Audio to Tree Alignment for Puebla-Nahuatl

Ashish Patidar
apatida@iu.edu

Abstract

This paper discusses the method of aligning audio data to tree alignment by utilizing ELAN files and conllu files. The paper suggests converting the audio data into a time-stamped transcript in the form of an ELAN file, and then converting the transcript into conllu files in order to create dependency trees without timestamps. However, these files can be difficult to work with due to missing values for some fields, which can make it challenging to develop a consistent alignment algorithm.To overcome this issue, the paper proposes a model for assigning timestamps to dependency trees in an efficient and precise manner. This approach can help researchers align audio data with tree alignment more accurately, allowing for better analysis of Puebla Nahuatl language.

1 Introduction

Audio-to-text alignment is the process of synchronizing a spoken audio recording with a written transcript of the audio. This can be useful for various applications, such as speech recognition or language translation. In order to align an audio recording with a transcript, the audio and transcript are first divided into smaller segments, typically corresponding to individual words or phrases. These segments are then aligned with each other, matching the words or phrases in the transcript with their corresponding audio segments. This alignment process typically uses ASR and NLP techniques to match the audio segments with their corresponding transcript segments accurately.

So instead of doing force alignment of audio to a transcript, we can use ELAN files, ELAN files are a type of file used to store annotations for audio and video data. An ELAN file consists of a number of different elements, including the audio or video data being annotated, the transcript of the audio or video, and the annotations themselves. These annotations can include time-aligned transcriptions of the audio or video, as well as linguistic or other types of annotations.

Sentences are converted to dependency trees using online tools and are stored in CoNLL files. CoNLL files are a specific type of file used to store dependency trees. These trees show the grammatical relationships between words in a sentence and can be useful for analyzing and understanding the structure of a sentence. Conllu files are typically used in natural language processing (NLP) tasks, such as part-of-speech tagging or syntactic parsing.

Aligning ELAN files with CoNLL files can be a challenging task, as the two file formats may use different conventions for encoding the annotations and storing the data. As a result, aligning ELAN and CoNLL files typically involves a process of mapping the annotations in one file to the corresponding text in the other file, using a combination of manual and automated methods.

Once the ELAN and CoNLL files have been aligned, the resulting data can be used for a variety of purposes, such as training machine learning models for natural language processing tasks, or for conducting linguistic research

2 Dataset Description

Openslr.org is a dedicated site for hosting speech recognition and translation data. The Puebla Nahuatl speech dataset is collected from openslr.org, which includes the transcriber files, audio files, metadata, and time aligned ELAN-files.

Transcriber files are transcript files and ELAN files are generated from audio files and transcriber files. To achieve goal of this project we will use ELAN files from the data set only.

3 Literature Review

To align audio to transcript, the most common technique is to use force alignment. Forced alignment is a very straightforward technique which could be implemented using any of the two algorithms: 1) DTW(Dynamic Time Wrapping) 2) Automatic Speech Recognition via HMM(Hidden Markov Models) Both algorithms have different approaches and are widely used for audio-to-transcript matching.

The DTW algorithm matches two temporal sequences with different speeds, but with certain rules. An optimal match is found when all the rules are followed and have a minimum cost. The nearest-neighbor classifier can be used to find the cost or distance. ASR via HMM this method uses a direct method of recognizing the words and their start and end time in the audio, this can be implemented with the help of hidden Markov models. The recent development in this domain has suggested that using HMM's for audio to text gives comparably good results with a small machine wear-off compared to using DTW for the same.

Several toolkits are available to automatically synchronize audio to text and three most commonly used toolkits are: 1) Aneas: Aneas toolkit is used for quick and efficient audio to text alignment at sentence-level, apart from alignment aneas can be used to extract alignment information; but aneas is not designed in a way to work with non-perfect synced audio and text. 2) Gentle: Gentle toolkit can be used for word level alignment as well as aneas toolkit, but it is comparatively slower than Aneas. One major constraint in using gentle is that it only works with English pronunciation dictionaries which limits its usage in NLP domain. 3) Montreal Forced Alignement: MFA is the most commonly used toolkit, as it can perform sentence-level as well as word-level alignment and works fine with languages other than English. Also, MFA can handle non-perfect synced audio and text in a decent manner.

Apart from audio to text alignment, audio annotation is also an important part in improving searchability, organization, and accessibility of audio files. Audio annotation is the process of adding descriptive information to an audio file. This could include details about the content of the audio, such as the topic, key words, or a transcription of the spoken words. It could also include information about the speaker, such as their name, gender, or accent. ELAN files are the audio annotated files which may contain timestamps of all the sentences.

4 Methodology and Algorithm

Since, the available data already contains time-aligned audio annotation ELAN files I will utilise them to get the name of audio files and time stamp for all the sentences in CoNLL testing file.
These CoNLL and ELAN files contains text in Puebla Nahuatl as well as translated text in spanish and either of them could be used to perform sentence level alignment between both the files.
The pympi library in python is used to read and access ELAN files, and conllu library is used to read the CoNLL test file. In CoNLL file many labels can be inserted as comments with the dependency tree of the text like text itself, sent_id, labels etc. Similarly, for ELAN files there can be many labels and for the dataset selected ELAN files are pretty consistent with the fields in them, but the CoNLL file is not clean, and hence, it is impossible to create a generalised algorithm to align available ELAN file and CoNLL file.
For aligning the ELAN file with CoNLL file, I suggest a method to approach in three steps:

4.1 Annotation Number and File Name

Some of the sent_id in the CoNLL files contains the name of EAF file alongwith annotation number at the end, and annotation numbers are also mentioned as labels for the sentences. I will use the speach library in python to extract text with file name, and time stamps from ELAN file and then match the same with file name and annotation number extracted from the sent_id of the sentence in CoNLL file.
In CoNLL files some dependency trees are created by merging two or more different sentences with different annotation number. For this type of sentences, I am repeating the sentence twice and with different annotation number and extracting time for all the annotation numbers. After extracting time stamps of all the annotation numbers the sen-

tence consist of the start time of first annotation number is selected and end time corresponding to last annotation number is selected.

4.2 Sentence Merging and Matching

For the dependency trees that do not have file name and annotation number in sent_id of the sentence, I tried to match the sentences. For matching sentences, all the sentences with their time stamps, audio file names and .eaf file name for all the available ELAN files are collected in a dataframe and all the CoNLL sentence with their reference number(sometimes annotation number) are stored in other dataframe. The texts that are aligned in first step are already removed from both the dataframes to save some computing power. Matching texts is not as straightforward as matching their annotation number as texts with same annotation number can be divided in various parts to create several dependency trees from them and to match these texts I first merged all the parts of the sentences with same reference number and sent_id and added new column to CoNLL files dataframe. Now, all the parts of a sentence will have same merged sentence which is used for aligning with the ELAN file. Once, the sentences are matched in both the dataframes time stamps of matched sentences will be extracted along with audio file name and sentence itself.
For the dependency trees made up of multiple unit sentences and annotation number cannot be identified, I tried to align unit sentences and get there timestamp and later on aggregating timestamp as start time for the first sentence start time and end time as end time of last sentence in the dependency tree.

4.3 Matching and Merging for Translated Sentences

The sentences that were matched in above step will be removed from both the dataframe containing ELAN file data and CoNLL file data. Now for remaining sentences same method is used to match spanish translation available in both the files, this is done if some sentences are not matched in second step because of missing a punctuation or a character in either of the files. There is still a chance that some sentences will still not be aligned if both files are not cleaned and preprocessed properly.

These three steps can be generalised for the files with same conventions and to use it for files with different conventions or fields with some small changes in the code, as most of the methods are generalised.

5 Evaluation

For aligning sentences between two files, there are not many evaluation parameters which can be used. But the most obvious parameter is how many sentences in test files are aligned with ELAN files. One more parameter could be to check if the alignment done is correct or incorrect, but for a basic model like this it is very difficult to check the accuracy.
Hence, our evaluation parameter is number of sentences aligned/total number of sentences in CoNLL file.

6 Conclusion and Results

Aligning ELAN files with CoNLL file is always an arduous process. While some libraries in python makes it easy to extract data from these files but aligning them is difficult as both file system follows two totally different conventions. Developing a generalized method to align them is next to impossible because it is not always possible to have values for all the fields in CoNLL file for all the sentences. This is because for large datasets a CoNLL is file is aggregated work of different persons and all of them follows different format.
The experiment we performed on CoNLL file allowed us to align 810 sentences out of 1227 present in the file. This is a very basic model built on the available parameters from ELAN files and CoNLL files, which would be definitely improved in future.

7 Limitations and Future Scope

First limitation is that this method is not generic although l the methods used are generic but conditions of alignment can be different based on the fields of both the files. Other limitation is some sentences/slangs are repeated many times and for such sentences whose file name and annotation numbers cannot be extracted it is difficult to say if the sentences are accurately aligned.

This model is basic model for sentence to sentence alignment, but in future this model can be used for doing this alignment on word level and extracting time stamp for each word in the sentence. A word level dependency tree would be

very beneficial for research purposes. And, also from the limitations it is clear that their is lot of scope of improvement in this model.

References

[1] Xavier Anguera, Jordi Luque, and Ciro Gracia, *Audio-to-text alignment for speech recognition with very limited resources*

[2] Jian Zhu, Cong Zhang, David Jurgens, *Phone to Audio alignment without text: A semi-supervised approach*

[3] Federico Simonetta, Stavros Ntalampiras, Federico Avanzini, *Audio-to-Score Alignment Using Deep Automatic Music Transcription*

[4] Anirban DUTTA, Gudmalwar ASHISH KUMAR, Ch V Rama RAO, *Performance analysis of ASR system in hybrid DNN-HMM framework using a PWL euclidean activation function*

[5] Peter Wittenburg, Hennie Brugman, Albert Russel, Alex Klassmann, Han Sloetjes , *ELAN: a Professional Framework for Multimodality Research*

[6] E. Auer1, A. Russel, H. Sloetjes, P. Wittenburg, O. Schreer, S. Masnieri, D. Schneider, S. Tschöpel, *ELAN as Flexible Annotation Framework for Sound and Image Processing Detectors*

[7] Jiatong Shi, Jonathan D. Amith, Xuankai Chang, Siddharth Dalmia, Brian Yan, and Shinji Watanabe, *Highland Puebla Nahuatl-Spanish Speech Translation Corpus for Endangered Language Documentation*

[8] Jiahong Yuan, Wei Lai, Chris Cieri, and Mark Liberman, *Using Forced Alignment for Phonetics Research*

[9] Kalinda Pride, Nichol as Tomlin, Scott AnderBois, *LingView: A Web Interface for Viewing FLEx and ELAN Files*

[10] *https://github.com/dopefishh/pympi*

[11] *https://github.com/pyconll/pyconll*

[12] *https://github.com/EmilStenstrom/conllu*

[13] *https://github.com/neocl/speach*