases are translations from trimmed video clips to sentences. The papers in the past two years basically use these two databases, indicating that the current research is still mainly focused on the translation of trimmed video clips to sentences.

mission critical point analysis The video captioning task can be understood as a seq2seq task from video image sequence to text sequence. In the methods in recent years, most articles use LSTM to construct the encoder-decoder structure, that is, use lstm encoder to encode the features of the video image sequence, and then use lstm decoder to decode the text information. Such a video captioning model structure was first proposed in the article "Sequence to Sequence – Video to Text" of ICCV2015[1]
Our system receives a brief video as input and produces a description of the primary activity in the video in natural language. Real-world films frequently contain complicated dynamics, so techniques for creating open-domain video descriptions should be cognizant of temporal structure and allow for input in the form of a frame-by-frame sequence as well as output in the form of a word-by-word sequence that can be of varying length. We suggest a brand-new end-to-end sequence-to-sequence model to produce subtitles for films as a method of solving this issue. Recurrent neural networks—more particularly are used for achieving the solution. LSTMs—which have proven to be quite effective in creating image captions. [3] A multiple parallel stream 2D CNN (two-dimensional convolution neural network) model is demonstrated to have high accuracy to 99.99 percentage for recognizing the hand locations. [2]

3. Methods

The work of creating video captions falls under the domains of Computer Vision (CV) and Natural Language Processing (NLP), where CV assesses the visual material and NLP transforms understanding into words in the appropriate order. In the recent years, it has been persuasively demonstrated that Convolutional Neural Networks (CNNs), which are employed for a range of vision tasks, can build a rich representation of the input image by embedding it into a fixed-length of vector.

Since we are studying the data-set describing the ASL. How we are capturing the feature of the motions? We need a confidence measure of the association for each pair of body part detect ions.

Multi-modal features are used in many articles as input features, i.e., how to extract the feature information from the video. primarily comprise the following:
1. based on video picture information utilizing action recognition models (like C3D) to extract video dynamic (spatial+temporal) characteristics, as well as merely using CNN (VGGNet, ResNet, etc.) to extract image (spatial) features.
2. Encode sound using BOAW (Bag-of-Audio-Words), FV (Fisher Vector), and other sound-based capabilities.
3. Past features: This feature might offer significant prior information, such as the video's category.
4. Text-based features: In this context, text-based features mean taking some text from the video and using it to create features for video captioning. I've observed two different kinds of these traits. One method is to first conduct image captioning on a single frame of video and utilize the output as the input feature for video captioning. The alternative is to tag videos and use the results as features. encoding-decoding architecture Even while lstm is used as the encoder-decoder in the majority of the work, there are still some variances in how each method is specifically configured.

3.1. Data Processing

Acquiring video data of 1702 items.Videos' length are between 1 second to 33 seconds. We have cleaned the dataset with reading the video in cv2 library to separate audio and frames from the video. From that process, we

have found 6 videos not readable and we have deleted the videos from the dataset.

Because ASL and English is not completely able to be translated word by word. There are phrases, expressions, and short sentences are the corresponding English captions. We are separating the video data-set into train data-set and validation data-set.

Table 1. Data set

Total	Train	Valid	Test
1696	1084	272	340

Unlike methods that merely use RGB images, the model was trained utilizing the low-cost depth datasets stated above. These datasets retained RGB images by extracting hands from the depth data sample using view and color information linked with depth data in order to provide richer information descriptions for the hand positions.

3.2. Label Cleaning

we have found the data name is linked with the video meaning, which should be the label that we are using to match with the video captioning result. After data cleaning, we have separated the label from the data name separately.

Table 2. Label Table

Id	Name	Label
0	BACK-[fingerspelled-version].mp4	back
1	what-DO$_{DO-DO-[lexicalized-fingerspelling].mp4}$	what do
2	10 dollars $_{1}0dollars.mp4$	10 dollars
3	25 cents $_{2}5-cents.mp4$	25 cents
4	3 o clock version.mp4	3 o clock version
...	...	...
1691	ZIG-ZAG–(stripes on an object).mp4	zig zag
1692	ZIP-LIPS-[zip-your-lips].mp4	zip lips
1693	ZIP-up-[general-version].mp4	zip up
1694	ZIP-up-[jacket version].mp4	zip up
1695	ZIP-[Z-I-P].mp4	zip

We have also separated the table data randomly into train, valid and test set. Meanwhile, we saved the data into new separated folders along with three new dataframe table with labels.

3.3. Natural Language Model

We are introducing the Word2Vec model in the file, This module implements the word2vec family of algorithms, using highly optimized C routines, data streaming and Pythonic interfaces.

We have train the model with all train listed words, and tested with the words in testing database.

The LSTM model we are building later is using this model as the base model for the translating.

Word2Vec(vocab=1044, vector_size=100, alpha=0.025) first 100 words : ['you', 'your', 'like', 'what', 'how', 'have', 'think', 'sign', 'go', 'many', 'to', 'name', 'class', 'want', 'live', 'up', 'version', 'eat', 'school', 'house', 'deaf', 'do', 'hair', 'before', 'suppose', 'this', 'favorite', 'look', 'city', 'here', 'people', 'me', 'in', 'finish', 'use', 'car', 'past', 'family', 'where', 'hearing', 'teacher', 'dad', 'take', 'learn', 'equal', 'much', 'prefer', 'self', 'year', 'cl', 'all', 'every', 'asl', 'with', 'book', 'night', 'can', 'shirt', 'feel', 'a', 'pizza', 'color', 'dog', 'doctor', 'why', 'movie', 'next', 'for', 'i', 'play', 'sister', 'children', 'law', 'work', 'need', 'earn', 'most', 'student', 'one', 'come', 'phone', 'money', 'backpack', 'tomorrow', 'or', 'yesterday', 'and', 'move', 'weekend', 'mom', 'sometimes', 'from', 'clothes', 'he', 'test', 'bedroom', 'college', 'wish', 'any', 'mind']

we have set all data into 32 frames data frames with 256 by 256 size. by checking with one table

Table 3. Label Table

Id	Number
Frame Per second	29.97002997002997
Total Frames	168.0
Height	1080.0
Width	1920.0

3.4. Model Building

Long Short-Term Memory Network (LSTM) and Convolutional Neural Network (CNN) will be used for Natural Language Processing and picture feature extraction, respectively (NLP).

Convolutional Neural Networks are customized deep neural networks that are capable of processing data with input shapes similar to a 2D matrix. CNN is particularly helpful when working with photos and can readily express images as a 2D matrix. CNN is primarily used to classify images and determine whether they are a bird, a jet,

Superman, etc. Images are scanned from top to bottom and left to right to extract key details, which are then combined to identify the images. It can handle images that have been resized, rotated, translated, and perspective-shifted.

Long short term memory, or LSTM[1], is a form of RNN (recurrent neural network) that is useful for solving issues involving sequence prediction. We can anticipate the following word based on the prior text. By getting over RNN's short-term memory restrictions, it has distinguished itself from regular RNN as an effective technology. Through the use of a forget gate, the LSTM may carry out relevant information while processing inputs and reject irrelevant information.

To generate the captioning part, we have learned from LSTM model and in order to create a fixed dimensional vector representation in this work, we first perform a feature transformation to an image. We are trying the convolution layers to extract the features from the video images. Then we are using Recurrent Neural Network (RNN) to generate the feed forward neural networks to sequence.

The attributes will be combined to forecast the caption's subsequent word. For images, CNN is utilized, and for text, LSTM. The performance of the trained model is measured using the BLEU Score.

Based on research by Donahue et al. (2014) that demonstrate two LSTM layers are superior to four and a single layer for picture to text tasks, we employ a two layer LSTM model to generate descriptions for movies.

Input features, or how to extract feature data from a video; multi-modal features are frequently utilized in articles. primarily comprise the following: based on video picture information utilizing action recognition models (like C3D) to extract video dynamic (spatial+temporal) characteristics, as well as merely using CNN (VGGNet, ResNet, etc.) to extract image (spatial) features.
Past features: This feature might offer significant prior information, such as the video's category.

Text-based features: In this context, text-based features mean taking some text from the video and using it to create features for video captioning. I've observed two different kinds of these traits. One method is to first conduct image captioning on a single frame of video and utilize the output as the input feature for video captioning. The alternative is to tag videos and use the results as features.

The next step in our pipeline is to construct a natural language description once we have obtained the individual action feature vectors and relationship feature vectors using the aforementioned methods.

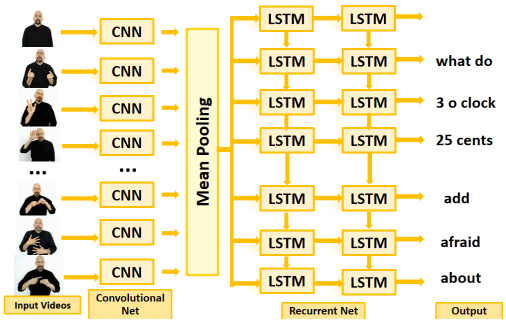

Figure 2. The network's structure for video descriptions. We gather the multiple features from the entire video and feed them into each frame as an individual feature pool. the LSTM network is time-stepped. Up until it selects the end-of-sentence tag, the LSTM outputs one word based on the video features (and the previous word) at each time step.

4. Results

It used to be quite difficult to caption videos, and it required a lot of model training and fine tuning. Additionally, a significant barrier to training video-related models is a shortage of good quality training data. Today, it is possible to fine-tune powerful open source models on a variety of tasks and obtain amazing performance on a small amount of processing resources thanks to large-scale pretrained transformer-based models and transfer learning.

Different approaches have been investigated based on the fundamental encoder-decoder structure to enhance the model effect in various areas, such as multi-modal features, multi-task learning, and so on. This discipline is currently going through a period of tremendous development, and new techniques are constantly being developed. The key to enhancing the model, in my opinion, is to find more efficient ways to extract the local semantic data from the video and incorporate it into natural language. However, I also believe that this path is one that has promise. The use cases for creating text descriptions for videos are numerous.

5. Conclusion

Traditional machine learning algorithms frequently struggle with the difficult task of annotating videos for the following reasons:

Since videos are far more difficult and expensive to gather and annotate than photos, and since captioning is a much more complex and ambiguous task for human annotators than classification, there are few datasets that have a large number of videos and supporting text captions.

Second, accurate modeling of each video frame's semantics for every conceivable type of video content is necessary for video captioning. It is nearly hard to produce any meaningful text without having a solid understanding of how video frames are put together.
Thirdly, since the best caption may entail combining data from various frames taken at various time intervals, video captioning calls for the system to keep track of long-term relationships. Finally, it can be difficult to generate conditional text in natural language. The model must produce language that sounds like human speech while simultaneously capturing the semantics of the video.

In this study, we introduced the unconstrained ASLing dataset, which was gathered in contexts where participants were free to dress in their typical daily attire. We concentrated on creating models to help us comprehend how to translate ASL with high quality using multi-feature models and without any gloss information. We displayed the attention weights based on the three fused features to better understand the inner workings of the models and discovered that the model dynamically learned and attended to each of these features depending on the input frame type.

References

[1] S. AYDIN, Ö. ÇAYLI, V. KILIÇ, and O. Aytuğ. Sequence-to-sequence video captioning with residual connected gated recurrent units. *Avrupa Bilim ve Teknoloji Dergisi*, (35):380–386. 2, 4

[2] I. Noreen, M. Hamid, U. Akram, S. Malik, and M. Saleem. Hand pose recognition using parallel multi stream cnn. *Sensors*, 21(24):8469, 2021. 2

[3] S. Venugopalan, M. Rohrbach, J. Donahue, R. Mooney, T. Darrell, and K. Saenko. Sequence to sequence-video to text. In *Proceedings of the IEEE international conference on computer vision*, pages 4534–4542, 2015. 2

[4] S. Venugopalan, H. Xu, J. Donahue, M. Rohrbach, R. Mooney, and K. Saenko. Translating videos to natural language using deep recurrent neural networks. *arXiv preprint arXiv:1412.4729*, 2014. 2