A Pipeline for the Creation of Multimodal Corpora from YouTube Videos

Nathan Dykes
Friedrich-Alexander-Universität Erlangen-Nürnberg
nathan.dykes@fau.de

Anna Wilson
University of Oxford
anna.wilson@area.ox.ac.uk

Peter Uhrig
ScaDS.AI Dresden/Leipzig
TU Dresden
peter.uhrig@tu-dresden.de

Abstract

This paper introduces an open-source pipeline for the creation of multimodal corpora from YouTube videos. It minimizes storage and bandwidth requirements, because the videos themselves need not be downloaded and can remain on YouTube's servers. It also minimizes processing requirements by using YouTube's automatically generated subtitles, thus avoiding a computationally expensive automatic speech recognition processing step. The pipeline combines standard tools and provides as its output a corpus file in the industry-standard vertical format used by many corpus managers. It is straightforwardly extensible with the addition of further levels of annotation and can be adapted to languages other than English.

1 Introduction

The analysis of multimodal communication has become mainstream in linguistic research in the past few decades, which results in a higher demand for multimodal corpus resources of ever-increasing size for more and more languages and varieties. While there are very good reasons for the manual creation of multimodal corpora when specific varieties are needed that usually occur beyond the public sphere, these approaches do not scale well due to the prohibitive cost of manual data collection, transcription and, possibly, annotation.

In corpus linguistics, a common approach for written corpora is using existing publications, often newspapers and other periodicals, or crawling web pages and social media. This is also possible for multimodal corpora, as illustrated by the NewsScape English Corpus (Uhrig, 2018, 2022), which is based on American TV News collected by the NewsScape project at UCLA and the related processing tools developed in the context of the Distributed Little Red Hen Lab (see e.g. Steen et al. (2018)). However, the processing pipeline is highly adapted to the peculiarities of the data, in particular the TV subtitles and metadata recorded, so it does not generalize well to other domains/datasets.

YouTube is a very interesting source for multimodal corpora for several reasons. One is the sheer number of videos hosted on the platform, and another is its breadth, which ranges from professionally produced and edited programs provided by broadcasters and other media outlets, via a variety of content created by more or less professional YouTubers, to content that bears witness to the relatively anarchic nature of the platform. Thus, YouTube is a treasure trove for the creators of multimodal corpora, who can select the videos they deem most representative of the language or variety they wish to study.

In this paper we introduce a processing pipeline for the creation of multimodal corpora from YouTube videos, making use of the automatically-generated subtitles provided by YouTube. We combine existing processing tools into a usable pipeline that needs as its input a set of YouTube URLs and provides as its output a corpus that can be imported directly into CQPweb, an open-source corpus manager (Hardie, 2012).

2 YouTube Captions as Corpus Data

As mentioned above, one of the most time-consuming and thus most expensive steps in the creation of multimodal corpora is the transcription of the spoken text. In TV broadcasts, subtitles are often created by humans, increasingly supported by automatic speech recognition (ASR) technology. YouTube allows content creators to provide their own subtitles to go with the videos, and some large broadcasters systematically provide the subtitles they broadcast for the YouTube recordings of the same program. However, measured by the scale of YouTube's size, this is a minuscule proportion of videos and, again, does not scale well. We will ignore these types of subtitles in the present pipeline and instead focus on YouTube's automatically gen-

Proceedings of the 1st Workshop on Linguistic Insights from and for Multimodal Language Processing, pages 1–5
September 22, 2023. ©2023 Association for Computational Linguistics

erated captions.

YouTube's automatic captioning system makes use of ASR to provide subtitles on videos in the following languages: Arabic, Dutch, English, French, German, Hindi, Indonesian, Italian, Japanese, Korean, Polish, Portuguese, Romanian, Russian, Spanish, Thai, Turkish, Ukrainian, and Vietnamese.[1] If a video is detected to be in one of these languages, YouTube will create automatic subtitles, which can be displayed on the video once the ASR process has finished. Content creators have to actively disable this if they do not want their video to be captioned, so most videos come with automatic captions. One of the major advantages of automatic captions compared to manually created captions as found on TV is that YouTube's captions come with relatively accurate timing information on the word level (if the right format is used – see next section) while the manual subtitles are usually presented line by line and tend to lag behind, especially on content that is (or was originally) broadcast live.

2.1 Downloads and Format(s)

YouTube downloads are a tricky business. Generally, YouTube as a for-profit company generating revenue through advertisement views has little interest in allowing bulk downloading of their data. On the other hand, there are legitimate uses of YouTube downloads that the open source community provides software for, which needs regular updates to keep up with the constant changes introduced by YouTube. In the first versions of the pipeline presented here, youtube-dl[2] was used to download the closed captions and write metadata files. The current version uses yt-dlp[3], which markets itself as "A youtube-dl fork with additional features and fixes". By default, youtube-dl and yt-dlp save downloaded files with the video title as the file name. Given that YouTube videos can contain almost arbitrary characters, not all of which are supported by all file systems, and given that video titles need not be unique, we use YouTube's 11-character video ID as the filename for the download and in all further processing.

YouTube stores its videos and subtitles in a variety of formats to provide the appropriate quality and formats depending on factors such as playback device, screen resolution/window size, and Internet connection speed. Audio and Video formats are not of interest for the purpose of the present paper, but the subtitle formats are. Some formats, for instance the popular SubRip format (.srt), only have line-level timing information and are thus not ideal for multimodal corpus building, because the corpus becomes more useful when every word has timing information associated with it. For this reason, the present pipeline uses the WebVTT format (.vtt), which at the time of implementation was the only format providing word-level timing information and a rough indication of ASR confidence encoded via the text color.[4]

In addition to the subtitles, our pipeline uses yt-dlp to download the info json file, which contains metadata about the video, e.g. upload date, uploader and channel, which are included in the corpora created.

2.2 Accuracy

To the best of our knowledge, YouTube does not publish statistics on the accuracy of the closed captions. Not surprisingly, the results are directly related to the quality of the audio signal, which is best in studio recordings of professional speakers of the standard language. This is in line with YouTube's own statement that "automatic captions might misrepresent the spoken content due to mispronunciations, accents, dialects, or background noise."[5] Furthermore, manual inspection showed that the reliability is severely reduced in languages such as Russian (where morphological forms are often incorrectly rendered even if the lemma is correctly recognized) or Turkish, where we see high error rates on the admittedly small samples tested. We assume that future versions of YouTube's captioning system will be based on Google's recent Universal Speech Model (Zhang et al., 2023), which should improve accuracy in lesser-resourced languages (and possibly add support for a much wider variety of languages).

3 NLP pipeline

Our pipeline is available for download at `https://github.com/RedHenLab/youtube_pipeline`. The various processing steps and their corresponding input and output data formats are given as an

[1] https://support.google.com/youtube/answer/6373554
[2] https://youtube-dl.org/
[3] https://github.com/yt-dlp/yt-dlp

[4] YouTube has since removed the text coloring from WebVTT subtitles and introduced the json3 format, which provides more fine-grained information on the ASR confidence. A version of our pipeline with json3 support will be made available by the start of KONVENS.
[5] https://support.google.com/youtube/answer/6373554

overview in Table 1. In principle, it is possible to add extensions or replace individual components of the pipeline at any given processing step as long as input and output formats remain intact.

3.1 Tokenisation

As YouTube provides the WebVTT format with word-level timing information, we have a type of implicit ("whitespace") tokenization to begin with (see however below), which might already be sufficient for certain applications. However, because our pipeline includes PoS tagging and syntactic parsing, we need to tokenize further to ensure compatibility with the downstream tools. For English, the vast majority of cases requiring additional tokenization can be solved with a regular expression that splits up contractions (*'s|'ve|'re|'d* etc.) before the apostrophe. In our tests, this approach was sufficient for more than 99% of videos. However, with larger corpora, the tokenization became increasingly challenging as several kinds of rare exceptions had to be addressed. Firstly, despite the captions usually appearing with no punctuation, individual files did occasionally contain punctuation marks which had most likely been introduced by manual modifications carried out by the content creator. Secondly, although typically each word is assigned a separate start time, some common expressions are treated as multi-word units, which means that they are displayed to the viewer as a chunk and thus have the same start timestamp (e.g. some instances of *a lot* or *a little*, repeated fillers like *uh hu* etc.). Thirdly, defaulting to setting token boundaries at common contraction or genitive markers occasionally produces errors. For instance, one of our videos contains the compound *bird's-eye-view*, where this ad-hoc tokenization would have produced the obviously nonsensical tokens *bird* and *'s-eye-view*. For these reasons, a more elaborate tokenization was necessary, for which we use SoMaJo (Proisl and Uhrig, 2016)[6] during our first processing step, where the text is converted to the CoNLL-U format that stores each token with the associated timestamps. Each token that is affected in this step is assigned to the same timestamps as the one original token in the .vtt file.

3.2 Punctuation Restoration

As mentioned in the section on tokenization, the automatic captions usually do not contain punctuation marks. This is problematic for NLP processing since the identification of phrase and sentence boundaries relies on this information. Standard NLP tools are trained on text with punctuation so that the accuracy of PoS tagging is reduced without it and syntactic parsing becomes downright impossible without sentence boundaries, which are typically derived from punctuation information. Not to mention the poor readability for researchers analyzing data without punctuation. It was therefore necessary to automatically insert punctuation marks in plausible positions. Fortunately, there are off-the-shelf solutions to this exact problem. We chose Alam et al. (2020)'s tool due to the promising results on different languages, and its rather straightforward usability out of the box.[7]

In its original version, this tool treats commas, colons and dashes as commas; and full stops, exclamation marks and semicolons as full stops. We fine-tuned the tool on the Brown Corpus family with slight tweaks to the original scripts, in order to also insert exclamation marks and dashes as separate categories, which we expect to be useful for analyses interested in fine-grained interactional phenomena. Given suitable training data, the process can easily be adapted to other languages. In this step, we also insert explicit sentence boundaries as a prerequisite for syntactic parsing.

3.3 Tagging, Parsing and Corpus Construction

In order to prepare the data for tagging, the punctuated text files are aligned with their original CONLL versions that contain the timestamp information. Newly inserted punctuation marks receive the same timestamp as the last token for which timing information is available. The data is then annotated for PoS, lemma and other morpho-syntactic features with UDPipe 1 (Straka et al., 2016), which was selected because it supports a large number of the languages for which YouTube provides automatic captions. Since we use standard CoNLL-U files as input and output, it is comparably easy to plug in a different library if needed.

4 CQPweb

The tagged and parsed files are then converted to vertical text files (.vrt), which is the standard input format for the Corpus Workbench (Evert and

[6] Although SoMaJo was only developed for English and German, it has been successfully applied to other languages.

[7] The original tool can be found at https://github.com/xashru/punctuation-restoration. Our pipeline uses a fork of this repository that is linked in the README.

Processing Step	Input Data	Output Data
YouTube download	Text file with YouTube URLs	WebVTT subtitles and info-JSON metadata file
Subtitle extraction and tokenization	WebVTT subtitles	CoNLL-U input for NLP
Raw Text extraction	CoNLL-U	plain text
Punctuation restoration	plain text	plain text with punctuation marks and sentence boundaries
Merging punctuation restoration results	CoNLL-U and plain text with punctuation marks and sentence boundaries	CoNLL-U
NLP with UDPipe	CoNLL-U	CoNLL-U
creation of corpus files	CoNLL-U and info-JSON metadata file	vertical file for each video
corpus aggregation	vertical files for each video	one vertical file for the entire corpus

Table 1: Overview of processing steps with input and output data

Hardie, 2011) and, by extension, CQPweb (Hardie, 2012), which we currently use to conduct our analyses. In this step of the pipeline, the annotated files are combined with relevant metadata from the info-JSON files associated with each video. Currently, we extract information on the uploader, the channel, the video title, the upload date, and the duration in seconds. Timestamps are added in separate columns so that we can jump directly to the right position in the video for every word in the corpus.

CQPweb is a browser-based frontend to the Corpus Workbench. As compared to other readily available corpus tools, CQPweb has several advantages which make it particularly suitable for our research endeavours. Firstly, it allows for very flexible queries combining arbitrary levels of annotation; thus allowing us e.g. to search for combinations of linguistic and gestural features. Secondly, its core functionality can be enhanced through custom plugins and visualizations, which we use to link to the YouTube videos in the right position.

5 Conclusion

The pipeline we presented here enables corpus linguists to create multimodal corpora from YouTube in a straightforward way. The user needs to provide a text file with YouTube links, which can be links to individual videos or to entire YouTube channels, which will then be downloaded. After the download, all successfully retrieved subtitle files will be processed by the NLP pipeline, which will output a single .vrt file and an accompanying list of attributes for import into CQPweb. In addition, due to the open and simple formats used, the pipeline can be extended with further annotation levels, e.g. based on automatic prosodic or computer vision analysis, which can be added as extra columns in the vertical file. Together with the custom visualization for video playback and the download plugin provided for CQPweb, a fully functional multimodal corpus is at the linguist's fingertips.[8]

Limitations

The full pipeline presented in this paper is currently only available for auto-generated subtitles in English, but an earlier (and simpler) multilingual pipeline (whitespace tokenization, no punctuation restoration, briefly presented in Uhrig (2022)) has been successfully applied to a Russian-language YouTube dataset.

Ethics Statement

Researchers using our pipeline are faced with three ethics questions. The first concerns their relationship to the video producer and the people recorded in the video. Are any personal rights violated by including the video in question in a corpus? The second question is in their relationship to the legal requirements and codes of conduct when collecting data, e.g. questions of copyright, where there are exemptions for academic research in many but

[8] See Uhrig et al. (2023) for the use of this pipeline in a larger research project and its application in a case study.

not all jurisdictions. The third is the relationship between the researcher and YouTube as the content provider, whose terms and conditions may restrict certain types of automated downloads in certain jurisdictions. Researchers are solely responsible for their own use of this pipeline.

Acknowledgements

The research presented in this paper was made possible by generous funding provided by the Deutsche Forschungsgemeinschaft (project number 468466485) and the Arts and Humanities Research Council (grant reference AH/W010720/1) to the second and the last author. The authors gratefully acknowledge the scientific support and HPC resources provided by the Erlangen National High Performance Computing Center (NHR@FAU) of the Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU) under the NHR project b105dc to the second author. NHR funding is provided by federal and Bavarian state authorities. NHR@FAU hardware is partially funded by the German Research Foundation (DFG) – 440719683. The authors also gratefully acknowledge funding by the Defence Science and Technology Laboratory, Ministry of Defence, awarded to a project led by the second author.

References

Tanvirul Alam, Akib Khan, and Firoj Alam. 2020. Punctuation restoration using transformer models for high- and low-resource languages. In *Proceedings of the Sixth Workshop on Noisy User-generated Text (W-NUT 2020)*, pages 132–142.

Stefan Evert and Andrew Hardie. 2011. Twenty-first century corpus workbench: Updating a query architecture for the new millennium. In *Proceedings of the Corpus Linguistics 2011 conference*.

Andrew Hardie. 2012. Cqpweb—combining power, flexibility and usability in a corpus analysis tool. *International journal of corpus linguistics*, 17(3):380–409.

Thomas Proisl and Peter Uhrig. 2016. Somajo: State-of-the-art tokenization for german web and social media texts. In *Proceedings of the 10th Web as Corpus Workshop*, pages 57–62.

Francis F. Steen, Anders Hougaard, Jungseock Joo, Inés Olza, Cristóbal Pagán Cánovas, Anna Pleshakova, Soumya Ray, Peter Uhrig, Javier Valenzuela, Jacek Woźny, and Mark Turner. 2018. Toward an infrastructure for data-driven multimodal communication research. *Linguistics Vanguard*, 4(1).

Milan Straka, Jan Hajic, and Jana Straková. 2016. Udpipe: trainable pipeline for processing conll-u files performing tokenization, morphological analysis, pos tagging and parsing. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 4290–4297.

Peter Uhrig. 2018. NewsScape and the Distributed Little Red Hen Lab – A digital infrastructure for the large-scale analysis of TV broadcasts. In *Anglistentag 2017 in Regensburg: Proceedings. Proceedings of the Conference of the German Association of University Teachers of English*, pages 99–114, Trier. Wissenschaftlicher Verlag Trier.

Peter Uhrig. 2022. *Large-Scale Multimodal Corpus Linguistics – The Big Data Turn*. Habilitation thesis, Friedrich Alexander Universität Erlangen-Nürnberg.

Peter Uhrig, Elinor Payne, Irina Pavlova, Ilya Burenko, Nathan Dykes, Mary Baltazani, Evie Burrows, Scott Hale, Philip Torr, and Anna Wilson. 2023. Studying time conceptualisation via speech, prosody, and hand gesture: Interweaving manual and computational methods of analysis. In *Proceedings of the 8th Gesture and Speech in Interaction Conference*.

Yu Zhang, Wei Han, James Qin, Yongqiang Wang, Ankur Bapna, Zhehuai Chen, Nanxin Chen, Bo Li, Vera Axelrod, Gary Wang, Zhong Meng, Ke Hu, Andrew Rosenberg, Rohit Prabhavalkar, Daniel S. Park, Parisa Haghani, Jason Riesa, Ginger Perng, Hagen Soltau, Trevor Strohman, Bhuvana Ramabhadran, Tara Sainath, Pedro Moreno, Chung-Cheng Chiu, Johan Schalkwyk, Françoise Beaufays, and Yonghui Wu. 2023. Google usm: Scaling automatic speech recognition beyond 100 languages.