Automatic Timestamp Generation for YouTube

Aladdin Persson
Chalmers University of Technology
aladdin.persson@hotmail.com

Sanna Persson
Chalmers University of Technology
sanna.persson@live.com

Abstract

We present a comprehensive investigation into the problem of automatically generating timestamps for videos. To address this challenge, we propose and evaluate two different automatic pipelines for timestamp generation. Our first approach combines automatic speech recognition with state-of-the-art language models to generate suggestions for timestamps. The results of our study demonstrate that our pipeline generates adequate keywords for the video but the timing of the timestamps remains a challenge. Overall, our study offers a promising approach for automatically generating timestamps for videos, and in particular it highlights the potential for further work in this area. Our code for the project is available at: github.com/SannaPersson/Automatic-Youtube-Timestamps

1 Introduction

Long-format content becomes more common at the same time that attention spans become shorter. To navigate long videos on YouTube the solution commonly used is timestamping. Similar to how a book has chapters, they allow the user to watch what they find most interesting. In timestamps, the sections of the video are split with a time in the format (HH:MM:SS) and a keyword after, e.g. 0:00 Introduction. Timestamps are normally created manually by the creator or the audience in the comment section. This is, however, a tedious task where every part of the video must be watched carefully to identify the transitions between topics. YouTube has also recently released a beta version of automatic chapters that creators can enable as a feature. This feature still has some issues in generating adequate chapter titles with a reasonable length for each topic.

The problem of automatically generating timestamps for a video is a challenging task in the field of computer vision and natural language processing. It requires the use of sophisticated algorithms to accurately analyze the visual and audio content of a video, as well as the ability to understand and interpret the spoken or written language in the video.

One major challenge in this task is the variability of the visual and audio content. Videos can be filmed in different settings, with different lighting conditions, and with different types of speakers or actors. Furthermore, the language used in the video may vary in terms of accent, vocabulary, and grammar, making it difficult for a machine learning model to understand.

Another challenge is the need to accurately identify the start and end times of specific events or actions in the video. For a specific video, there is not one but many possibly useful and correct timestamp predictions and the one chosen by the creator reflects their style and bias.

In this paper, we investigate two different automatic pipelines for timestamp generation and discuss ways of approaching the problem. We have generated the domain to only include audio data since we believe that for most content it should be sufficient to determine the content.

2 Background

In our pipeline for automatic timestamps, we combine automatic speech recognition with state-of-the-art language models to obtain a suggestion for timestamps.

2.1 Whisper model

Whisper (Radford et al., 2022) is a recently released automatic speech recognition (ASR) system that was trained on 680,000 hours of multilingual and multitask data collected from the web. The system uses a simple end-to-end approach,

implemented as an encoder-decoder Transformer. Input audio is split into 30-second chunks, converted into a log-Mel spectrogram, and passed into an encoder. A decoder is then trained to predict the corresponding text caption. Whisper's use of a large and diverse dataset leads to improved robustness to accents, background noise, and technical language. Further, it enables transcription in multiple languages, as well as translation from those languages into English. The performance of Whisper is shown to be very robust and achieves state-of-the-art results in a zero-shot setting on many common benchmarks.

2.2 Text summarization

Text summarization is the process of automatically generating a summary of a given text document. This is a useful task in natural language processing as it allows for the efficient and effective processing of large amounts of text data.

There are two main types of text summarization methods: extractive and abstractive. Extractive methods involve selecting important sentences or phrases from the original text and concatenating them to form a summary. Abstractive methods, on the other hand, involve generating new text that expresses the main ideas of the original text in a condensed form.

The BART model (Lewis et al., 2019) is a popular text summarization model that falls under the category of abstractive methods. BART, which stands for "Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation and Comprehension", was developed by Facebook AI Research and is pre-trained on a large corpus of text data. It is a transformer-based model that generates a summary of the input text. The model is trained to reconstruct the original text from a corrupted version of it, where some words are randomly replaced with a special token.

BART has been found to be highly effective in generating high-quality summaries of text documents. It has been used in various natural language processing tasks, such as text summarization, machine translation, and dialog systems. The model is also included in the popular Hugging Face library, making it easily accessible to researchers and practitioners.

2.3 GPT-3

GPT-3 (Generative Pre-trained Transformer 3) (Brown et al., 2020) is a state-of-the-art language model developed by OpenAI. The model is pre-trained on a massive dataset of over 570GB of text, taken from the internet, books, and other sources. This pre-training allows the model to learn a wide range of language patterns and structures, which it can then apply when generating new text.

GPT-3 is based on the transformer architecture, which was introduced in the paper "Attention Is All You Need" (Vaswani et al., 2017). The transformer architecture uses self-attention mechanisms, which allow the model to weigh the importance of different parts of the input when making predictions.

During training, GPT-3 is presented with input and asked to predict the next word. To make this prediction, the model considers the entire input sequence and generates a probability distribution over the vocabulary for the next word. During inference, the model can be fine-tuned on a specific task, such as language translation or text summarization.

GPT-3 has been shown to excel at tasks such as language translation, question answering, and text summarization, many of them in a zero or few-shot setting. There are multiple versions of GPT-3 with the OpenAI API they can be finetuned on a custom dataset for more specific tasks.

2.4 Zero-shot classification

Zero-shot classification is a type of machine learning problem in which the model is tasked with classifying unseen or novel classes, without any training examples from those classes. The goal is to leverage prior knowledge about the classes, such as semantic attributes or class relationships, to generalize to new classes.

There are several ways to approach the problem of zero-shot classification, but one common method is to use a semantic embedding space, where each class is represented as a point in the space. The embedding space is learned using a large dataset of labeled examples from seen classes, such as ImageNet, and it is designed to capture the semantic relationships between classes.

Once the embedding space is learned, new classes can be represented by their semantic attributes, such as shape, size, and color, and these attributes can be used to predict the class of an unseen example. The prediction can be done by finding the class with the closest semantic embedding

to the example in the embedding space.

In summary, zero-shot classification is a machine learning problem where a model is trained to classify unseen classes, without any training examples from those classes. This is achieved by using semantic attributes or class relationships to generalize to new classes. The common method is to use a semantic embedding space, where each class is represented as a point in the space, and then use the semantic attributes of unseen examples to predict their class.

The BART model by Lewis et al. (2019) is a general language-understanding model which has also been trained for the task of zero-shot classification. The implementation used is provided by the HuggingFace library.

2.5 HuggingFace

The Hugging Face library is a Python library for natural language processing (NLP). It contains a wide range of pre-trained models that can be fine-tuned to specific tasks, such as text classification, named entity recognition, and machine translation. The library also includes tools for data preprocessing and evaluation, making it a comprehensive and user-friendly resource for NLP research and development.

One of the key features of the Hugging Face library is its use of transformers, a type of neural network architecture that has been shown to achieve state-of-the-art results on a wide range of NLP tasks. The transformer architecture, first introduced in the paper "Attention is All You Need" by (Vaswani et al., 2017), utilizes self-attention mechanisms to weigh the importance of different input tokens, allowing the model to effectively handle input sequences of varying lengths.

In addition to providing access to pre-trained models, the Hugging Face library also allows for fine-tuning of these models on specific datasets, making it a useful tool for transfer learning in NLP.

3 Method

We propose to build a pipeline for generating timestamps for YouTube videos automatically. The data will be transcripts from videos and the goal is to produce adequate times and chapter titles that could serve as timestamps of the video. We will obtain data by downloading transcripts through the YouTube API or producing them with the Whisper model. The labels will be timestamps scraped from YouTube descriptions. We have investigated using two different deep learning models for achieving this.

3.1 Data collection

The data was collected from YouTube from a list of approximately 30 creators who produce timestamps of the desired format. Both the audio and the description were downloaded and the timestamps were extracted from the description text. In total, a dataset of 2000 long-format (longer 25 minutes) videos were downloaded. The transcripts of the videos were either collected from YouTube or generated with the Whisper model.

The timestamps were normalized such that extra characters such as "-" were removed and that each timestamp had the format of "0:00 Introduction, 3:30 Interview". We further removed timestamps that did not describe the content and were instead clickbait for different parts in the videos.

3.2 Pipeline 1: Fine-tuning GPT-3

In the first pipeline the key idea is to extract summaries from audio transcripts and then let a fine-tuned GPT-3 model predict a suitable keyword for the summary which is a potential timestamp. The pipeline that we implemented is to:

- Summarize the transcript into a short summary for every 2-3 minutes using a Transformers BART model (Lewis et al., 2019)
- For each summary find the corresponding time stamp in the video
- Fine-tune OpenAI GPT Babbage (Brown et al., 2020) on predicting a list of chapter titles corresponding to each summary as well as identifying when several summaries have the same topic.
- Use a zero-shot classification BART (Lewis et al., 2019) to predict which chapter title that each sentence in the transcript belongs to.
- Smooth the chapter transitions such that each timestamp is at least for a certain time and there is no switching back and forth.

3.2.1 Fine-tuning settings

For the fine-tuning, we formatted our data such that the input was given as at most ten summaries. Each summary is produced by the BART model

(Lewis et al., 2019) of an audio transcript from a few minutes of a video. The label was the corresponding timestamp for each summary. In the case the summary corresponded to two different timestamps, only the first was used.

We fine-tuned the OpenAI GPT-3 Babbage model for four epochs on the custom dataset with the default settings for finetuning the model.

3.2.2 Zero-shot classification and post-processing

We used the HuggingFace implementation of the Bart model (Lewis et al., 2019) trained on the Multi-Genre Natural Language Inference dataset for the zero-short classification. For each sentence in the transcript, the model classifies it into one of two or three of the keywords of the predicted keywords by the fine-tuned GPT-3 model. The keywords were decided with a sliding window such that at each time in the transcript the classification model could choose between the current keyword and the adjacent ones. The reason for this decision was that having two many classes seemed to dilute the classification model's result. The main reason for using the classification model on each sentence was to find the timing of each timestamp transition.

The resulting keywords for each sentence were then post-processed according to a few simple rules. Each keyword must persist for at least 30 seconds to be valid and there can be no repeating keywords. In this step, we observe that many of the incorrect keywords are removed.

3.3 Pipeline 2: Whisper - you only listen once

In the second pipeline, we tried to make the most minimal pipeline possible to minimize information loss and identify limiting factors. In this approach, we let Whisper directly predict the timestamps from the audio clip. Whisper was trained on 30-second audio clips from the collected dataset and the corresponding label in this clip was the keywords for the timestamps and their corresponding duration.

We experimented with conditioning the model with the video title and previous timestamps but this is an area to investigate more in detail moving forward.

3.4 Evaluation

In line with previous generation models, the best evaluation is that of user experience. We have therefore provided a few example outputs of timestamps from the first pipeline in the Appendix. In the released code, it is also possible to try the inference function yourself.

For the first pipeline, the inference time is approximately 10 minutes for an hour-long video on a CPU. If the video does not have a transcript, inference time for the speech recognition model will be added to this.

4 Results

The results below concern Pipeline 1 which is described in detail in section 3.2. Due to the time constraints of the project, we were not able to finalize the results of the Whisper approach. We believe this is the most promising approach and discuss future improvements in section 5.2.

We have chosen to evaluate the pipeline by generating timestamps for example videos. In the Appendix A we see a few examples that are generated by the model, neither of these has timestamps already and they are not included in the dataset the GPT-3 model was fine-tuned on. The example videos are all within different categories with the common factor that the content can be discerned from the audio.

We observe that if we compare the generated timestamps with the video content the model is often quite correct in the keyword. The issue lies in the timing as well as the detail in the keyword distinction. For example, in educational content, the timestamps need more detail and domain knowledge to be useful to the audience.

5 Discussion

From our results, it seems that the first pipeline, section 3.2, can extract many seemingly relevant keywords for a multitude of video topics. The timestamps, however, lack the timing to be helpful to a user to navigate a long video. In some cases such as in the fourth example in Table A, the model repeats keywords or predicts way too long keywords.

5.1 Limitations

In the first pipeline, we observed that a major limitation was the zero-shot classification step. From one sentence it can be hard to predict what subject is being discussed and a possible area of improvement is to provide a context of the entire video.

The zero-shot classification is crucial for the timing of the timestamps and noisy classification can easily make relevant keywords into useless timestamps.

Another limitation was the variation of the data and we noted that certain timestamps, common for certain creators were over-represented in predictions on unseen content. One such example is that the Lex Fridman podcast usually includes the keyword "Advice for young people" which can be seen predicted in the fourth example in Table A.

We further regret that we were not able to investigate the second pipeline more in detail during the scope of this paper. The motivation for the second pipeline was that we noted that the first pipeline quickly abstracts a video in several 1-2 sentence summaries. In this process, the visual element is lost as well as the information in the audio of speakers, intonation and surroundings. We examined a sample of human performance of extracting keywords from an audio clip and it seems that a human without any particular domain knowledge can relatively accurately determine keywords from only a 30-second clip. We are therefore confident that a model trained on this task could also perform inference on such short clips.

When designing the first pipeline we made the decision to train on data which divides the video into longer chapters. This is a style choice that may affect the accuracy of the timestamps.

5.2 Future work

For future research, we see two different interesting approaches: combining whisper keyword extraction with GPT-3 language understanding and video summarization.

Our early results working on the second pipeline with Whisper keyword extraction from audio directly suggest that fine-tuning this model on a varied dataset could lead to generating adequate keywords directly. These keywords could be noisy and repeated which could then be summarized with a language model. Our experiments with GPT-3 (Brown et al., 2020) suggest that such a model would be able to extract the most important keywords and their times from a long list of timestamps.

Another important aspect is that we chose to look at the audio of a video which we believe is sufficient for podcast-like content. For videos where the visual element is dominant instead, our idea will face some challenges. In this case, a video captioning model would likely be required to obtain natural language descriptions of the content from which keywords could be extracted.

6 Conclusion

In this project, we have experienced that automatic timestamp prediction is a challenging task. We limited our domain to audio and text and obtained reasonable keywords that correctly described the audio content. The main obstacle we identified was generating adequate timing of the timestamps while still producing useful timestamps.

References

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. https://doi.org/10.48550/arXiv.2005.14165 Language Models are Few-Shot Learners. ArXiv:2005.14165 [cs].

HuggingFace. https://huggingface.co/facebook/bart-large-mnli facebook/bart-large-mnli · Hugging Face.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. https://doi.org/10.48550/arXiv.1910.13461 BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. ArXiv:1910.13461 [cs, stat].

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2022. https://doi.org/10.48550/arXiv.2212.04356 Robust Speech Recognition via Large-Scale Weak Supervision. ArXiv:2212.04356 [cs, eess].

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. https://doi.org/10.48550/arXiv.1706.03762 Attention Is All You Need. ArXiv:1706.03762 [cs].

A Generation results - Pipeline 1

Youtube link and title	Predicted timestamps
Machine Learning for Natural Language Processing: Transformer Language Models https://www.youtube.com/watch?v=hcU62ItWKA0	0:00:00: What is a language model? 0:00:51: Transformers architecture 0:05:51: How to train the model? 0:09:35: Free and open source models
Joe Rogan Experience #1169 - Elon Musk https://www.youtube.com/watch?v=ycPr5-27vSI	0:00:00: The Hyperloop 0:00:32: Elon Musk and AI 0:09:56: Barack Obama's views on AI 0:30:47: Artificial intelligence 0:34:57: Extending the human timeline in the universe 0:43:38: Tesla game 0:58:13: Flat Earth theory 1:08:56: Flying cars 1:12:08: Electric cars vs gasoline powered ones 1:18:14: Justice system 1:34:50: Worst person in the world scenario 1:40:50: Texas 1:43:09: Huawei 1:47:13: AI and society 1:53:00: Tesla, SpaceX, Uber, etc 1:53:49: Solar panels 2:01:10: Charging system 2:05:31: Case for the Tesla Model 3 2:15:19: Social media 2:36:26: Elon Musk and Steve Jobs
Deepica Mutyala — The New Standards of Beauty — Talks at Google https://www.youtube.com/watch?v=mfeynfAY9kw	0:00:00: Intro 0:01:29: Red Lipstick & Holiday Celebrations 0:02:17: Deepika's Story 0:05:21: Social Media Insecurities 0:12:10: Beauty & Identity, 10 Years Later 0:21:24: Funding Deepika's New Brand 0:29:07: Advice for Young People 0:33:30: How do you normally plan your day? 0:38:46: Book that has been very helpful to you: "Burn Rate" by Andy Dunn, "The Truth About Men" by Shailja Gollapunjia, "The Truth About Men" by Shailja Gollapunjia, "Burn Rate" by Andy Dunn, "The Truth About Men" by Shailja Gollapunjia 0:45:22: Book that has been very helpful to you: "Burn Rate" by Andy Dunn, "The Truth About Men" by Shailja Gollapunjia, "Burn Rate" by Andy Dunn, "The Truth About Men" by Shailja Gollapunjia 0:47:30: Deepika's new beauty brand, Live Tinted 0:48:36: What is the most memorable moment of The Today Show

The Liver King Lie https://www.youtube.com/watch?v=BKpmAGZQetc	0:00:00: Intro 0:00:35: Truths About Liverkings Content & Brand 0:04:24: Steroids Or Supplements? 0:08:13: How To Spot a Pseudo-Primal 0:11:04: The Liver King's 9 Ancestral Tenet 0:17:06: Brian Johnson & Liver King 0:27:30: Email 1 - January 2021 0:33:57: Email 2 - January 2021 0:34:38: Email 3 - February 2021 0:35:37: Brian's protocol for boosting GH & IGF-1 0:37:04: Closing Thoughts 0:43:50: Current Diet & Carnivore Style 0:52:22: Brian Johnson's HGH Story
Pro-Grade Acoustic Treatment https://www.youtube.com/watch?v=bB4ihxp5ALY	0:00:00: Acoustic Treatment 0:00:44: Diffusion absorbers 0:03:29: Measuring the room 0:07:30: Final thoughts and comments