



Università
della
Svizzera
italiana

Faculty
of
Informatics

Bachelor Thesis

June 14, 2019

SubThis!

A Web-Based Application to support the TED Translators Initiative

Gregory Wullimann

Abstract

TED is a global community devoted to spreading ideas, usually in the form of short, powerful talks known as "TED Talks." TED began in 1984 as a conference where Technology, Entertainment and Design converged. Today TED covers almost all topics—from science to business to global issues—in 116 languages. Meanwhile, independently run TEDx events to help share ideas in local communities around the world. TED Talks are viewed 100 million times every month, 1.2 billion times a year. This means more than 3 millions views per day.

Different languages represent a barrier for the TED mission. In 2009 passionate viewers around the world started to translate TED talks to share them with friends and family. Recognizing a real need, TED started the TED Translators program to allow volunteers to translate talks into any language. TED Translators are volunteers who subtitle TED Talks, and enable the inspiring ideas in them to overcome languages and borders. As of today, the community counts more than 30 thousand translators and more than 140 thousand translations in more than 100 languages, including Klingon.

Subtitling a talk is a non trivial and very time consuming task. Until now, this task has been carried out almost entirely without the support of tools. In the last years, the Italian TED Translators community—and in particular the Teams of TEDxLakeComo, TEDxVarese, and TEDxCesena—adopted a series of open-source tools to reduce the time needed to generate high quality subtitles. However these tools are disconnected between themselves and still require significant manual intervention and computer science expertise. The aim of this project is to collaborate with the TED Translators initiative to provide a unified and effective toolchain to generate subtitles for TED and TEDx Talks.

Currently in order to manually create a subtitle an expert translator needs at least 4 hours, with SubThis! we aim to reduce the process to 30-60 minutes.

Advisor:

Prof. Dr. Michele Lanza

Co-advisor:

Dr. Roberto Minelli

Approved by the advisor on Date: 14.06.2019

Contents

1	Introduction	3
1.1	State of the art	3
2	The Subtitling Process	4
2.1	Transcription	4
2.2	Segmentation	5
2.3	Forced alignment	5
3	Approach	7
3.1	User Interface	7
3.1.1	Adding a media	7
3.1.2	Creating a subtitle	7
3.1.3	Main UI of SubThis!	7
3.1.4	Notification System	9
3.2	Architecture	9
3.3	Database	11
3.4	Folder structure	12
3.5	Automatic Transcription	12
3.6	Automatic Segmentation	13
3.7	Forced alignment	13
4	Summary	14
4.1	Contributions	14
4.2	Future work	14
4.3	Conclusion	14

List of Figures

1	Steps to create a subtitle	3
2	The Amara subtitle editor	4
3	Subtitles states	4
4	Example of transcription	5
5	Example of segmentation	5
6	Example of forced alignment	6
7	Creation of a subtitle	7
8	Main User Interface of SubThis!	9
9	Notification example	9
10	Project architecture	10
11	Sequence diagram	10
12	ER Diagram	11
13	Laravel file structure	12
14	Example of Speech Matics transcription using Harvard Sentences	13
15	Precision of segmentation by fragments based on input length	13
16	Precision of segmentation by lines based on input length	13

1 Introduction

TED is a community that spreads ideas around the world with short talks, known as "TED Talks."¹ In 1984 TED started as conference where Technology, Entertainment and Design converged. Nowadays the TED Talks cover almost any topics in more than 100 languages. With the large success of TED other independent events, known as TEDx, started in communities around the world. Translating all talks in more than 100 languages is a challenge. In 2009 viewers started to translate TED talks in order share them with anyone. TED developed a system to allow collaboration between volunteers to translate talks into any language. Today the community of TED Translators counts more than 30 thousand volunteers that make subtitles in more than 100 languages. Generating high quality subtitles is a nontrivial and very time consuming task. This process is mostly done manually and on average an experienced TED Translators takes about 4 to 5 hours. In recent years, some Italian TEDx teams, started to use some open-source tools to reduce the time needed to generate high quality subtitles. The problem with these tools is that they are not connected between them and are not easily accessible to non tech savvy people. The process of generating a subtitle can be decomposed in 3 phases. The transcription is the first one, given an audio (or video) file we obtain the plain text containing what has been said in the file. Once we have the plain text, we start the segmentation process. The segmentation divide the plain text into fragments by respecting a set of guidelines², these fragments will be the part of text that will be displayed on the video. The final step is called forced alignment, or synchronization. With this step we create a mapping between fragments and timestamps in the audio file. When the forced alignment is done we obtain the final subtitle file. These processes are currently done manually, but the vast majority of the process can be made in an autonomous way, with only a manual quality check needed at the end. The objective of this project is to create a web platform which allows translators to easily generate and edit subtitles without having to install separated tools.

Figure 1 summarizes the steps to create a subtitle: Transcription (Section 2.1), Segmentation (Section 2.2), Forced alignment (Section 2.3), and the clean up.

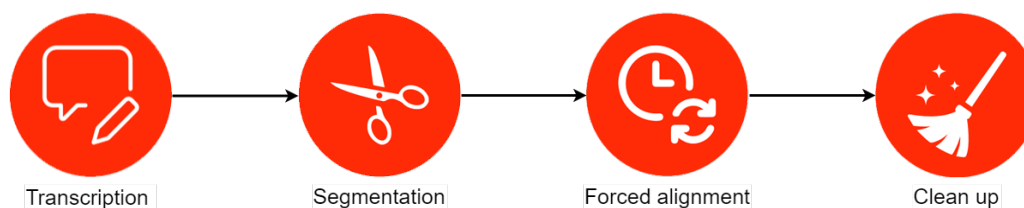


Figure 1. Steps to create a subtitle

We can resume in short the contributions of our project as:

- Partially automatic generation of high quality subtitles, which require low manual intervention. This process includes the automatic transcription, automatic segmentation, and the forced alignment.
- A user interface for tools needed by the community of TED Translators.
- A user interface for editing subtitles.

1.1 State of the art

Currently there are various tools, such as Amara,³ ReadBeyond's Lachesis,⁴ Aeneas,⁵ and many speech-to-text services, that some translators uses in order to facilitate their job. The problem with these tools is that they do not offer an user interface and must be used via a command line, which can be difficult to use for non tech savvy people.

Existing platforms such as Amara, depicted in Figure 2, offer some tools to help the translators. Amara is used by thousands of TED Translators, and not only, around the world but it does not offer more than a glorified text editor. These tools require a lot of manual intervention and they are not enough to drastically reduce the time needed to generate a subtitle.

¹<https://www.ted.com/talks>

²<https://www.ted.com/participate/translate/guidelines>

³<https://amara.org>

⁴<https://github.com/readbeyond/lachesis>

⁵<https://www.readbeyond.it/aeneas/>

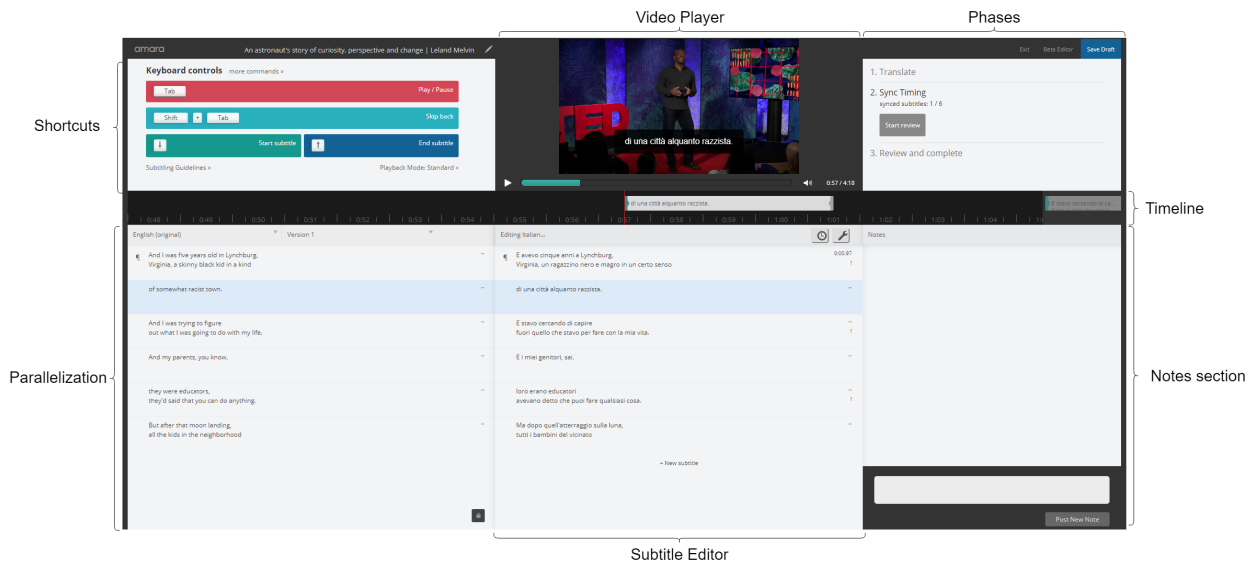


Figure 2. The Amara subtitle editor

The transcription can be made with one of the many available services, such as Google Cloud Speech-To-Text,⁶ IBM Watson Speech to Text,⁷ or Speech Matics.⁸

Other tools for segmentation and forced alignment exist, examples include ReadBeyond's Lachesis and Aeneas, but they are not accessible to users without experience in programming.

The project aims to combine many of these existing tool in an centralized easy to use web platform.

2 The Subtitling Process

Creating a subtitle is not a trivial task. In this section we will explain the three main steps required and how much time it takes for an experienced translator to create a subtitle starting from a video of interest. Starting with the an audio (or video) file we get the transcribed text with the transcription process, then with the segmentation we obtain the segmented text, which are the subtitles line that will be displayed on the video, and then with the forced alignment we obtain the final subtitles text. This steps are depicted in Figure 3.

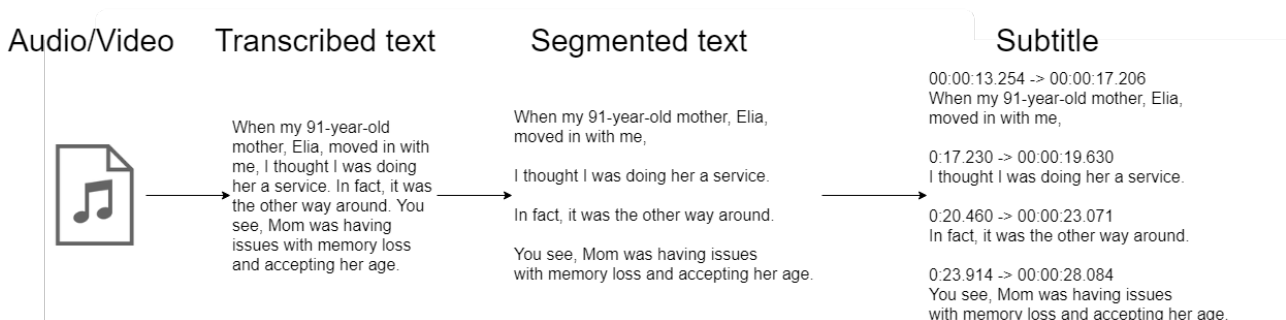


Figure 3. Subtitles states

2.1 Transcription

The creation of a subtitle starts with an audio (or video) file. This first step is to write what has been said in the audio, this process generates a plain text. This process, called transcription, is currently done manually, a translator takes about 2 hours to do it.

⁶<https://cloud.google.com/speech-to-text/>

⁷<https://www.ibm.com/watson/services/speech-to-text/>

⁸<https://www.speechmatics.com/>

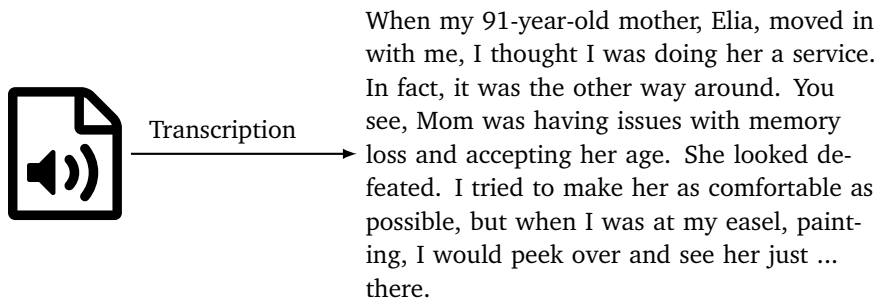


Figure 4. Example of transcription

2.2 Segmentation

Once we have transcribed the text of the video (or audio) file, the next step is to divide it into fragments, that will be displayed on the video when playing. This process is called segmentation.

Fragments must meet the guidelines imposed by TED, some of the main rules are:

- When a subtitle is longer than 42 characters, break it into two lines.
- Never use more than two lines per subtitle.
- Keep broken lines as close in length as possible.
- Keep 'linguistic wholes' together when breaking lines.
- Keep the subtitle reading speed at a maximum of 21 characters/second.
- Compress subtitles over 21 characters/second. Try to preserve as much meaning as possible.

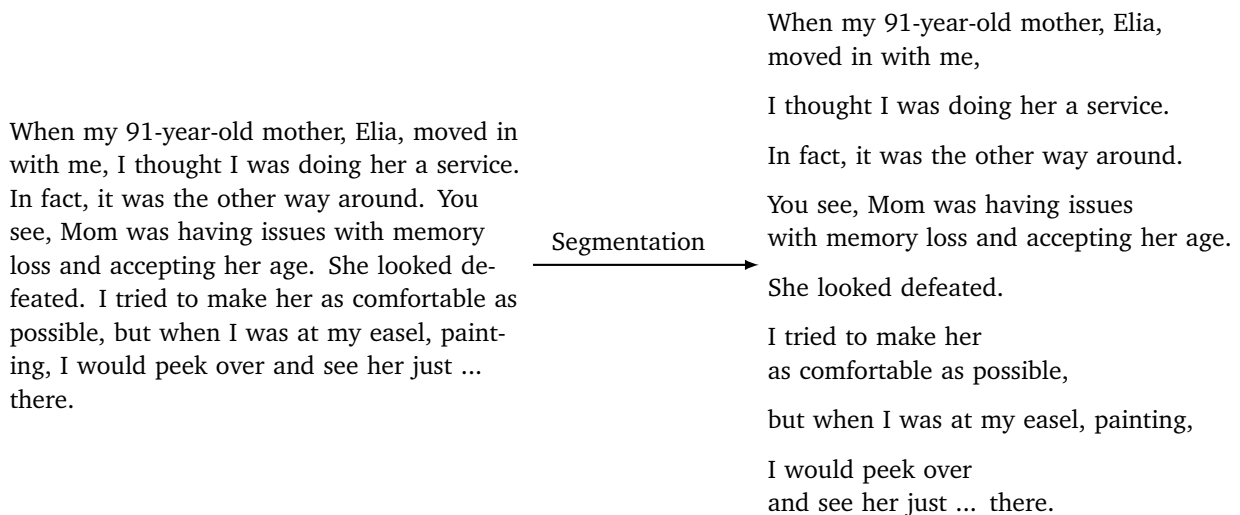


Figure 5. Example of segmentation

2.3 Forced alignment

The final step of creating a subtitle is called forced alignment, or text synchronization. This process maps text fragments (i.e., subtitle lines) to specific timestamps in an audio file.

When my 91-year-old mother, Elia,
moved in with me,
I thought I was doing her a service.
In fact, it was the other way around.
You see, Mom was having issues
with memory loss and accepting her age.
She looked defeated.
I tried to make her
as comfortable as possible,
but when I was at my easel, painting,
I would peek over
and see her just ... there.

Forced alignment
→

00:00:13.254 → 00:00:17.206
When my 91-year-old mother, Elia,
moved in with me,
00:00:17.230 → 00:00:19.630
I thought I was doing her a service.
00:00:20.460 → 00:00:23.071
In fact, it was the other way around.
00:00:23.914 → 00:00:28.084
You see, Mom was having issues
with memory loss and accepting her age.
00:00:29.128 → 00:00:31.016
She looked defeated.
00:00:31.914 → 00:00:35.329
I tried to make her
as comfortable as possible,
00:00:35.353 → 00:00:37.749
but when I was at my easel, painting,
00:00:37.773 → 00:00:41.368
I would peek over
and see her just ... there.

Figure 6. Example of forced alignment

3 Approach

The objective of our project is to create an interface to create and modify subtitles. We developed a web platform which provides a simple and unified interface for the existing tools, complementing them with novel features requested by members of the Italian TEDx community.

3.1 User Interface

The objective of the platform is to minimize the work load for the user, providing quick feedback on the quality of the subtitle. The platform combined with the tools for the transcription, segmentation, and synchronization should drastically reduce the time the user has to work on the subtitle, while originating high quality subtitles. Users will have to register in order to use the platform, in this way each user can control her own media and subtitles.

3.1.1 Adding a media

We defined a **media** as either a video or audio resource. A new media can be created using the following parameters:

- Title
- Description
- Either a file or a link to a media (e.g., a YouTube video)

3.1.2 Creating a subtitle

Each media can have one or more subtitles, with the idea that every subtitle is for a different language. In Figure 7 we can see the form for creating a subtitle. The user has different option to choose from when creating a subtitle. Depending on the option chosen other fields are shown, for example when selecting "Subtitle" a field to upload the subtitle files will be shown. Giving the option to upload different files formats users can use the platform in different ways, for example by only using the segmentation and forced alignment function but using another editor. In the example depicted in Figure 7 the user selected the option "Transcribe text from audio". This means that upon clicking on the "Add" button our platform will generate the plain text containing what has been said in the audio.

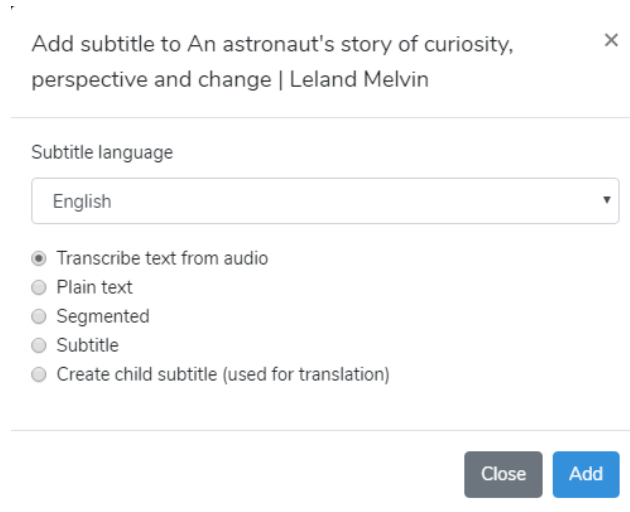


Figure 7. Creation of a subtitle

3.1.3 Main UI of SubThis!

The Main UI of SubThis!, depicted in Figure 8, is made of 4 main parts: the header, the video, the timeline, and the actual subtitle editor.

Header

The header contains the metadata about the active media, such as title and description. From the header is possible to perform various action, shown in Figure 8.A:

- Create a new subtitle

- Select a subtitle
- Start synchronization or segmentation (depending on the state of the subtitle, the states can be seen in Figure 3)
- Enable translation mode
- Download the subtitle in various format
- Delete the subtitle
- Manage settings about the editor
- Manage configuration of the subtitles (such as maximum number of characters per line)

Video Player

Below the header we have the video player, which is implemented using VideoJS.⁹ The video player will show the current subtitles.

Timeline

The timeline shows the audio waveform and all the fragments. The waveform is needed to manually adjust the fragments. The elements of the timeline can be moved and resized to correct the timing of the fragment, the fragments that do not respect the guidelines are colored in red, example shown in Figure 8.B, whereas the correct one are colored in green.

Editor

The last part shows the editor, where each row is a text fragment, i.e, subtitle line. Each fragment can be edited and the user receives instant feedback on the correctness of the fragment. If a row is blue it means that is the current fragment (see example depicted in Figure 8.C). If a row is red it means that the fragment does not follow the guidelines and more information about the errors can be found on the right side, Figure 8.D, for example, shows a subtitle line whose 'line ratio' is 0.25, which is not in line with TED Guidelines. We used alternating row colors, white and grey, to show correct fragments.

Right-clicking on a fragment, will pop-up a contextual menu where users can use functions such as adding a new fragment or duplicating it, an example can be found on Figure 8.E.

⁹<https://videojs.com/>

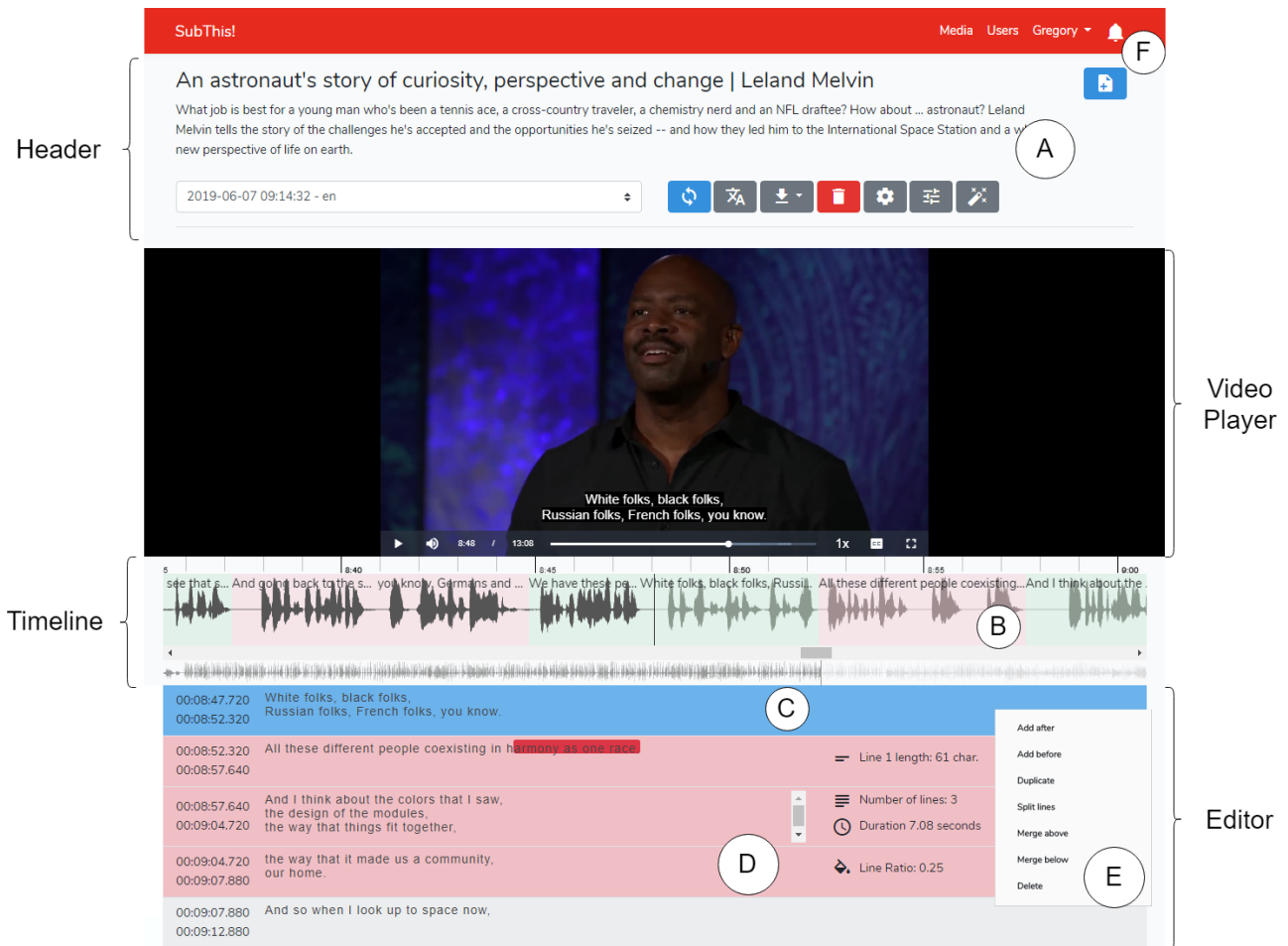


Figure 8. Main User Interface of SubThis!

3.1.4 Notification System

All the steps needed to create a subtitle (e.g., segmentation and forced alignment) can take several minutes to complete. So we implemented a notification system to inform the user when a process is finished. The notification menu can be opened by clicking the icon shown in Figure 8.F

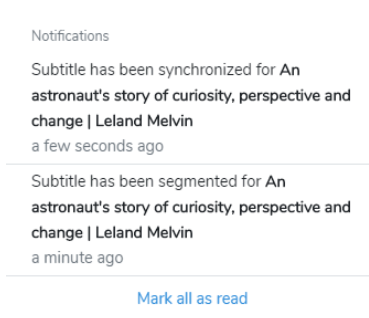


Figure 9. Notification example

3.2 Architecture

The general architecture of the project is illustrated in Figure 10. The user interface was made with Bootstrap,¹⁰ which provide a CSS toolkit to develop quickly and easily user interfaces, we decided to use it to save time instead of creating the user interface from scratch. To handle the data reactively in the front-end we used Vue.js,¹¹ which is a javascript front-end framework, the way VueJS handles components made the work easier and more ordered when

¹⁰<https://getbootstrap.com/>

¹¹<https://vuejs.org/>

working with the front-end. The back-end framework that we used is Laravel,¹² with this framework we mostly handle the API for the web interface. The choice for Laravel was a matter of personal preference, Laravel offers a well ordered directory structure which will make the work easier in the future. Other than the API, Laravel manages the 3 main Python applications used for the segmentation, forced alignment, and transcription. These 3 Python applications are used since the tools we use offer libraries in Python.

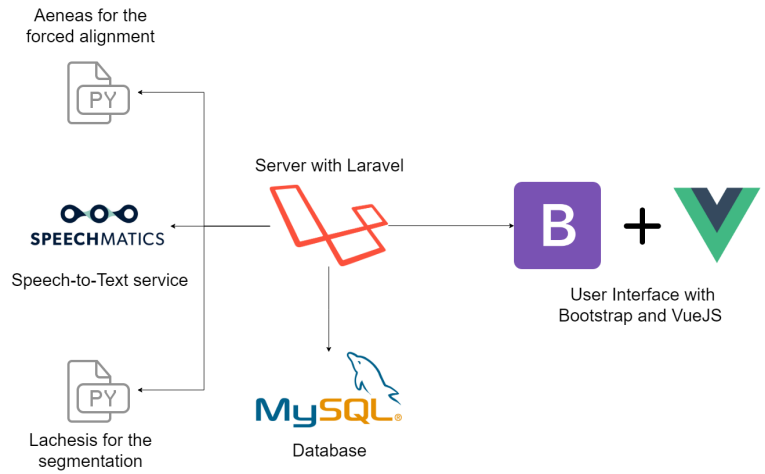


Figure 10. Project architecture

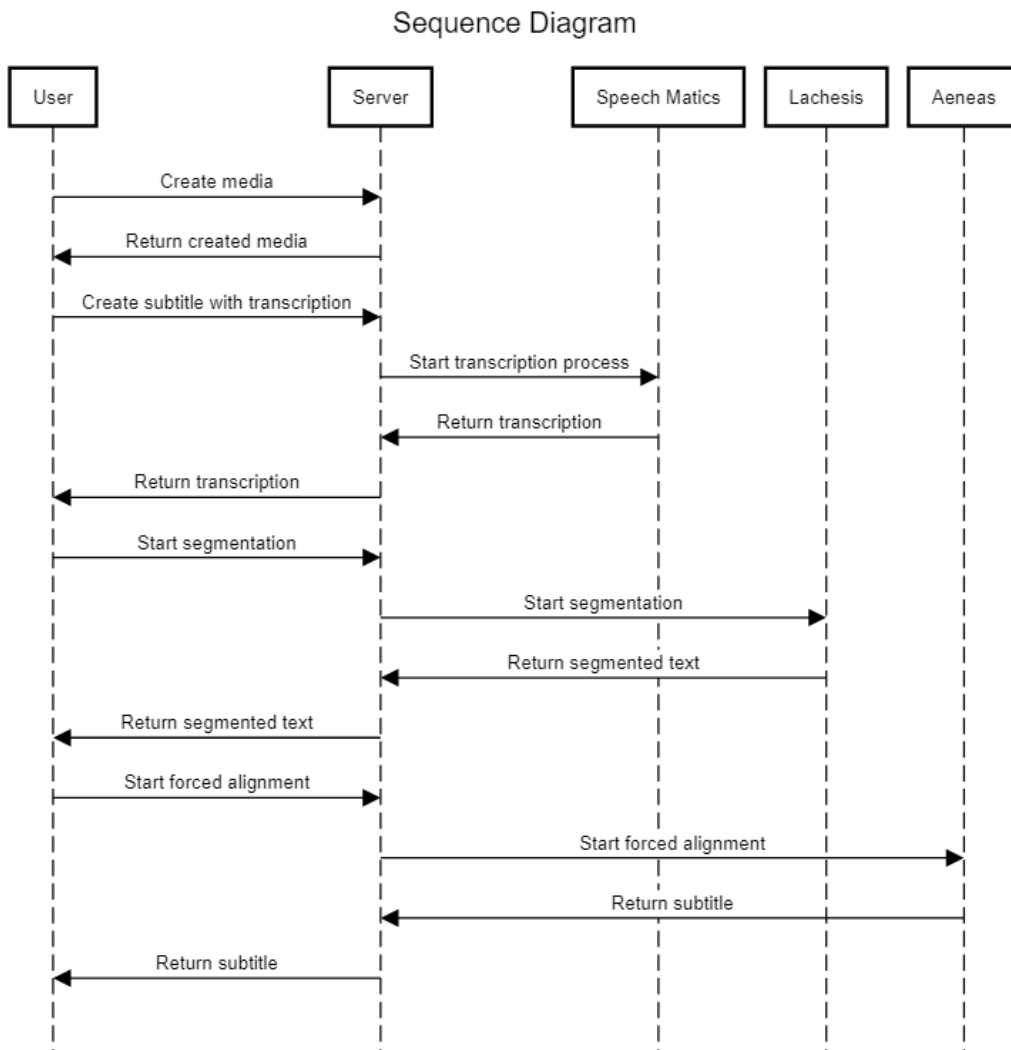


Figure 11. Sequence diagram

¹²<https://laravel.com/>

3.3 Database

One of the requirements was to give users the possibility to register and login and to also to give the possibility to halt the subtitling process and resume it whenever they want without losing any progress. So we needed a place where to store the information about users, media and the subtitles. For this reason, we store all the information about the users and the media in a MySQL database. In Figure 12 we can see the ER Diagram of the database.



Figure 12. ER Diagram

Users

This table stores users' information, such as name, email, and password. The role attribute is used to distinguish users with admin access (value set to 0, and standard user, value set to 1). In this way is also easy to add other roles by using different numbers for different roles. For example, we foresee the figure of the "TED Language Coordinator," the only person who can approve a subtitle created by a TED Translator.

Invites

To maintain the quality of subtitles high and to limit the number of users on the platform, we decided to allow users to register only under invitation. This table is used to store the invited users. When a user will try to register the web application a invite code will be asked.

Password resets

This table is used for storing the reset token when a user forget her password.

Notifications

This table store the users' notifications, which will be explained in Section 3.1.4.

Media

The Media table store information about the media, such as title, description, and language. Each media belongs to a user.

Subtitles

The subtitles table with the media table are the most important ones. The subtitles table belongs to a media and a user. The three attributes `plain_text`, `segmented`, and `subtitles` store the text version of the subtitle depending on which status it is.

The other attributes such as `max_lines`, `max_chars`, etc. are used to check the correctness of the subtitles within the editor, they are needed since the controls change depending on the language of the subtitle.

Jobs

The various processes such as segmentation and forced alignment work with a queue, or jobs, system since they need some time to process the data. This table is needed to manage the queue.

3.4 Folder structure

The folder structure follows the standard structure that Laravel uses.¹³ An important folder is the resources folder. Here we have `js` folder which contains the VueJS base files and the components files used to build the web application. The `scripts` folder contains the files used for the segmentation, synchronization, and speech-to-text. In the `NPL/tmp` folder we have the models and subtitles files used to train the machine learning.

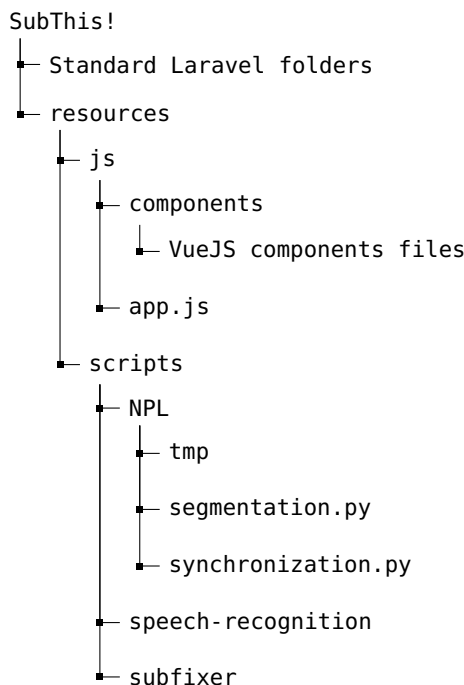


Figure 13. Laravel file structure

3.5 Automatic Transcription

One of the most useful features is the possibility to automatically transcribe the audio to text. The automatic transcription is implemented using the service Speech Matics, which is a paid service. We tried other free alternative such as CMUSphinx [3]¹⁴ but the results were not good enough and the process was slow. However, if the whole Italian TEDx community will use the tool, we estimated they would only incur in a cost of USD 500 per year but they will be saving thousands of working hours.

With Speech Matics we are able to get the transcription of a 10 minutes audio in less than 2 minutes. If we factor in the time to revise and correct the transcription, this step should require less than 30 minutes to complete.

¹³<https://laravel.com/docs/5.8/structure>

¹⁴<https://cmusphinx.github.io/>

The birch canoe slid on the smooth planks.
 Glue the sheet to the dark blue background.
 It's easy to tell the depth of a well.
 These days a chicken leg is a rare dish.
 Rice is often served in round bowls.
 The juice of lemons makes fine punch.
 The box was thrown beside the parked truck.
 The hogs were fed chopped corn and
 garbage.
 Four hours of steady work faced us.
 Large size in stockings is hard to sell.

Speech Matics

The birch canoes slid on the smooth planks
 glue the sheet to the dark blue background.
 It is easy to tell the depth of a well these
 days a chicken leg is a rare dish. Rice is
 often served in round bowls the juice of
 lemons makes fine punch the box was down
 beside the parked truck. The hogs are fed
 chopped corn and garbage. Four hours of
 steady work face to us a large thighs and
 stockings is hard to sell.

Figure 14. Example of Speech Matics transcription using Harvard Sentences¹⁵

3.6 Automatic Segmentation

Segmentation has been implemented with ReadBeyond's Lachesis. Lachesis combines machine learning techniques like Conditional Random Fields (CRF) [2] and Natural Language Processing (NLP) [4][1] tools like Part-Of-Speech (POS) [5] tagging and sentence segmentation to split the text into closed caption lines.

We decided to use UDPipe [6]¹⁶ to perform POS tagging since it supports more languages than the others available.

Lachesis offer a simple way to build machine learning models based on our input to perform the segmentation. We built a model by analyzing 1,000 subtitles taken from TEDTalks' YouTube Channel,¹⁷ which are high quality human controlled subtitles. In this way the machine learning model is able to perform segmentation based on what it learned from the given subtitles. Another advantage of using machine learning is that it is easy to introduce new language by simply creating new models.

The segmentation process can be improved. Preserving linguistic wholes is hard when breaking lines while respecting the TED guidelines. In Figure 15 we can see that the precision of correct fragments is about 40% over a set of 1,000 subtitles. In Figure 16, similarly to the previous figure, we can see the precision line by line is about 50%.

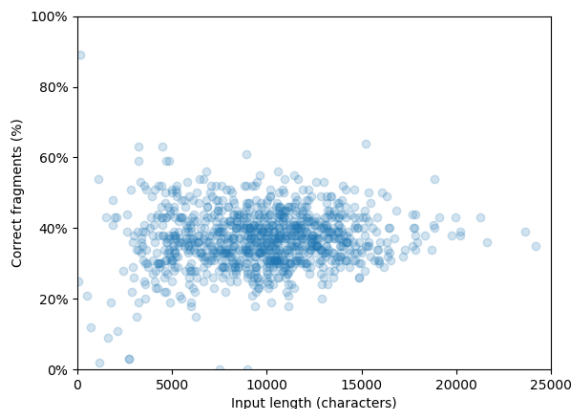


Figure 15. Precision of segmentation by fragments based on input length

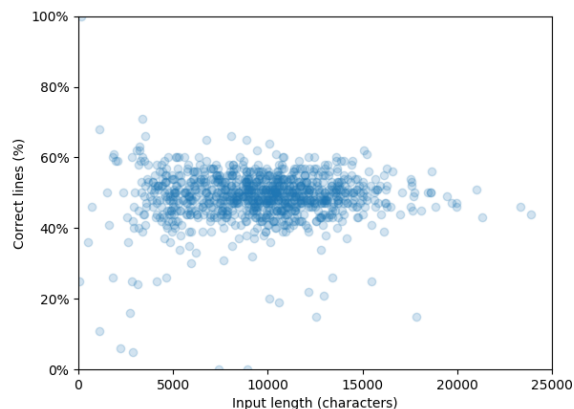


Figure 16. Precision of segmentation by lines based on input length

3.7 Forced alignment

Forced alignment has been implemented with ReadBeyond's Aeneas.

Aeneas automatically generates a synchronization map between a list of text fragments and an audio file that contains the narration of the same text.

¹⁵https://en.wikipedia.org/wiki/Harvard_sentences

¹⁶<https://ufal.mff.cuni.cz/udpipe>

¹⁷<https://www.youtube.com/channel/UCAuUUnT6oDeKwE6v1NGQxug>

4 Summary

4.1 Contributions

The contributions of this project can be summarized as:

- Partially automatic generation of high quality subtitles, which require low manual intervention. This process includes the automatic transcription, automatic segmentation, and the forced alignment.
- A user interface for tools needed by the community of TED Translators.
- A user interface for editing subtitles.

4.2 Future work

The web platform is ready to use and functional, but a number of changes could improve the user experience.

- Collaborations: The platform allows only the user that created the subtitles to modify it. A system to create groups of users would facilitate the collaboration between users.
- Subtitle editor: The current editor is rich of functionality, but it lacks the support of customizable short-cuts, which would make the users more efficient.
- Segmentation: As seen in the Section 3.6 the segmentation process is still imprecise, an important step to improve the tool would be to improve the algorithm of the segmentation, since it is still require heavy manual intervention. A possible improvement would be using subsegmenter¹⁸ developed by Federico Sangati.
- Integration with Amara: Amara has a system that allows users to have their subtitles reviewed by other users and the publish them directly on Youtube. Since Amara provied an API it could be possible to create the subtitles on SubThis! and then review them on Amara.

4.3 Conclusion

The TED community spreads their ideas with short talks known as "TED Talks." This talks cover almost all topics and are translated in more than 100 languages. In order to maintain the talks available in all the world a community of volunteer was born, the TED Translators initiative. The TED Translators initiative counts more than 30 thousand volunteer which translate every day subtitles, most of them without the help of any tools. Translating and subtitling subtitles is important to the TED and TEDx communities since they help spreading their talks and ideas. We have seen that creating a subtitle is not a trivial task and it requires effort and time. Each step is done manually but it can partially automated to speed up the whole process. With SubThis! we have created a platform which combine existing tools in a simple user interface which allows translators to quickly and easily create high quality subtitles. We have seen that the automatic transcription will greatly reduce the working time at a low cost and with good results, the automatic segmentation has space for improvement but in the current state will already save time to the translators and the forced alignment combined with the subtitle editor will give the users a high quality subtitle in a short time. From preliminary evidence, using our platform should drastically reduce the time needed to create a subtitle. We estimate that creating a subtitle will take around 1 hours, compared to the 4-5 hours needed in the classical way. The platform has been recently presented to the Italian TEDx community, in the context of the TEDx Italian Gathering 2019. Currently, we are in touch with the global TED curator and responsables of the TED Translators initiative to evaluate a potential collaboration with them.

¹⁸<https://gitlab.com/kercos/subsegmenter>

References

- [1] S. Bird, E. Klein, and E. Loper. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.", 2009.
- [2] J. Lafferty, A. McCallum, and F. C. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. 2001.
- [3] P. Lamere, P. Kwok, E. Gouvea, B. Raj, R. Singh, W. Walker, M. Warmuth, and P. Wolf. The cmu sphinx-4 speech recognition system. In *IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP 2003), Hong Kong*, volume 1, pages 2–5, 2003.
- [4] C. D. Manning, C. D. Manning, and H. Schütze. *Foundations of statistical natural language processing*. MIT press, 1999.
- [5] L. Màrquez and H. Rodríguez. Part-of-speech tagging using decision trees. In *European Conference on Machine Learning*, pages 25–36. Springer, 1998.
- [6] M. Straka, J. Hajic, and J. Straková. Udpipes: Trainable pipeline for processing conll-u files performing tokenization, morphological analysis, pos tagging and parsing. In *LREC*, 2016.