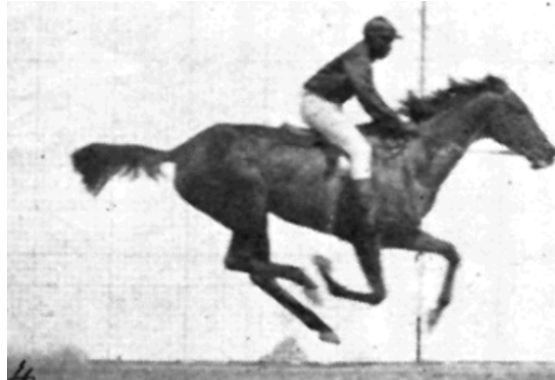




**TÉCNICO**  
LISBOA



## **Rethinking Video Interfaces for Usability and Editor's Performance**

**Miguel Borges Ribeiro**

Thesis to obtain the Master of Science Degree in

### **Mestrado Bolonha em Engenharia Informática e de Computadores**

Supervisor(s): Prof. Duarte Nuno Jardim Nunes

#### **Examination Committee**

Chairperson: Prof. Daniel Gonçalves

Supervisor: Prof. Duarte Nuno Jardim Nunes

Member of the Committee: Prof. João de Almeida Varelas Graça

**September 2020**

Dedicated to Unbabel. We were onto something.

## Acknowledgments

It is undeniable that Unbabel forged me into who I am today. As my first (real) job experience, being in a startup that grew from 40 to 200+ people, being part of all the evolution in a company for three and a half years broadened my horizons of professional and social life.

So, for that, its imperative to thank Vasco Pedro, João Graça, Bruno Silva, Hugo Silva and André Silva for committing and starting this project.

Thank you Paulo Dimas for trusting in me when I first came to Unbabel and teaching so much about the "real world". Starting the discovery team together and seeing it evolve was one of the things I'm most proud in all my journey in the company. Creating video was truly a risk, and I still believe we did a great job.

A mandatory thank you to Helena Moniz for reviewing my work and giving me so many ideas and guidance through the whole process of this thesis.

A big thank you to all people who once were part of the Video-Team - José Cortez, Luisa Ramos, Hasan G, Alexandre Solleiro, Luis Serrano, Rafael, Claudia Letra, André Teixeira, Sérgio Copeto and Miguel Carvalho. We were truly, a great team.

Thank you Prof. Nuno Jardim Nunes, Ph.D, for giving me full liberty on this thesis. It was a bumpy start, but it all fell into place in the end.

As I see this delivery as the closing chapter of my academic life in IST, I could not end without giving acknowledgements to the people that accompanied me, throughout these years. Since we were pretty much defined by group names, to all members in HSQD, É lidar, and Saudações Cordeais. What a ride.

And finally, to my parents. Which never pushed me to do anything, and just left their bird fly by itself.

## Resumo

Com a rápida evolução das plataformas que incorporam vídeo nos últimos anos, surgiu uma grande procura na inserção de legendas. Com tal crescimento, novas estratégias para a sua produção também surgiram, especialmente com o uso de inteligência artificial e estratégias de design para o seu desenvolvimento.

Com base, procurámos soluções propostas por investigadores na produção de transcrições e legendas cujo impacto tenha sido considerável e atual.

Após tal investigação e experiências propomos um novo fluxo na criação de legendas, sem precedentes. Adicionalmente, exploramos uma maneira inovadora de exibir texto na transcrição de um vídeo com a assistência de Inteligência Artificial. Com resultados promissores, mas não conclusivos.

Os nossos resultados mostram que uma abordagem semelhante a um editor de texto integrado com tecnologias de reconhecimento da fala para a edição de transcrição, pode ser uma maneira promissora para assistir os editores com trabalhos de transcrição.

**Palavras-chave:** Reconhecimento automático de voz, Reconhecimento de voz assistido por computador, Legendas, Transcrição, Tempo de resposta, Taxa de erro de palavras

## **Abstract**

With the evergrowing demand for video captions, the focus has turned into assisting humans with AI and Design strategies in order to make them faster and better.

We take a look into the state of the art solutions for transcription and caption production and some implementations from researchers and companies whose impact in this industry has been considerable.

With that, we propose a new flow to create captions which is unprecedented. Furthermore, we explore an innovative way to display the text when transcribing a video with AI assistance, with promising, but not conclusive results.

Our results show that a text-editor approach integrated with Automatic speech recognition (ASR) technology for transcription editing could be the optimal way to assist humans with ASR baselines for transcription.

**Keywords:** Automatic speech recognition, computer-assisted speech recognition, Captions, Transcription, turnaround-time, Word-Error-Rate

# Contents

Acknowledgments . . . . .	iii
Resumo . . . . .	iv
Abstract . . . . .	v
List of Tables . . . . .	vii
List of Figures . . . . .	viii
Nomenclature . . . . .	1
Glossary . . . . .	1
<b>1 Introduction</b>	<b>2</b>
<b>2 Motivation</b>	<b>4</b>
<b>3 Problem statement</b>	<b>7</b>
<b>4 Main Concepts and Definitions</b>	<b>8</b>
4.1 Measuring Quality . . . . .	9
4.1.1 Word Error Rate . . . . .	10
4.1.2 Bilingual Evaluation Understudy . . . . .	10
4.1.3 Multidimensional Quality Metric . . . . .	11
4.2 Turnaround-time . . . . .	12
<b>5 State of the art</b>	<b>13</b>
5.1 Standard Workflow . . . . .	13
5.2 Basic features of Transcription and Captioning Interfaces . . . . .	13
5.3 Representing time . . . . .	14
5.4 ASR applications . . . . .	15
<b>6 State of Tech</b>	<b>18</b>
6.1 The split - Rev's approach . . . . .	18
6.2 Respeaking . . . . .	20
<b>7 Experiments at Unbabel</b>	<b>21</b>
7.1 First steps in video . . . . .	21
7.1.1 Pipeline Overview . . . . .	22

7.1.2	Implementation details . . . . .	23
7.2	Interfaces . . . . .	23
7.2.1	Similarities between interfaces . . . . .	23
7.2.2	Transcription Tool . . . . .	24
7.2.3	Captioning Interface . . . . .	25
7.3	Design process . . . . .	27
7.4	New Transcription Interface . . . . .	28
7.5	Word-Mapping . . . . .	28
7.5.1	Exploring behaviours in Transcription Tool . . . . .	29
7.5.2	What are we testing? . . . . .	29
<b>8</b>	<b>Thesis experiments</b>	<b>31</b>
8.1	Experiment preparation . . . . .	31
8.1.1	Video selection . . . . .	32
8.2	User tour . . . . .	32
8.3	Testing Procedure . . . . .	34
8.4	Environment . . . . .	35
8.4.1	Consent form signature . . . . .	36
8.5	Pre testing . . . . .	36
8.6	User Profiles . . . . .	36
<b>9</b>	<b>Results</b>	<b>37</b>
9.1	Unbabel's pipeline . . . . .	43
<b>10</b>	<b>Discussion</b>	<b>44</b>
<b>11</b>	<b>Conclusions</b>	<b>46</b>
	<b>Bibliography</b>	<b>47</b>

# List of Tables

9.1	Results from average time taken . . . . .	38
9.2	Results from median time taken . . . . .	38
9.3	Results from coefficient of variation on time taken . . . . .	38
9.4	Results from T-student on time taken . . . . .	39
9.5	Results from average BLEU scores . . . . .	40
9.6	Results of median BLEU scores . . . . .	40
9.7	Coefficient of Variation on BLEU scores . . . . .	40
9.8	Results from average WER scores . . . . .	41
9.9	Average clicks with and without shortcuts in the first task . . . . .	41
9.10	Average clicks with and without shortcuts in the second task . . . . .	42
9.11	Turnaround-time comparison between users that used and did not use keyboard shortcuts	42



# List of Figures

1.1	Film/Video Timeline . . . . .	2
1.2	Horse in Motion (1878) . . . . .	3
2.1	YouTube display of captions while watching the video (left) and on preview (right) . . . . .	5
4.1	Example of the same video segment in a transcription and in captions . . . . .	9
4.2	Word Error Rate formula . . . . .	10
4.3	MQM graph . . . . .	11
5.1	Standard workflow by transcribers and captioners . . . . .	13
5.2	Rev’s captioning timeline on top, and Trint timeline on the bottom with waveform . . . . .	15
5.3	Interface used in [17] . . . . .	15
5.4	Proposed pipeline using ASR as a first step in [26] . . . . .	16
5.5	Examples of the interface combinations from [19] . . . . .	17
6.1	Rev type interface [29] . . . . .	19
6.2	Rev sync interface [29] . . . . .	19
7.1	Unbabel captioning pipeline . . . . .	22
7.2	Unbabel transcription interface [34] . . . . .	24
7.3	Unbabel captioning interface [34] . . . . .	25
7.4	Example of a caption on the interface and its produced result . . . . .	25
7.5	Table of the icons present in the captioning interface and its corresponding feature . . . . .	26
7.6	Unbabel’s timeline in the Captioning Tool . . . . .	26
7.7	Visual demonstration of sentence level and word level ASR . . . . .	28
7.8	Word-mapping text example . . . . .	29
7.9	The 3 different approaches of text display we are testing . . . . .	30
8.1	Example of the first step in the Tour . . . . .	33
8.2	Editor dashboard with the 2 tasks to be performed . . . . .	35
9.1	English test score distribution . . . . .	37
9.2	Turnaround-time boxplots . . . . .	39

9.3	Correlation between WER and BLEU scores where Word-mapping is blue, Sentence-level red, and sentence-level Word-mapping yellow. . . . .	41
9.4	Average time per minute of video in Transcription per week. Notice at week 50 word-mapping of 2019 put live. . . . .	43

# Chapter 1

## Introduction

According to Cisco data in 2020, 80% of the content consumed online will be video [1]. Today that number is already over 70%. This means that video is becoming and will continue to be the most impactful and preferred media in the world [2].

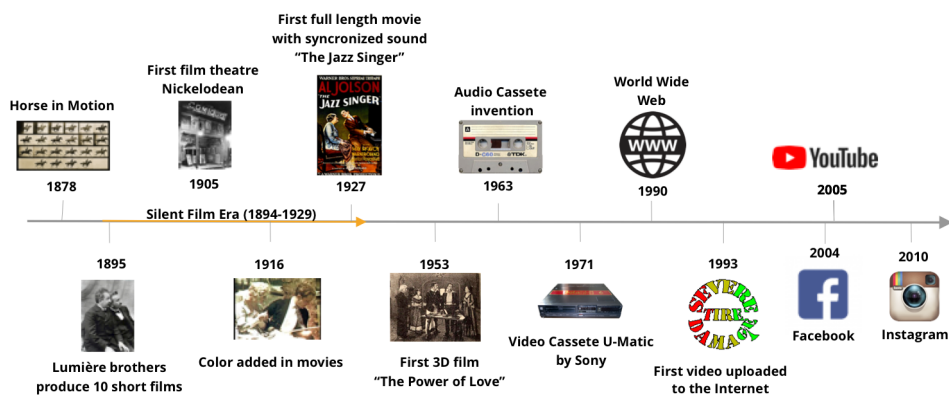


Figure 1.1: Film/Video Timeline

However, the core impact of video was not always recognized, as we will briefly highlight through a historical perspective of the growing importance of video, which can not be depicted from the cinema history itself. It all began in the 1800s, when various people started experimenting with photos, blending them together to give the sense of a motion picture. It was by then that "The Horse in Motion" emerged in 1878, the first movie ever made. This agglomerate of multiple camera pictures ended up answering one popular question: "Are all four of a horse's hooves ever off the ground at the same time while the horse is galloping?". The answer can clearly be seen in Figure 1.2:



Figure 1.2: Horse in Motion (1878)

In 1895, Auguste and Louis Lumiere invented the first motion picture camera and projector. Following that, 21 years later in 1916, the color was added to movies, one of them being the famously acclaimed “Wizard of Oz”. Until then, the sound was not present in the movies, that will not be accomplished until 1927, when “The Jazz Singer” was released with a synchronized analog sound system.

Even so, it was not long after the invention of the film that efforts were made to convey the dialogue of the actors to the audience. At the time, they used *intertitles* [3], which were texts drawn or printed on paper, that were filmed and placed between sequences of the film. These were first seen in 1903, and remained until the end of the silent film era. From 1927 onwards, with the invention of sound films, the audience heard the actors, so the intertitles disappeared and a new dimension of challenges emerged.

For a movie to be successful, its audience reach has to have a great coverage tackling also listeners to whom the language of the film was not their native one. To that end, with intertitles, one could simply translate the texts and film them. Without the translation process, the movie ended up revolving around dubbing, a process through which the sound was spoken in a different language than the original one. However, most of the film producers and distributors found (and still find [4]) this method complex and expensive. So, the viable solution was to insert the intertitles in the images themselves, thus creating what we now call subtitles. This is the first step of subtitling, linked to cinema and to audiences communication. Today we have subtitles everywhere. When we go to the movies, in TV stations, like the BBC, which has 100% of its content subtitled, or even Facebook or YouTube videos.

## Chapter 2

# Motivation

Currently, when we discuss videos, we instantly think of online videos. The reason for such immediate thought is due to the fact that online videos are a core reason for the growth of internet traffic, obviously related to the advent of social media. Online social media disrupted the video production industry in the last decade. And, since smartphones with powerful video capabilities became widespread, everybody could easily become a video producer. The rise of video content is partially due to the upgrade of cameras in smartphones, and social media promoting the creation of videos to share your life with features like Insta-Stories or Facebook Live. We are nowadays all video producers, as content user generators in the social media.

The known fact that social media and video are intrinsically linked and are constantly rising can also be supported in other marketing statistics, as listed below:

- More than 1 billion hours of videos are watched on YouTube each day [5].
- 87% of online marketers use video content [6]
- One-third of online activity is spent watching video. [2]
- 85% of the US internet audience watches videos online [7].
- The average user spends 88% more time on a website with video [8].

Fair to say that there is also video content within the same social media platforms that do not need translation processes, since the visual and captions are informative enough. To that respect it is also worthwhile mentioning that another critical piece of marketing data also shows that: 85% of Facebook videos are watched without sound [9]. This is an astonishing amount of 8 billion views per day. It is possible due to the videos having textual or speech captions narrating what is being shown, which are simply captions of what is being said without translation processes involved.

YouTube also started showing captions on video previews (Figure 2.1), much like Facebook. Furthermore, this platform incentivises creators and video producers to provide captions and subtitles for their audience, supporting them with multiple tools to achieve that purpose, some of which will be commented later on. The common denominator in the social media platforms is the clear notion that for producers,

the bigger the audience the better. To accomplish such task and reach out to the broadest audience, the global market, language barriers have to be abolished, hence the crucial importance of subtitles.

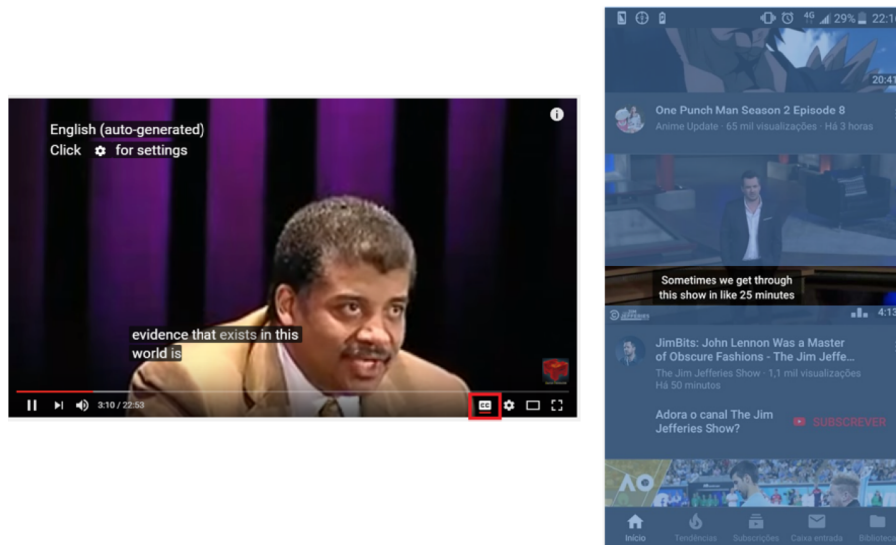


Figure 2.1: YouTube display of captions while watching the video (left) and on preview (right)

It's worth noticing that video captions benefit every human being [10]. Captions are especially important for the deaf and hard-of-hearing [11], they benefit children that are learning how to read [12, 13], people that are learning a new language [10], and also adults literacy [14]. They are the ultimate gate to access information widely.

This work is done in collaboration with Unbabel, a Portuguese AI startup providing their customers with a combined workflow of translation (Machine Translation + Human Post-editing/Translation). Unlike other automatic translation services, Unbabel integrates crowd-sourcing of more than 50 000 bilingual human post-editors. The translation pipeline once developed exclusively for text is tackling also captioning and subtitling, due mostly to the stats like the ones presented above. There is also a personal motivation for this work and I should say that this motivation is a driving force for my work. My interest in video comes a long way, since I was part of the Labs Team at Unbabel, which is responsible for creating new products. I was collaborating in the creation of a video platform. Until April of 2020 I was working as a Front-End developer at a small team, the Video Team, dedicated to create and explore how humans and AI can cooperate in transcribing and captioning. My research path, thus, focuses on understanding usability and performance metrics to better assist the editor.

The core research question that I will explore is: ***do standard video platforms allow the editor to perform their best in terms of speed and quality?*** The standard platforms tackle video as a monolithic and solid task with a single platform to do captions, without separating the complex stages into distinct phases. Although this thesis is not about subtitles directly, one question that arises for companies and has arisen for Unbabel as well is: Should we translate the captions, or create the subtitles directly from the video? This question has no immediate answer, but for the first hypothesis to be proved or demystified, it is only natural that the captions need to be as good as possible. And that is the objective of

this thesis, to assist the editor and to provide idiosyncratic platforms tuned to different tasks: captioning, transcribing, and (although not the direct object of interest) subtitling.

## Chapter 3

# Problem statement

With the rise of video content and its forecasted growth, as explained previously, the search for automated or semi-automated captions, transcriptions, and subtitles is a key research and development topic. Companies, and video producers, keep looking for the cheapest and fastest provider, while trying to increase the quality and reducing the costs. And since the demand keeps increasing, providers are not able to keep up with the requests, and are looking for new efficient ways of delivering these specialized transcription, caption and translation services.

Recent state-of-the-art methods for transcription and captioning include the use of Automatic Speech Recognition (ASR) to assist editors in their work. Instead of starting the video from scratch (blank), they begin with a baseline created by the ASR. This, as we will see, may save time in certain scenarios (clean and read speech), but it can also have the opposite effect in others, since, usually, user-generated content is not well dealt with by ASR systems.

Furthermore, the creation of transcription interfaces has seen a lot of attention in recent years due to the boom of volume. However, this trend is not accompanied but a growing interest in terms of research. To that respect, research on transcription User Experience (UX) is still very scarce [15–17]. As we will see, ASR technologies are still a bit off when it comes to the quality produced, meaning that transcribers will have to use interfaces to post-edit and/or produce the transcript. That is where this thesis will be focused on, in the creation and evolution of User Experience in transcription interfaces, with its main goal of decreasing editing time, increasing quality, which in return will mean better pricing.

Additionally, as a further step as we will explain, we'll create a captioning interface combining learning from research and some of Unbabel's experience with users.



## Chapter 4

# Main Concepts and Definitions

Diaz-Cintas and Remael (2007), distinguish between different kinds of subtitles: Intralingual subtitles, Interlingual subtitles, and Bilingual subtitles. Intralingual subtitling is the shift from oral to written content in the same language, hence the reluctance on calling it subtitling, since there is no translation involved in the process. This variety is also called captioning or closed-captioning. The other types involve translation of the content, which, although not the focus of this thesis, can be briefly described as the translation of the content into different languages (Interlingual) or even to two main languages (Bilingual).

Up to this moment, we have been using 3 main concepts, which will be clarified in this thesis. A more thorough analysis based on the differences between the most important ones (transcription, caption and subtitle) will be addressed.

Transcription is the process of transforming a video or audio into text (Diaz-Cintas Remael, 2007). The transcription process can be done in three ways: ASR, human transcriptionists, or a mixture of both, as we will be described further on. Transcription is a process in which the words that are said are written exactly like they were spoken. For example, when transcribing a language with a strong accent, sometimes the pronunciation of the word could be shortened, such as 'going to' to 'gonna'. Note that, transcription is not the same as the other two, in the sense that the produced result is solely a text, while captions have time-encodings with specific text.

Captions are time-encoded pieces of the transcription that can include storytelling audio elements included in the original video (music signs, speaker names, noises present on the video's audio). Time-encoding is the union of a counter number, start time and end time with the correspondent segment of the transcription, which can be seen here as an example (on the right):

Transcription	Captions
I'm going to show you my music skills. Here comes the sun.	1 00:00:00,400 --> 00:00:05,400 I'm going to show you my music skills
	2 00:00:05,900 --> 00:00:07,900 (gentle music) ♪ Here comes the sun ♪

Figure 4.1: Example of the same video segment in a transcription and in captions

In the right-hand example of Figure 4.1, we have two captions that are labeled with their corresponding numbers, followed by the start time and end time, and finishing with the segmented transcription. To illustrate the difference between captions and transcription, we can see in previously mentioned figure that transcription does not use audio elements (eg. “gentle music”), and does not include segmentation, as you can see in caption “1” where “music skills” is in the bottom line instead of one continued sentence as seen in the transcription

Subtitles can be seen as captions that are produced in a different language than the original one present in the video. These are used mostly by viewers that can hear the audio but have trouble understanding the spoken language. Although we are not going to focus on subtitles in this thesis, it is worth noting that subtitles are not mere translations of captions. In different languages, the structure of the text needs to change to keep coherence and fluency on the native grammar. One example of this is how English vs. Portuguese phrase the expression “brown eyes”, since Portuguese would say “eyes brown”, is a noun-adjective structure. This implies that the time encodings and transcription associated with the caption could be changed. This mere example, not as complex as one may find when entire captions/sentences can be restructured from one language to another, serves the purpose of bringing to our attention the question and challenge previously stated on translating captions or producing subtitles directly from the video.

## 4.1 Measuring Quality

In this subsection, we’ll look into the quality metrics used for transcription and captioning in order to access its quality level. These measurements are usually made by annotators which are natives or proficient linguists of the corresponding languages. The quality of the job is assigned to the work that has been annotated and to the editor who did it, so we can keep track of its previous assignments.

### 4.1.1 Word Error Rate

The Word Error Rate (WER) is a common metric of performance to measure ASR transcription systems [18, 19], that is derived from the Levenshtein distance [20]. To achieve its result, we need to have the recognized word sequence (transcript that was produced by ASR) and a reference word sequence (the correct transcript). With that, the WER formula is as follows:

$$WER = \frac{S + D + I}{N} = \frac{S + D + I}{S + D + C}$$

Figure 4.2: Word Error Rate formula

Where, “S” is the number of substitutions made, “D” is the number of deletions, “I” the number of insertions, “C” the number of correct words, and “N” is the number of words in the reference. A common issue with this formula is that there is no distinction between different errors since some may be more disruptive than others or even easier to correct. Another issue with this metric is the need to correct the transcript in order to measure the quality produced by the system. This corrected transcript needs to be done by a linguist, which can be more expensive than the usual transcript done by a regular editor.

Even so, with a positive record of jobs, we can apply a certain level of confidence to an editor. The formula can be applied automatically, comparing the jobs of the ASR and the jobs produced by the editor without needing the intervention of an annotator 100% of the time. At Unbabel, for example, only a sample of the transcriptions is actually annotated (evaluated by a professional linguist), because of the costs.

Additionally, we can take this metric as a way to measure how much an editor has edited the task. For instance, assume that an editor finished a task with a WER of 0%. Either the video was perfectly transcribed, or the editor is fraudulent. This last case (which happens occasionally) is a red-flag that can be automatically signaled. Hence the value in this automatic metric.

### 4.1.2 Bilingual Evaluation Understudy

One metric that exists to measure the quality of a machine-translated text is called Bilingual Evaluation Understudy (BLEU). This metric is, just like WER, inexpensive, and still very much used today.

BLEU produces a score between 0 and 1, correlating the machine-translated text with the reference text, measuring similarity between the two where a value closer to 1 represents a similar text. The reason we are looking into this translation metric is because its usability exceeds the use solely in that area. This metric compares two texts of the same language and looks at their similarities differently from WER.

Although WER also compares text, BLEU takes a different approach to check text affinity. Instead of checking for insertions, deletions or corrections, BLEU looks into what words can be found in the produced text, that is present in the reference text. If those words are present, BLEU gives a high score,

even if the words are dispersed from each other. This is why this metric group words together in N-grams, to make sure that structure is also correct, not only content. This structure detail of BLEU, is the reason we are going to use it, something that is not present in WER since it only compares word by word.

### 4.1.3 Multidimensional Quality Metric

Multidimensional Quality Metric (MQM), is originally a framework used to describe and access the quality of translated texts and identify specific issues in them [21]. MQM can be simplified to a small “Core” of 20 types of issues that represent the most common problems arising in quality assessment of translated texts. Although there is no official MQM for transcription Unbabel adopted a framework that can be represented in the following graph:

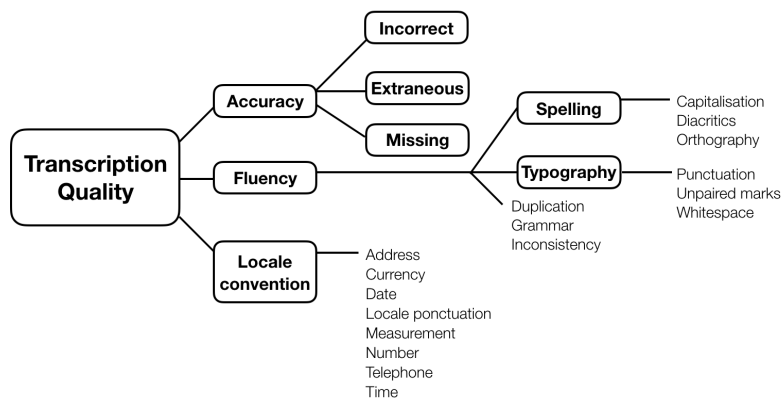


Figure 4.3: MQM graph

This graph consists of the core categories which encompass higher nodes and correspondent subtypes of errors. Each group can have subtypes. There are three main dimensions i) accuracy, ii) fluency, and iii) locale-convention. Accuracy aims to assess whether what is present in the source (video) matches what is in the target (transcript). Its errors are used to signal missing, extra, or incorrect words. Fluency has to do with the produced text, everything involving punctuation, spelling, or capitalization, without reference to the source. And finally, locale convention has to do with differences when producing transcripts for different languages or countries. Taking into example phone-number conventions, some countries could put their nine-digit number in groups of two or groups of three. To get the quality of transcription, linguists recruited by Unbabel annotate the work done by transcribers using the rules explained above. Each rule can have three levels of severity: minor, major, and critical, usually chosen by the annotator. After the annotation, the transcription quality score is measured as follows:

$$\text{Quality Score} = 100 - (\text{Issue Score} / \text{Job Words} * 100)$$

where

$$\text{Issue score} = \text{Sum}[i] (\text{Issue}[i] * \text{IssueSeverity}[i])$$

In the formula, “Job Words” stands for the number of words present in the transcription already done and Issue Severity translates into a number depending on the severity level chosen. These numbers are 1,5 and 10 corresponding to the order mentioned above. Above 95% MQM, the job is considered professional quality, under 70-90 is average, and below 70 is not good enough.

This metric is the most precise and trustworthy from the ones we have seen; however, it is also really expensive. Having the need to have a linguist look at a task and annotate each particular error or deviation has high costs. Because of that, we only choose to talk about it here to mention that a better and more accurate metric exists to access the quality levels of a task.

## **4.2 Turnaround-time**

Not related to these, is another concept which is one of the key points in this thesis, turnaround-time [22]. Turnaround-time represents the time from when the editor starts editing and then ends or submits the task. The lower the turnaround time, the faster the job has been done. It is important to lower this metric since it will correlate to a faster delivery for a customer, and also allow to reduce costs. Other definitions of turnaround-time could include the time from when the job is requested until it is delivered to the client, but since our focal point is in video interfaces, we will only consider editing time

# Chapter 5

## State of the art

In this section, we will see what features are usually present in transcription and captioning interfaces, followed by an analysis on state-of-the-art interfaces and finally, we will describe how ASR technology is being used. We will analyze features present in both and see how they can help to achieve the best possible quality.

### 5.1 Standard Workflow

In order to better understand the set of features involved, it's important to also understand the standard workflow of editors, either for transcriptionists or captioners. Simply put, the editors get access to a video or audio through an interface and perform their work there, usually starting from scratch and in a single interface, as a monolithic process.



Figure 5.1: Standard workflow by transcribers and captioners

### 5.2 Basic features of Transcription and Captioning Interfaces

With the rise of video content, multiple companies have launched their own transcription and captioning interfaces. Companies like Rev, Descript, Trint <sup>1</sup> and multiple others have taken their approach into making these interfaces, but they all share multiple features together.

<sup>1</sup><https://www.rev.com/>, <https://trint.com/>, <https://www.descript.com/>

First and foremost, the interfaces need to show the video since there are many subtleties that could only be transcribed if watching it. The video should come with the usual commands (play, pause, sound-bar), and should be controlled via keyboard shortcuts, which are key combinations or single keys that when pressed make the interface behave in a certain way [16]. The most used ones mentioned in most of the references are video play and pause, usually controlled by the 'Tab' key. We also have playback, which is a way of going back to the video. This is especially important for transcribers to re-listen to what was said, usually to check if what was written is accurate. In the same way, playback exists, usually, there also exists a forward action, skipping a few seconds to the front. Other than the undo/redo, which is assumed to be integrated (generally speaking), some processes/techniques of video acceleration or deceleration are also useful, because of the different speech rates in which people talk in the videos. Also, this can help in captioning to align the timestamps better. Both playback and other keyboard shortcuts are shown to be beneficial by transcribers in [17], from a usability questionnaire after the experiment.

Aside from the essential video, there are other useful features developed across commercial platforms. One of those being "search and replace". This feature allows a user to search for a word and correct all of its occurrences throughout the document. When working with ASR for noisy environments or low-quality recording conditions, it is usual that some words are recurrently wrong. This could be mitigated by correcting all of the words, or by correcting only one with this feature. Informal vocabulary produced by means of user content generation, for instance, is usually not correctly recognized, since the models are generally trained with more formal data. This issue can be reduced with the search and replace feature, especially in long videos where the incorrect words are repeatedly spoken.

The last feature most commonly used is the spell checker. This is essential when it comes to orthographic quality. There is also the possibility of having a specific spell checker just for transcriptions since there is a possibility of having to transcribe slang or shortened terms, something that does not happen in formal documents.

### **5.3 Representing time**

The most important difference between transcription and captioning is the use of time encodings to restrict transcription parts at a certain time. The need for a visually simple way for editors to move around time is therefore necessary and usually comes in the form of a timeline [17].

A timeline is a linear visual representation of a video, where you can see the time passing and where the transcriptions are located in the video [23]. This line could also have the video's waveform. This could provide a transcriber with some extra knowledge about the sound it is transcribing. For example, when time-encoding a caption, knowing precisely where the sentence ended is crucial to align the text with the audio. Also, the timeline can have some manipulation features such as draggable captions, much like a slider in the timeline.

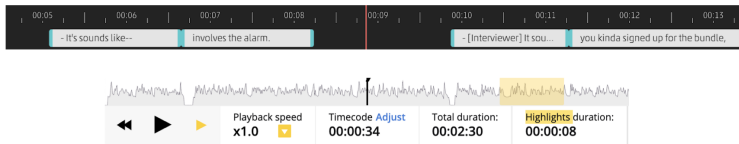


Figure 5.2: Rev's captioning timeline on top, and Trint timeline on the bottom with waveform

A different alternative to the timeline has been suggested and described in [24], which employs a table-like visualization to display the alternatives considered by the ASR component as a confusion matrix. These alternatives however are sentence restricted, meaning that only the words are modifiable, not time time-encodings. This can also be seen in a handwriting approach of ASR error correction in [25].

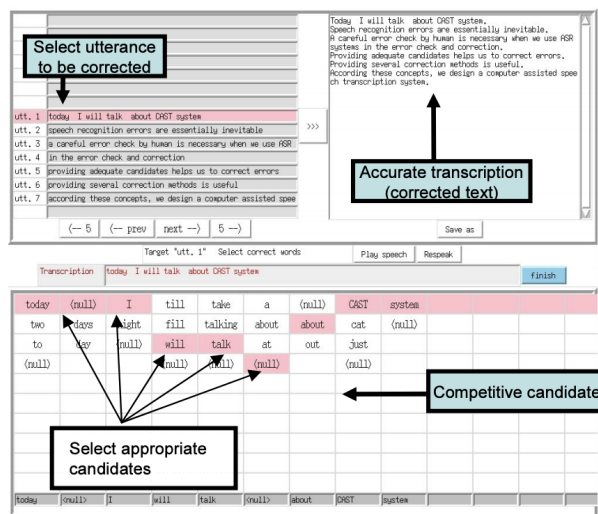


Figure 5.3: Interface used in [17]

## 5.4 ASR applications

With the increase of video content over the last few years, the demand for transcriptions and captions has also seen its uprise. To tackle this challenge there was a shift of attention to a well established AI technology called Automatic Speech Recognition (ASR) [19]. However, speech transcriptions by ASR are not yet perfect, and produce errors [19], especially for user content generation as social media data. To mitigate this issue, we consult the help of users through appropriately designed interactive interfaces to correct the errors produced by the ASR. This combination between user interface design and ASR is also known as Computer-assisted speech transcription.

ASR technologies only started to be used recently with the adoption of computer-aided speech recognition, to produce high-quality transcriptions. The two-step transcription strategy adopted in [17] consists of passing the audio through an ASR system and having transcribers pick up the corresponding output and start their work with that baseline. In spite of the not so perfect accuracy in speech recog-



dition, the time-encodings produced were sufficient for a boost in speed to produce captions. From the ASR only small tuning is actually required to produce the desired result hence the improvement

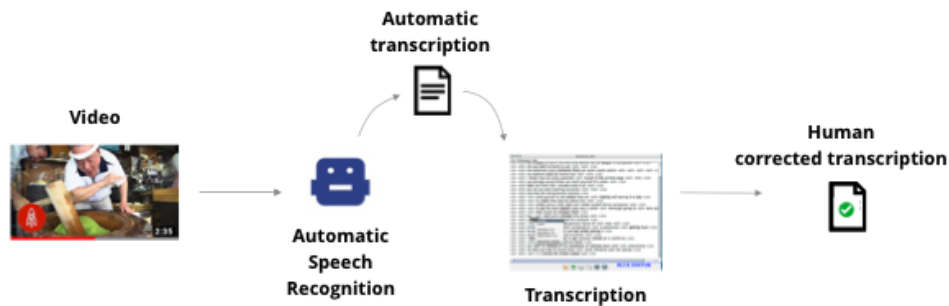


Figure 5.4: Proposed pipeline using ASR as a first step in [26]

With the endorsement of computer-aided speech recognition, there is usually a confidence level associated with the produced ASR since there are usually multiple alternatives for a certain word when it is automatically transcribed. In [17], the researchers provided the users with two interfaces, the green and blue one. The green one is described as a simple text editor, with some basic features, much like the ones I described at the beginning of this part. The blue one, on the other hand, highlighted words with low confidence levels of ASR. The user could then click on a highlighted segment to which the blue editor would activate a menu which showed the different possibilities of that specific segment.

According to this study, in terms of preference, the transcribers preferred the blue one to the green. There was a significant delay in terms of speed, when using the blue editor, since (according to the researchers), the usability of the blue editor was not as smooth as the green one, with the free typing. On the other hand, the quality was seen to be only slightly better on the blue editor, with its cost being taken from speed where the green one took the lead.

A similar study was done in [19], where the researchers gave their users four interface combinations. One which simply started from scratch with no ASR, other with the ASR output to post-edit, one where you had to start from scratch, but had input fields labeled with the ASR output showing with different colors the levels of word-confidence, and finally one where you could post edit the ASR and also had the labels with ASR showing confidence levels. The image below is an example of a sentence in these four scenarios where FS stands for “From-Scratch”, PE for “Post-editing” and ASRConf to represent the labels with the ASR:

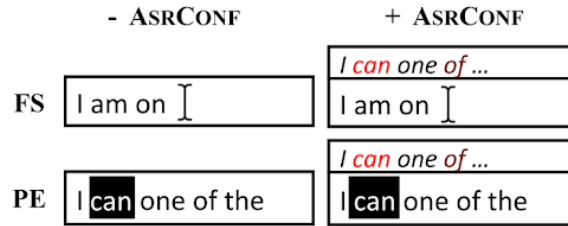


Figure 5.5: Examples of the interface combinations from [19]

The researchers firstly compared the From-Scratch against the Post-editing approach with no confidence levels in it; generally speaking, the latter requires less typing but requires special attention to verification and navigation in the produced ASR. Secondly, they compared the two approaches with ASR confidence labels (right column), where red symbolizes non-confident words and black confident words. The first approach, as seen in the top right corner of image N, involves the process of (re-)typing the whole transcript. Although the ASR is seen, the user has to manually type everything, an action that the researchers thought to result in the increasing attention of the users. The second approach provides the ASR labels, and the ASR ready to post-edit.

The results showed that a non-iterative From-Scratch approach was 'clearly outperformed' by the other which started with the ASR. Also, the difference between the interfaces where the ASR with confidence levels explicitly shown was not significant. However, as expected, the differences between performance happen when distinguishing levels of WER. For low WER (below 20%), the post-editing interfaces provided better results, but for high WER (20%-30%), the approach From-scratch with ASR confidence presented better quality. Which leads us to assume that if the ASR's WER is previously known, one can alter efficiently between approaches.

Also taking into account the findings in [15], it is better to start a transcription from scratch if the WER is higher than 30%. But, obtaining the WER of an ASR takes human work, which leads into a dead-end, since the pipeline would not be optimized if an annotator would first look into the ASR and then pass it accordingly to a transcriber. The editors, however, only start noticing the bad quality of an ASR at about 45% WER, to which point they start removing everything and starting from scratch, which is the most efficient way of doing the transcription instead of correcting the ASR output. We can, however, with some experiments, calculate the average WER of an ASR system and decide if it is worth continuing using it or no, depending on the results.

Although this thesis project is not focused on ASR technologies, it is of great importance to understand how to assist transcribers. Starting from a baseline is, most of the time, helpful and a time-saver especially for captions, because of the time-encodings. Even if the transcription quality is poor, the time-encodings only look at the wavelength and see if audio was present, to that respect captioners usually produce the time-encodings not taking into account what is actually being said.

## Chapter 6

# State of Tech

Research for captioning and transcription interfaces has not seen much light recently. There are multiple studies about how interlingual captions can be created with the use of machine learning algorithms such as neural networks [27] or deep learning[28]. Little has been studied, however, concerning interfaces and user experience. To better understand how our proposal is disruptive when compared to the current state-of-tech, we will be looking into the most acclaimed and used captioning interfaces of providers online, competitive in terms of quality and price.

### 6.1 The split - Rev's approach

Rev is a transcription and caption provider, being one of the providers which offers very competitive prices trusted by the New York Times, Disney and many other commercial partners. To achieve such a competitive low pricing compared to their competitors, special attention has to be given to their interfaces and tools to see how they could assist their captioners and lower their costs.

According to our research in their interfaces and multiple tutorial videos such as [29], Rev takes an interesting approach of splitting the captioning job in 2 stages, which they name 'Type' and 'Sync' – to the best of our knowledge, Rev is the only one doing that in the market. In Type, the users transcribe the video into small pieces of text that cannot go over a certain number of words. In the right of each sentence created, a timestamp is shown with the time in which that sentence was created (not written). This timestamp's only functionality is that, if clicked, the video will jump to that specific time. An image of this interface is shown below:

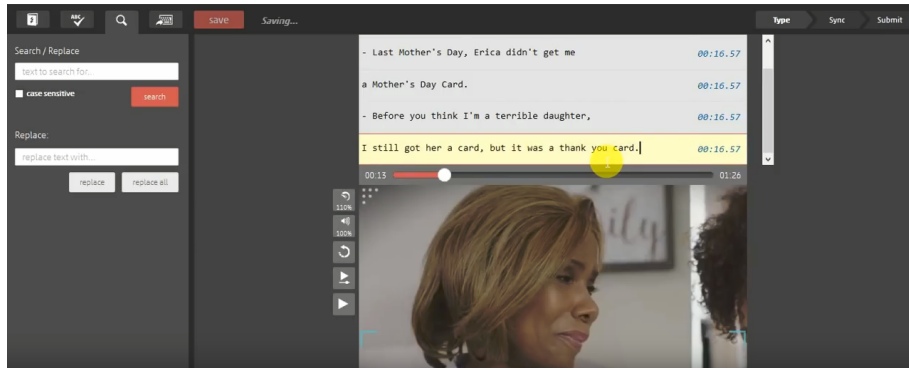
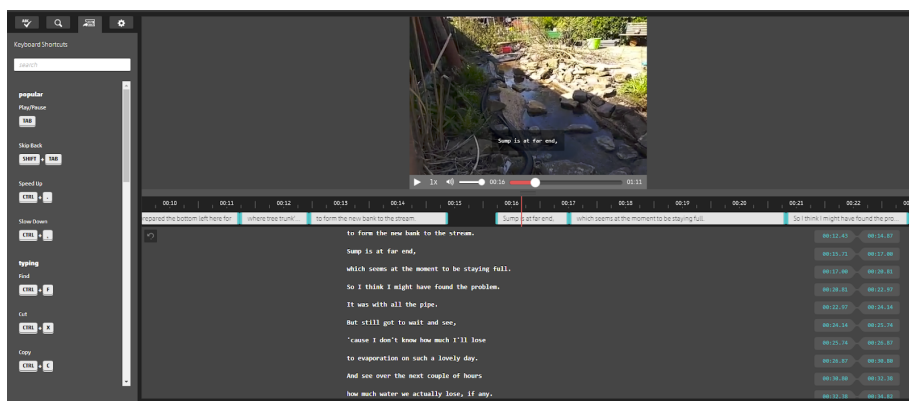


Figure 6.1: Rev type interface [29]

On the left, the user has information about features mentioned in the state-of-the-art section, such as keyboard shortcuts, search and replace, and spell checker. The user also has the ability to go around each sentence, in a similar way as a text editor (using arrows, creating new sentences with entering, deleting them with backspace, etc.).

When the work is finished, the user goes into the 'Sync' stage, where the work is done is brought on to. This interface now has an empty timeline in which the sentences made previously will be synced in to (hence the sync name we believe). For this, the user has to play the video and click on the up arrow indicating where the sentences start and end. For example, the user plays the video and presses the up arrow when it listens to the first sentence, which is highlighted in the bottom, and when it seems that it's over, he presses it again. This sentence, now caption, will be time-stamped with the times where the up arrow was pressed. The video would keep going and the user would do this for every sentence. Afterward, the times can be shortened, lengthened, or moved if the user sees fit, in the timeline. He can also edit the captions text, create new ones, delete or even merge. The following image shows the 'Sync' step':



the caption without their input. Segmentation has its own guidelines, so to automate it would be a huge timesaver for editors.

## 6.2 Respeaking

Respeaking has been a strategy used by captioners all over the world [30]. Respeaking is the process of repeating what is heard into an ASR software, which could be trained to a specific individual's voice and pronunciation. The ASR software then produces the captions as we know.

If we take into example conversational speech, it's usual WER is about 10% [31], but depending on the acoustic environment, or even the speed in which the speaker is talking, the WER can increase to over 30%. This is an example where we can use respeaking really effectively, because usually humans can understand conversational speech even with bad acoustic environment or fast talkers.

Although most of the respeaking technology has been used only by television networks, ASR quality has come to a point where it is very accurate when capturing slow and articulated speech, much like the one that should be produced by the speaker.

It is worth noting that the conditions for respeaking are really specific since one has to assure suitable environmental conditions for respeaking. Thus, a speaker has to be in a quiet environment to do his/her work, for not negatively impacting the ASR while listening to the video content with some sort of headphones (so the sound of the video does not interfere with the respeaking).

## Chapter 7

# Experiments at Unbabel

Unbabel is constantly developing new products to disrupt multilingual services globally. As a translation company with a community of bilinguals, Unbabel shifted some attention to subtitling. This started with a small application called Scribe that recorded a small clip of a video and sent it to editors to close caption, and from there those captions were sent to translate in the translation pipeline, creating subtitles. This process received very positive feedback, it was then shown to have a good response among the ones who tried it and confirmed the translations, and so Unbabel decided to push this experiment forward creating a small team, in which I was included, to focus on transcription, captions, and subtitles.

Before going specifically into the interfaces, some context has to be given about major decisions that were critical to the current state of the pipeline and its interfaces. To do that, we'll chronologically pass through these.

### 7.1 First steps in video

After the success of Scribe, it was clear that some attention had to be given to this content-type (video/audio). To pursue this interest, the first tool that produced captions in-house was built. This project was designed internally, and its objective was to create closed-captions from this interface alone. As an AI company, Unbabel uses machine translation as a baseline for translators to start their work, instead of beginning from scratch, since it has been shown to drop editing times dramatically [32]. The same model was applied to Scribe, meaning captioners always started from an ASR as a baseline, which we already saw, decreases editing times. Without much creativity on our side, this tool was called Captioning Tool.

At the time, the Captioning Tool interface consisted of the video, the captions, and some keyboard shortcuts. There was no timeline, automatic saving, line size restrictions, and a plethora of other small features such as performance improvements. And what we found was that captioners were having issues with the overload of information that was available for them. Some gave up in the middle of a task and some just struggled to understand some of the hidden mechanisms while interacting with the interface, such as, when writing something the video stopped, and only when the editor would stop

typing it would reset again. Worth noting that some of these editors had had no prior experience to captioning.

The initial results from early experiments were that the design of this tool had specifically been made to produce captions, not to correct text that was already produced by ASR. There was a huge gap in user experience, impacting negatively the user interactions, the turnaround time, and the quality. And so, the team decided to create a new interface, which uses, happened beforehand just focused on transcription and correction of the ASR. Only from there would a user pass to the captioning tool, where the text would be (hopefully) corrected. **This was the birth of a very disruptive strategy: split interfaces, like what Rev was doing, but using ASR as a baseline, which no competitor is using, in the same way as Unbabel was already using Machine Translation for the translation pipeline.**

The assessment of the quality in captioning was made through an in-house tool made by the Video Team, specifically tuned to this new product, and uses the metric MQM to get the accuracy of the job while also gathering the scores for WER.

### 7.1.1 Pipeline Overview

The following image is a representation of the current Unbabel captioning pipeline, including all the steps from when a client uploads a video, until it receives the SRT file back. As the picture displays, the architecture used by Unbabel is much more complex than the ones of the competitors.

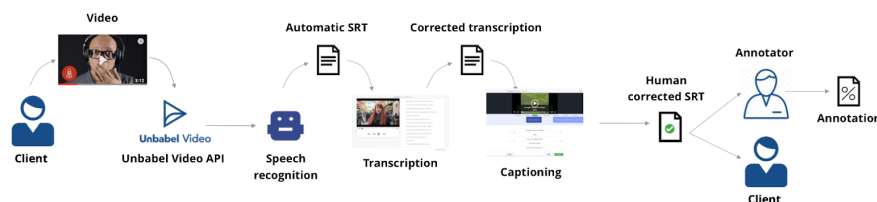


Figure 7.1: Unbabel captioning pipeline

A client sends a video to the Unbabel Video API, which is then immediately passed to Google's Speech-to-Text API, retrieving an automatically produced SRT file containing time-encoded sentences. These sentences are to be corrected and segmented by an editor. As previously mentioned, we divide the video's ASR correction into two steps. First, we pass it by a transcription interface which is the tool for a user to focus only on what is being said and making sure it corrects the mistakes of the ASR. From there, we send that corrected transcription to a captioning tool, whose aim is to correctly time-encode and segment the transcription with the video. Both of these will be explained in more detail in the next section.

After completion, the SRT is then given to the client. A sample of produced data is regularly annotated to assess the quality of the transcriptions and captions per client, domain, of even SRT engines (although we now solely use Google's).

## 7.1.2 Implementation details

To have a clear understanding of the work that was put into this pipeline, we will briefly have to look into the technologies that it uses, and see how they interact. As with all the user-facing platforms, we have back-end and front-end programs interacting with each other. Simply put, the front-end refers to the client-side ("what he sees"), and the back-end to the server-side ("what he does not see").

The back-end is written in a Python framework named Flask. It consists of a REST API, connected to a Postgresql database. Simply put, the back-end sends HTTP messages containing all the information necessary for the interfaces to render, and when the work on the interface is done, it sends to the back-end an HTTP request back with the information to be updated. Mostly done with POST or PUT HTTP requests.

On the other hand, the front-end, which is what I solely work on, is being developed in Vue.js, one of the most famous JavaScript frameworks [33], due to multiple factors such as speed, reactivity, and code simplicity. Vue is a progressive framework for building user interfaces and single-page web-pages. Progressive framework means that depending on the complexity of what is being built, Vue adapts itself to that complexity, meaning that a simple application should be easy and fast to be built, and for a bigger application Vue has all the right tooling to make sure you can scale up.

## 7.2 Interfaces

As we have seen, the job has been split into two different interfaces. In this subsection, we will look into some details of the interfaces such as features, behaviors and design choices.

### 7.2.1 Similarities between interfaces

Re-usability of code is one of the most important practices in programming. It helps to keep projects cleaner and simpler. But this is not only useful in programming, but in the design process of interfaces. Trying to keep the UI consistent by using the same components in multiple interfaces. With that, this subsection will touch on these shared features.

To start, when a user enters either interface, it gets prompted with a modal with some task instructions. These instructions give some context about the job the editor has to do. For example, on the Transcription interface, this modal asks the user to correct and punctuate the text, while on the Captioning tool it asks to segment and sync the captions with the video.

One other modal present in both Interfaces is the "Idle Mode". According to user testing on the Translation pipeline, it was noticeable that some editors were making no edits to the translation during some considerable amounts of time. These pauses were mostly related to fraud, so editors could earn more money, but as a result editing time data was not totally correct. To mitigate this issue, we applied Idle mode, which stops the counting editing time if the editor does not make any edits for more than a



minute. If the editor wants to start working again, he can just press the resume button and the task will continue as normal.

Another common feature in both the interfaces is keyboard shortcuts. This is a way to mitigate the usage of the mouse since editors are more efficient if they only use the keyboard. The most used shortcut is Pause/Play, and we also have Rewind, Forward, and Playback speed. The keys associated with these have to be specific and not used accidentally, since that would disturb the normal editing process. We then assigned these to easily accessible keys or key combinations such as 'Tab', 'Shift + Tab', or 'Cmd/Control + Tab'.

To conclude, we have the bottom bar, which consists of 3 different features. Firstly, it has a small notification that appears every 2 minutes advising the user that the work was saved. Then, we have the Skip button which is used if the user does not want to continue with the task. To this effect, the user has to provide the motivation for this skip, and so we provide some options about why this could be happening. Reasons such as "Audio has bad quality", "I can't understand the accent", etc. In the end, we have the submit button which prompts a small modal asking if the editor is sure he wants to submit the task. This was integrated since the Video Team received some feedback asking for this feature because some editors were miss-clicking this button submitting the task preemptively.

## 7.2.2 Transcription Tool

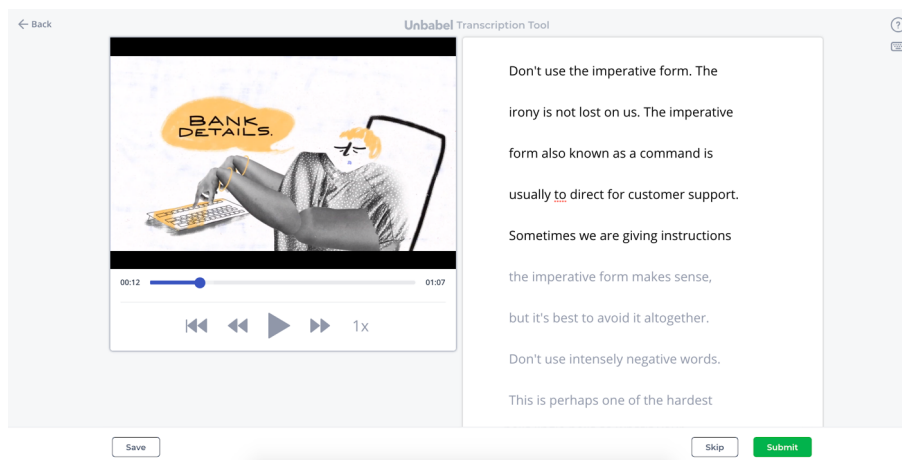


Figure 7.2: Unbabel transcription interface [34]

The transcription tool is composed of two parts. The video with the video commands on the bottom, and on the right, we have the ASR of the video split into sentences. There are the usual keyboard shortcuts such as start/stop or rewind/forward, and we allow the editor to move around sentences freely similarly to a text editor.

Although each sentence is encoded with a timestamp, there is no way to change these in this interface, since it's the main objective is to transcribe/correct and not align the sentences to the video. As we discussed above, it's usually the content of the captions that is wrong and not the time-encodings.

This allows us to build an interface that focuses solely on the transcription aspect of the equation. The timestamp editing comes in the next interface. This strategy may also mitigate the cognitive effort of an editor on taking care of two simultaneous complex tasks.

When playing the video, there has to be visual representation of the sentence that is currently being spoken. In this case, a timeline is not a viable solution, since we don't want the transcribers to be thinking about the timestamps of the captions, only its content. As a considerably easier solution, we turn into a simple highlighter that shows what text has already been spoken, and what is yet to come. This can be efficiently achieved by a different color of gray, where we have lighter gray on the future text and darker one on the already listened. This can be seen in Figure 7.2.

### 7.2.3 Captioning Interface

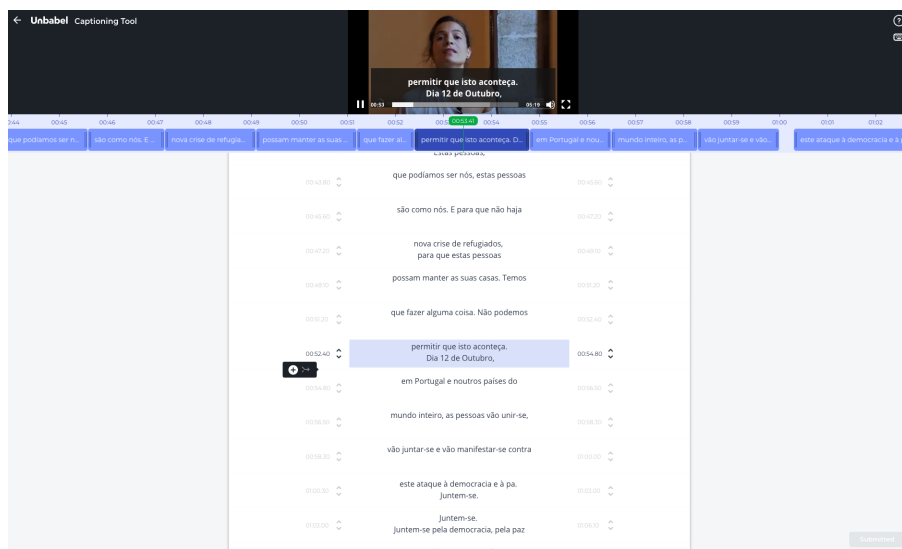


Figure 7.3: Unbabel captioning interface [34]

After submitting the transcription from the first interface, (usually) another editor does the captioning of that task. This interface was first built to be used as a standalone tool for producing the SRT's, and currently, it is still possible to do everything just in this tool.

In this interface, the captions have a 2 line area, that defines how the caption will actually look when exported. This area is limited to 2 lines only since the standard SRT format limits it that way. The editors should then form the captions as they would see suitable in each situation, following the guidelines that are provided beforehand. An example of a caption and its produced SRT are visible in the next figure:

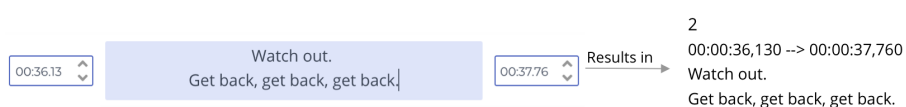


Figure 7.4: Example of a caption on the interface and its produced result

Differently from the transcription interface, there is a possibility here to add and remove captions. Also, the interface provides the option to clear the text from the caption, and an option to merge 2 captions together. All of these are represented with an icon, and when hovered, a tooltip is shown with the icon's feature. A table of these icons and their functionalities can be seen in the following image:






	Feature	Explanation/Rationale
	Add caption	Although the ASR produces captions automatically, there are times where creating new captions is necessary to keep coherency and readability.
	Merge captions	Merging captions puts together two adjacent captions. The resulting time encodings are the start-time of the first and end-time of the second.
	Set caption to current time	For precise time alignment, a user would click in this button to set the time of a caption to the exact time the video is in.
	Clear caption	Clear the text of the caption, leaving the time encodings the same.
	Delete caption	Delete the caption completely.

Figure 7.5: Table of the icons present in the captioning interface and its corresponding feature

As we have seen before, time representation is an important feature when time-encodings have to be produced and managed. To this end, a timeline was built on this interface, so we could drag, shorten or lengthen time-encodings freely, with as little traction as possible. We proposed that the editor did not need to write them specifically, since that is a cognitively demanding job, and instead he can "play around" with the captions in this component. Building this component, we have to take into account some limitations regarding the minimum time that a caption needs to be shown in the video (usually one-second minimum) and its connection to the others captions (they cannot overlap whatsoever). A more detailed image of the timeline can be seen in the following figure.



Figure 7.6: Unbabel's timeline in the Captioning Tool

Notice that each box has one handle on each side that can be used to lengthen or shorten the caption with the user's mouse. It can also be moved around if grabbed in the middle. To add clarity, we also put a bit of the text the caption has inside the boxes, and a little green timer corresponding to the video's current time.

One important aspect of this interface is how interconnected everything is. As shown in Figure 7.3, when a caption is active, it is shown in the video and highlighted in the timeline. This gives the editor

important information such as exact information on when a caption starts and ends and also how the caption looks when applied in the video.

Lastly, there was a usability improvement we did on this interface regarding splitting and merging captions together. We noticed through some user interviews, that users were having difficulties splitting and merging captions, due to the fact they had to "create a new caption, copy and paste the desired text from the caption I started on, and finally deleting the selected text from the original caption" - taken from one interview. Our approach was then to make an approach following some text-editor behavior.

Usually pressing the Enter button creates a new line, and in the captioning interface, it will now create a new caption. If then, the user clicks Enter when the caret (the little line where you are writing) is in the middle of a caption, the first portion will stay in the original caption, and the second portion will be put on a newly created caption. The timestamp of this caption will start with the minimum amount of time a caption can have, one second, and can then be adjusted like all other captions. In the same mind, if a user presses backspace at the beginning of a line, the paragraph joins the top one, which is what we made for merging captions, instead of having to click the purposefully built button, which we still left.

The ultimate objective is to empower captioners as they discover the tool. Some basic behaviors might come from instinct such as these last described, and we want them to feel approachable and simple.

## 7.3 Design process

Before jumping into the next steps we can take in this pipeline, it's important to understand how we came to these interfaces after months of work.

It is obvious that this is the result of much research, many experiments and user interviews. The point in this section is to clarify the process which took many of my days of work.

To begin, we looked into interfaces of our main competitors and saw what ideas they had come up with. The idea is not to reinvent the wheel, but to do something after building it. So, we took features that we saw could have a positive effect on editor efficiency and quality, designed it according to our internal design guidelines, and implemented it.

After that, we measured the performance of the feature with data from before the implementation and after, and see if it's results were pointing in a good direction.

If yes, the solution is simple, keep it. If not, we removed it, or thought about another way of implementing it. We have to remember that if another leading company has this feature, the benefits must be somewhere.

Then, around the beginning of each quarter, we used to gather around ten Unbabel editors and adjourned an interview with them. In these interviews we understood a bit about the editors that worked for us, features that they liked, features that they felt were missing and some questions about their overall work. They were particularly interesting, because we have to realise that these are the people that use the tools we make! And most of the time, they had really useful and interesting insights about features that could be advantageous. Some of which, we had never thought about. Hence the importance of

these interviews.

These processes were always discussed and analysed with professional linguists that had already worked in this area, and kept a close attention to the features we were introducing and removing from the tool.

As a side-note however, its important to comment the fact that we are building new things that have never been done before. There is always bound to be some traction by new and "old-fashioned" users. We have to think and decide regardless of the comments, if some features are worth to try. Some were and some were not.

## 7.4 New Transcription Interface

With the innovative approach of splitting the job between two interfaces, and starting with an SRT, there were still some ways we could further improve the pipeline, especially, in the Transcription Interface. To explain how we can further improve the Transcription step, we have to look into how we are working with the ASR that is rendered in the interface. There are two ways to request ASR from Google's API: on the one hand, we have sentence-level ASR and on the other hand we have word level (see Figure 7.7). Simply put, one has sentences timestamped, and the other has individual words. As you can see in Fig 15, the text seems to be split on a sentence level.

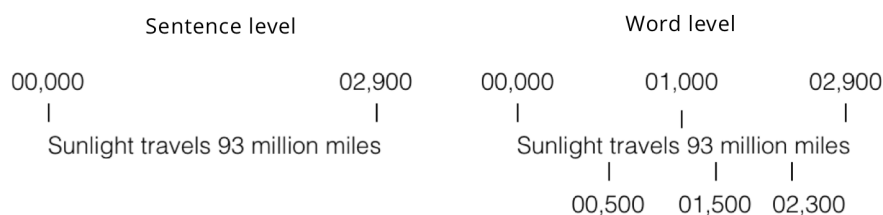


Figure 7.7: Visual demonstration of sentence level and word level ASR

The suggestion then, is to decrease the level of granularity of the produced ASR, so that editors can work faster. This is the experimental part of this thesis. To decrease transcription editing time with the implementation of a new interface designed to support word-level ASR, called Word-Mapping.

We will then **compare 3 different approaches and try to understand which is the most efficient one**, looking into parameters such as editing time, quality, and number of clicks.

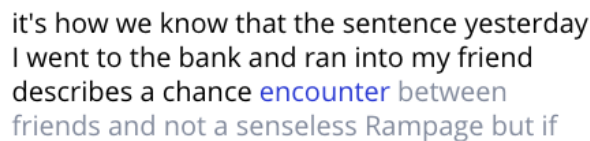
## 7.5 Word-Mapping

Although there is no scientific source for the name Word-Mapping it came from the literal usability of it. Each word is *mapped* with its start and end times, hence the name. And so, what are the advantages of this? The current Transcription tool already managed to achieve the task at hand, so what is the necessity of this approach?

To better understand how changing the level of granularity in the ASR will theoretically help the users transcribe faster, we will have to look into some specific behaviors in the current Transcription interface, and how they will change in the new one.

### 7.5.1 Exploring behaviours in Transcription Tool

Without the use of timestamps to assist the user in this interface, differently from the captioning tool, we will need to help him locate himself in the transcription job while the video is playing. What we want to avoid, is for the user to be lost in the ASR when listening to the video. In our first approach to solve this problem, as we have seen above, we proceeded to highlight the sentence which was being spoken. Now, we will highlight the specific word, as we can see in the sentence of Figure 7.8. In this case, the word "encounter" is currently being spoken, while the following word "between" is to follow.



it's how we know that the sentence yesterday  
I went to the bank and ran into my friend  
describes a chance encounter between  
friends and not a senseless Rampage but if

Figure 7.8: Word-mapping text example

With this behavior, the user has a visual queue of where the video is currently on the ASR. So how can the editor listen to a specific part of the text? There are multiple ways to accomplish that: clicking on the video progress bar, going back or forward on the video through a keyboard shortcut, but the most efficient way, would be to click on a word. We implemented word-mapping to have this advantage on the re-listening component of usability. In the old sentence level rendering, if the user clicked on a sentence, he would have to re-listen to all of it, something that users found quite frustrating. To mitigate that in this old approach, we decided to chunk down big sentences in about 15 words each (maximum).

The new approach of having word-mapped sentences, gave us liberty of placing them fully together, instead of chunking them down. This was seen as an improvement, because the user can have a smoother interaction with the text, going into the direction of working with this interface like a text-editor.

This text will then pass by an external service called Speechmatics <sup>1</sup>, which will chunk the text into small captions taking into consideration syntax and audio. Those captions are then, to be further adjusted in the Captioning Tool.

### 7.5.2 What are we testing?

In the end, what we are trying to find is in what way can we display the text to make users more empowered and efficient transcribing. We have already understood that there are many ways of improving

---

<sup>1</sup><https://www.speechmatics.com/>

editors efficiency, but none of those is text display related.

The three different approaches we are testing can be visually seen in the following image:

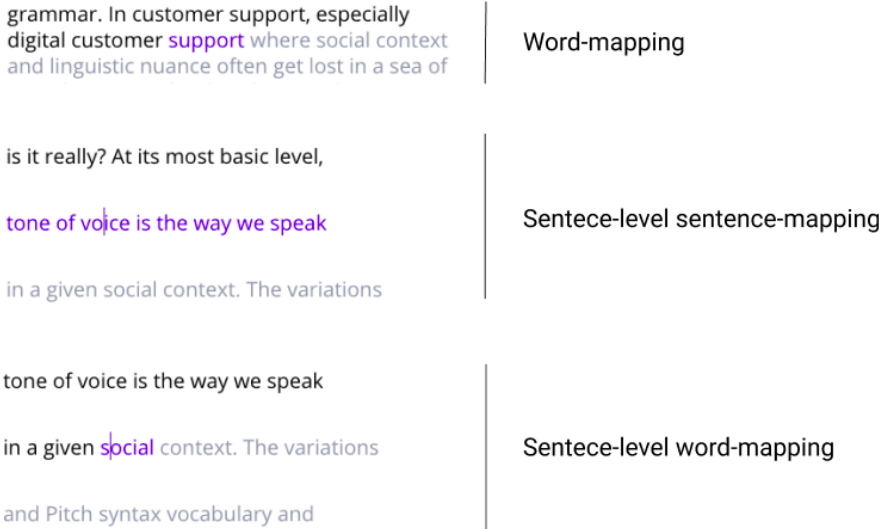


Figure 7.9: The 3 different approaches of text display we are testing

The gap seen in the sentence level approaches are paragraphs, so you need to forcefully go to the next or previous line with a button up or down, instead of just going forward in the text (clicking there is also a viable option). This was the approach we previously had, to render the Google’s ASR.

We are then comparing the new approach with the old one, and having a third approach, which can be seen as a mixture of the two. Sentence-level word mapping has the same display as the second one, but the timestamp distribution of the first. This is to understand if having word-mapping as a feature would by itself have improved editing times of the second interface or if having a free text-editor feel would be the difference.

# Chapter 8

## Thesis experiments

With these hypothesis, we are trying to build an interface that would revolutionize how text is rendered using an ASR baseline to work on. Because the market for video content is evergrowing, especially these past years[1], having an efficient way of producing transcripts, captions or subtitles is something worth exploring. To scope down the complexity we'll, as previously explained, look into Transcription solely, not forgetting that it could further on be used to produce captions or subtitles (if translated).

### 8.1 Experiment preparation

In order to test the efficiency of the tools, we'll use a famous experience research methodology: A/B testing [35]. This technique is a way to compare two or more versions of a single variable, typically by testing a subject's response to variant A against variant B, and determining which of the two variants is more effective. In this case, the variable will be how the text is rendered in the interface.

For no bias to be induced in this experiment, we won't use current editors from Unbabel's tools since they already know how it works. Instead, we'll recruit individuals who have the same profile as our current editors and have never seen our tools. Simply put, the only requirement for this experiment is a C1 or C2 English level, which we will access with the English test publicly available from Cambridge Assessment English [36], with the category "General English". This test was put in a survey online tool called Typeform, which would allow us to save the data of all questions and results.

If the Transcription job starts with an ASR, we can arguably say that it consists in correcting the text which was produced. However, there are some rules to consider if an editor is new and starts doing transcription jobs. There are certain guidelines to follow; things like punctuation and capitalization, and some other specific rules. For example, foreign languages spoken in the video should be put in brackets as such: "(speaking -Insert language-)". These specific things are not provided by the ASR, and in this exact situation of the foreign language, the ASR would produce something resembling what was spoken but in the wrong language. This is what we explain to our editors, and we'll also need to illustrate to the testers.

For the experiments we are running, it is mandatory to give the testers the least amount of information



possible, but still, making them empowered enough to produce high level quality. For this, we had to meticulously choose the videos we were going to use. Not only to reduce the amount of guidelines these testers had to read, but we also needed to consider the level of difficulty of the video and its length. Also, taking what we learned in [15], the WER of the video's ASR should not be over 30%, but lower, so it makes sense to start the transcription job with the ASR draft.

### **8.1.1 Video selection**

The first decision we faced when preparing the experiment was the duration of the videos. Looking into some data from the current Unbabel transcription editors, on average they take about 5.4 minutes to complete one minute of video. For the experiments, the users will have no experience whatsoever, so we need to assume they can take a lot more.

Setting a cap in this experiment at about 30 minutes per tester, we need to check how long a new user takes to complete a minute of transcription and then decide how long the videos should be. Ideally, we would want to split the experiment of transcribing into two different tasks. One would be the first task for the user to get used to the tool, and the second one would have already a informed user. We'll then also see if the learning process was quicker with our hypothesis.

Other than considering video duration, we also need to check WER values of the videos we choose. For this, we looked into some old jobs in Unbabel's pipeline and understood that for jobs to have even more than 10% WER, they would need to have some serious issue, that being, harsh accent, low sound quality or overlapping dialogue. In all other cases, the WER produced by the ASR was never higher than 7-8%. In fact, we are not really interested in correcting a hard job, but checking if the efficiency of going through the video is smooth and fast, never dismissing quality. Because of this, and to also avoid some copyright issues, we choose Unbabel videos produced in-house with a WER of 11% , taken from baseline test (explained further on), due to its lack of punctuation, capitalization and some missed words.

## **8.2 User tour**

Since we are going to use new editors, we had to find a way to consistently, repeatedly and equally, explain to them how to use the Transcription tool. For that, we had to build what we call a "User Tour".

A User Tour consists in a way to present the interface to the editor, explaining what the different functionalities available are, and what can be done and where. In the following image you can take an example of what this tour looks like:

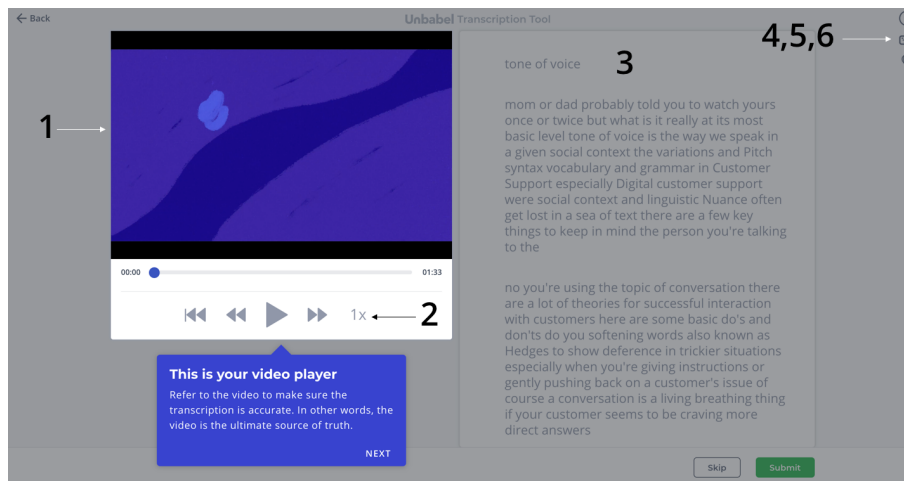


Figure 8.1: Example of the first step in the Tour

When the user starts the tour, a highlighted area is shown with a small description of what it is, and is useful for. Some functionalities can be quite obvious, but for a user's first time transcribing we have to make sure everybody is on-board with all of the features.

The following list includes the six different steps of the tour. The numbers on Figure 8.1 are here corresponding to each feature. The first point corresponds to the text shown on the interface, whereas the second one is a brief explanation on our rationale behind them:

#### 1. Video Player

- "Refer to the video to make sure the transcription is accurate. In other words, the video is the ultimate source of truth."
- The main focus has to be in the video, taking it as a source of truth and not the ASR;

#### 2. Playback Speed option

- "If you want to slow down or speed up the video, you can use this option. This can be helpful if speech in video is too slow or fast"
- If users have a hard time keeping up with the speed of the video, or think that it is slow, they can adjust the speed in which it reproduces;

#### 3. The first line of the transcription

- "This is your main working area. Please review the transcription comparing it to the video and improve it."
- This is simply to point out that this is the main area of work;

#### 4. The first button on the top right corner (Figure 8.1) - Guidelines

- "Guidelines help you understand the translation rules in depth. Please take the time to study it."

- If users want to consult the guidelines, they can with that button. For this experiment however, they have all the guidelines needed in paper next to them;

#### 5. The second button - Keyboard shortcuts

- "Using shortcuts helps you work much faster. We highly recommend using them early on, to get you into the habit."
- As we have seen in [17], keyboard shortcuts are beneficial for efficiency, so it was mandatory we introduced these to the editor;

#### 6. The third button - User Tour

- "There will be a time when you'll forget some of the wonderful features of this interface. When that happens, just click here."
- In case an editor wants to be reminded of the existing features, he can just click in this button, and the tour will start again;

Worth mentioning that the user tour only runs once, in the first task from the user. It will only run again if forced by the user with this last button.

## 8.3 Testing Procedure

The test structure was the following:

1. Introduction
2. Signing consent form
3. English Proficiency test
4. Guideline reading
5. Task 1 / Task 2

In Introduction we thank the participant for his participation and explain that the experiment will take about 30 minutes in total. To start, we ask the participant to fill a consent form. He will then do the English test which is prepared in Typeform, which has 25 multiple selection questions. If the participant gets less than 15 of the 25 right, he is not suited for the experiment. Either way, we'll ask him to do the complete experiment, although his results will not be counted for the results. This is so that we avoid "refusing" a participant.

Continuing, an overview of the task will be given. This overview contains context about the task and how to get to the interface through the dashboard they will have access to (Figure 8.2). This explanation has under no circumstance information about the tool. We want users to figure out the tool by themselves to avoid any performance bias in this study.

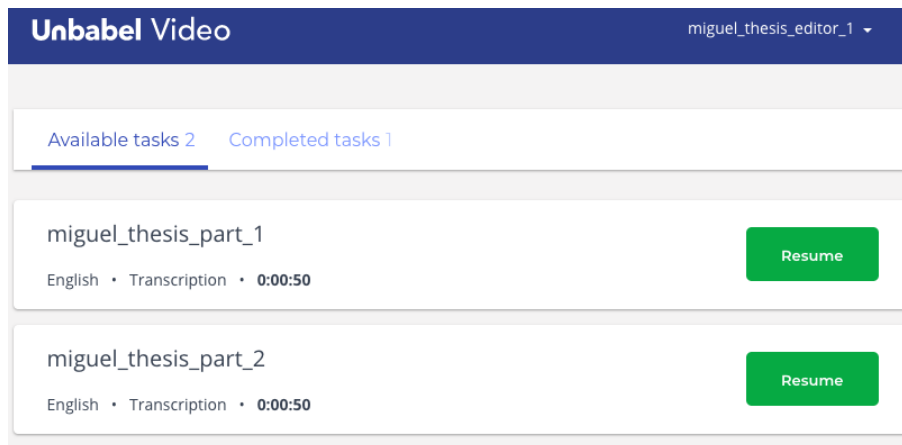


Figure 8.2: Editor dashboard with the 2 tasks to be performed

The users will then start the tasks in the controlled environment without any interactions with the experiment supervisor.

## 8.4 Environment

Originally, this experiment was prepared to be in person. A Mac-Air was ready with the Typeform and dashboard, but due to the current coronavirus epidemic, we could not have such a controlled environment.

Even so, solutions were found. Using Zoom [37], we could communicate with the testers remotely, making them read the necessary documents and sending them the transcription tasks. Not only that, we asked people to share their screen while doing the tasks and we recorded those sessions locally. On top of that, the Transcription tool also has implemented a recording tool called Fullstory [38]. This tool gives us information about clicks on specific areas, speed, and overall information about the page, automatically.

The reason we are recording these sessions with Zoom, is because Fullstory has been experiencing some downtime in the past few weeks and we want to ensure that the session was recorded if future analysis is needed.

In this experiment, since sound is the most important feature, we are asking participants to use headphones to avoid any outside sounds that might interfere with the user's understanding of the video. Originally, noise cancelling headphones were to be used, but due to the current circumstance that is impossible.

On the positive side, an issue we had with pre testers (see the following section) was that one of them was not used to the US keyboard we were using (mostly for punctuation purposes), and this situation asked that users did these tasks in their own personal computers.

### **8.4.1 Consent form signature**

Due to the current fact that these experiments cannot be done in person due to COVID-19, consent will be asked and recorded as mentioned, as a verbal agreement. All of the recordings collected in this experiment have the participant reading the following: " I (participants name) allow my data to be used in the context of the thesis experiments run by the student Miguel Ribeiro with the name "Rethinking Video Interfaces for Usability and Editors' Performance".

## **8.5 Pre testing**

Pre testing consists in testing the experiment fully, before actually testing it. This is to ensure the experiment is well designed and smooth for the user.

For this, two pre tests were made, one with a normal user much as the ones we are going to have, and one with a Professor specialized in Transcription and Audio-Visual content [39].

Both scored C2 levels in English, which is the highest possible, and after reading the guidelines they continued to doing the tasks.

In the initial draft of this experiment, some miss-calculations were done, and 6 minutes of video in total were asked to be transcribed. This led to exceeding the stipulated time of 30 minutes for the experiment, which is something we are not looking for. Neither finished the second transcription task, due to a duration of above 7 minutes of editing per minute of video.

We then decided to shorten the amount of minutes done by half, so the experiments could finish under those 30 minutes on average. We then decided to use one full video of three minutes split into two pieces.

## **8.6 User Profiles**

Before jumping to results, some information can be said about the 30 users.

The users were acquired doing an announcement in a public Unbabel channel using Slack, an internal communication platform. Of the ones that applied, I picked them in order according to application time.

All of them, without exception, also worked in Unbabel at the time of the experiments in the week of 13 to 17 of April 2020. These were fellow employees which I knew, but none had seen the interface whatsoever. This is to avoid any bias or previous experience.

Also, none of the users had ever done any job related to transcription or captioning, and hadn't seen any interface with the purpose of producing them.

The distribution of sexes was of 16 male to 14 female and of those, 28 where between the ages of 21 to 31, and two of them between 32-41.

Of these, one was a native English speaker, and skipped the English test. Not to mention it forward, this user's results were on par with the others, so no special attention will be given.

# Chapter 9

## Results

The experiment was performed by 30 different participants, two of which were not considered for the results since they were interrupted midway during the transcription task. In the end, the time per total testing session averaged 25 minutes.

Due to the interrupted tests, we finished up with an uneven number of tests on the last interface. While the first two had 10 participants, the last one had eight.

Starting with the English test, participants had an average score of 20,08%, with the value distribution seen in the following:

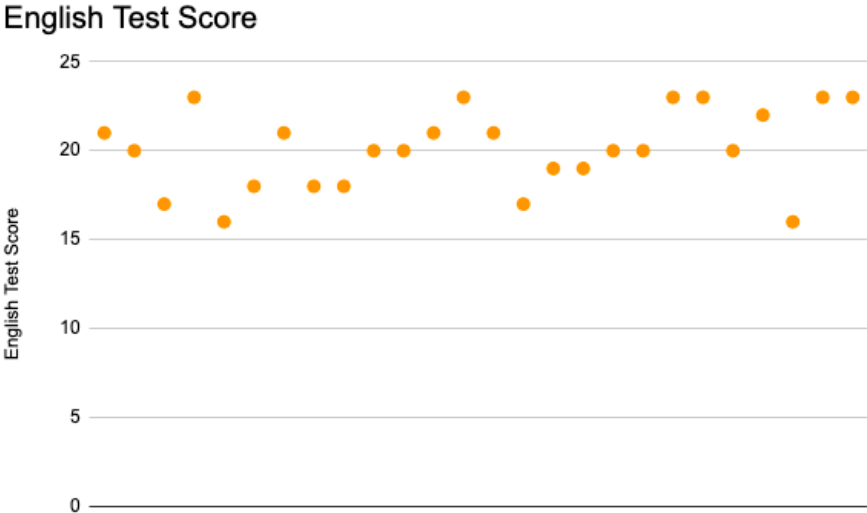


Figure 9.1: English test score distribution

*Luckily* none of the participants scored less than 15, which would by default invalidate their test results.

Before looking into the editing-time results, we have to explain how we are measuring the time taken and how we'll compare it. Since the tasks are 93 seconds and 67 seconds correspondingly, it is natural that the first task will have a longer editing time. To even out the metric, we use time taken per minute

Average minutes per minute of task	Task 1	Task 2
Word-mapping	5.8	4.3
Sentence-level Sentence-mapping	6.63	4.96
Sentence-level Word-mapping	6.78	5.45

Table 9.1: Results from average time taken

Median minutes per minute of task	Task 1	Task 2
Word-mapping	5.86	4.34
Sentence-level Sentence-mapping	6.30	5.01
Sentence-level Word-mapping	6.45	5.42

Table 9.2: Results from median time taken

of video, this way we can normalize results in order to be compared without bias. Simply put, we divide the seconds taken, by the total duration of the video in seconds.

Now, looking into editing-time, we can compare the average time per minute in the table bellow:

We can see from Table 9.1 that word-mapping has the best turnaround-time <sup>1</sup> average per minute, followed by sentence-level sentence-mapping, and then sentence level word-mapping. Analysing the median, we can see that the times are really close to the average. The coefficient of variation points at us that the data is not running much from the average score since its values are below one.

To have a better sense of the results from turnaround-time, we can also look into the boxplot from its values:

---

<sup>1</sup>Time from when a user starts a task until he delivers it

Coefficient of variation per minute of task	Task 1	Task 2
Word-mapping	0.34	0.35
Sentence-level Sentence-mapping	0.44	0.31
Sentence-level Word-mapping	0.29	0.17

Table 9.3: Results from coefficient of variation on time taken

T-student Percentage		WM	SL SM	SL WM
Task 1				
WM	x	53,51%	79,78%	
SL SM	-	x	78,68%	

T-student Percentage		WM	SL SM	SL WM
Task 2				
WM	x	63,24%	92,18%	
SL SM	-	x	56,83%	

Table 9.4: Results from T-student on time taken

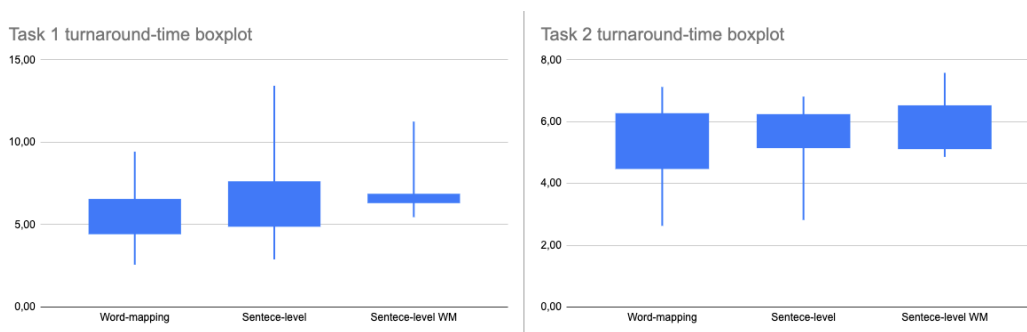


Figure 9.2: Turnaround-time boxplots

We can notice that the averages from both boxplots are slightly lower in Word-mapping and that the minimum is also the lowest of them all in that interface. Still, no real conclusions can be drawn from here, and we have to analyse the T-student values to see if any real outcomes can be drawn.

A T-student test consists in comparing two sets of quantitative data that are collected independently of one another. Simply put, It's results can, in our context, make us understand if an interface is faster than another when comparing them to each other.

As we can conclude from Table 9.4, no considerable findings can be taken from turnaround-time efficiency as the closest result to a considerable p value is not concluding enough.

However, an interesting result present on our experiments was a decrease of 25,78% of editing time between the first and second tasks. It only makes sense though, since the users get experienced to the interface and can then perform the job faster.

We'll now look into how the quality has changed between interfaces. We want to make sure that quality increases or maintains, to see if the new feature comes with a cost.

For quality we used BLEU score comparing each individual test in a systematic format, with the baseline (also in the same format) we got from an expert linguist [40]. She completed the tasks just like the other participants, having access to the same interface and guidelines.

The results from the average BLEU score per interface can be seen in the latter:



Average BLEU score per task	Task 1	Task 2
Word-mapping	79,61	82,31
Sentence-level Sentence-mapping	80,54	83,45
Sentence-level Word-mapping	78,65	81,34

Table 9.5: Results from average BLEU scores

Median BLEU score per task	Task 1	Task 2
Word-mapping	79,65	81,20
Sentence-level Sentence-mapping	80,10	82,95
Sentence-level Word-mapping	77,75	81,35

Table 9.6: Results of median BLEU scores

Here the results are fairly similar between all the interfaces with some small discrepancies between the three. We can see that the best results go from the sentence-level sentence-mapping followed by word-mapping and finally sentence-level word-mapping. We assume that the difference in 1 BLEU point from the best to second best scored is not a significant variation to be taken as a considerable decrease in quality. This can simply be because some participants missed some guideline step while doing the editing, but this will be further discussed in the Discussion session. Also worth mentioning, that BLEU is not the best metric to be used in quality, as we have explained, but due to high costs to analyse these 56 tasks with MQM (2.8k), we'll just use this automatic metric.

We'll now look into how much editing was done by the participants using WER comparing the produced work and the ASR from the videos. We will then correlate this with the quality results, and see if we can find a connection between the two. To start we'll look into a average WER comparison of edits per interface.

From the averages, it seems that the second interface has less edits then the other two. However, looking at the data without considering averages, about 60% of the Word-mapping tasks have more edits than the other 2 interfaces. This can be explained due to the fact that the segmentation present in sentence-level makes it so that, for example, there is no need to put a space after a comma, if the comma is at the end of the sentence. Because of that, this space which would count as an addition for the WER formula in Word-mapping, is not considered here.

Coefficient of variation of BLEU score per task	Task 1	Task 2
Word-mapping	0.05	0.05
Sentence-level Sentence-mapping	0.02	0.04
Sentence-level Word-mapping	0.33	0.42

Table 9.7: Coefficient of Variation on BLEU scores

Average WER score per task	Task 1	Task 2
Word-mapping	8,87%	9,05%
Sentence-level Sentence-mapping	6,43%	6,42%
Sentence-level Word-mapping	9,47%	10,03%

Table 9.8: Results from average WER scores

Task 1	Average video clicks	Average text clicks
With shortcuts	9,83	63,6
Without shortcuts	88,87	101,6

Table 9.9: Average clicks with and without shortcuts in the first task

With this realization after looking at the results, we concluded that no true conclusion could be drawn from the analysis of correlating these two metrics.

The following plot shows the distribution of WER scores with BLEU, which in the end, gave no conclusive results using Pearson values.

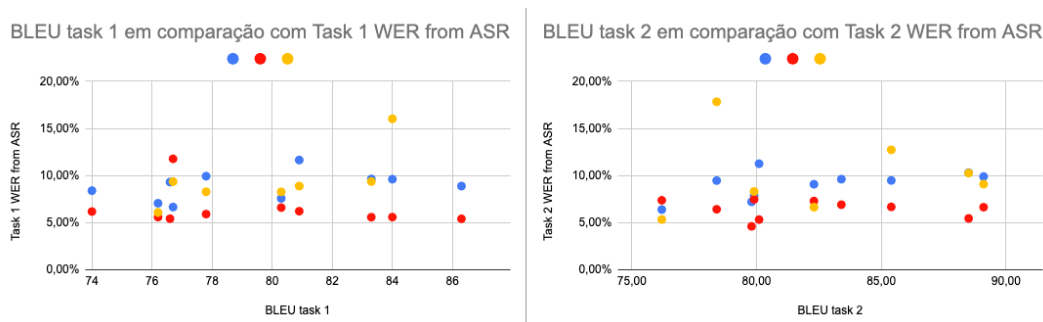


Figure 9.3: Correlation between WER and BLEU scores where Word-mapping is blue, Sentence-level red, and sentence-level Word-mapping yellow.

As we can see, the results are reasonably mixed, and so, we can conclude that WER and BLEU have no correlation in our experiment.

In the experiment, we gave users the chance to use keyboard shortcuts. Although we didn't mention them in the briefing, before they initiated the task, the user tour gave indication that keyboard shortcuts would help their editing process. Luckily, because we recorded these sessions, we can see who of these opened the keyboard shortcuts menu and used them. Let's then see, if keyboard usage saved some clicks in the experiments.

We will now compare mouse clicks between the users that used and did not use shortcuts. The following tables show the number of clicks in the two main areas of the interface that required clicks: the video player, and the text area.

As we can see, there is a sharp drop in mouse clicks when using shortcuts. About 61.36% and 50.2% for task 1 and task 2 respectively. There was even one user that had in his session a total amount

Task 2	Average video clicks	Average text clicks
With shortcuts	14,16	70,52
Without shortcuts	72,16	97,58

Table 9.10: Average clicks with and without shortcuts in the second task

Video turnaround-time average	Task 1	Task 2
With shortcuts	0:10:32	0:06:04
Without shortcuts	0:09:43	0:05:19

Table 9.11: Turnaround-time comparison between users that used and did not use keyboard shortcuts

of zero video clicks, using solely shortcuts to manipulate the video.

Does this mean then that keyboard shortcut usage decreased editing time? Unfortunately no; as we can see in the following table comparing users that used shortcuts and those who did not.

Our first assumption was that having keyboard shortcuts would increase editing speed right from the beginning, but according to [41, 42], only heavily used interfaces by the same individual would start seeing an improvement. From the start, mouse clicks seem to be faster, but keyboard shortcuts have a quick improvement with usage. We can only suppose then, that with more interface practice, this discrepancy would start being noticeable.

# 9.1 Unbabel's pipeline

To have a better understanding of how word-mapping impacted Unbabel's own pipeline we tracked editing times from editors, doing around one thousand tasks since the release of word-mapping in December 2019, until April of 2020. The following graph is a representation of those times, since the beginning of 2019.

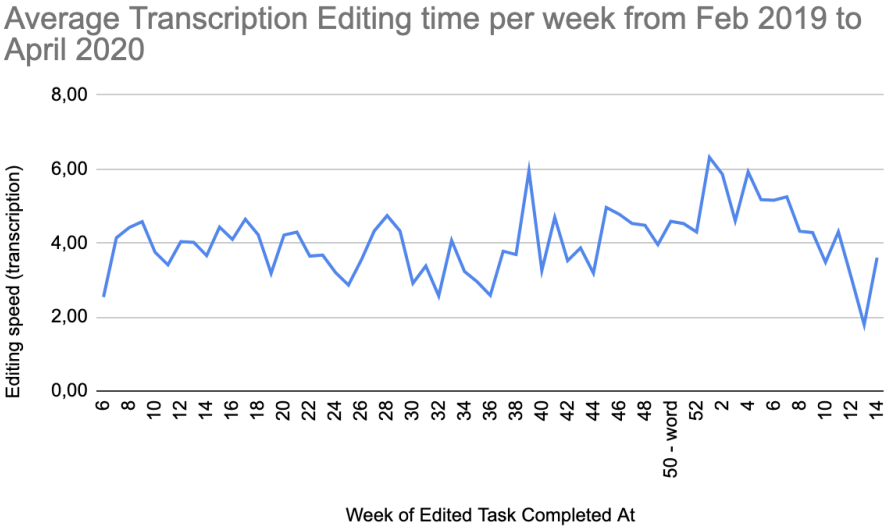


Figure 9.4: Average time per minute of video in Transcription per week. Notice at week 50 word-mapping of 2019 put live.

In the second week of December 2019, Word-mapping was released, and we saw an increase of editing time, the opposite of what we were expecting, which was a bit demoralizing. Even so, we assumed that due to the differences in interaction, editors were learning how to interact with this new interface, so we decided to wait a couple more weeks to understand if the editing times would maintain. We then started seeing at the beginning of 2020, a steady decrease in editing time, which reached the lowest ever at the end of March. Unfortunately, we cannot take any real conclusions from Figure 9.4 since we needed a couple more weeks to study the deviations of the editing times and see if they would maintain or increase. From the time where we're writing this analysis, transcription was mostly deprecated in Unbabel, due to some internal company changes. Maybe, in the course of a few months, when transcription is again pursued, we might take some more conclusive results.

# Chapter 10

## Discussion

After building a state of the art interfaces for transcription and captioning that went head to head with competitors (according to internal data, not in the scope of this thesis), we still wanted to pursue something better. Implementing a misleadingly simple feature such as word-mapping, that took a couple months to nail down, was our way of trying to push the boundaries of editor efficiency when transcribing.

Transcription is the first step when trying to caption a video, therefore we see transcription as a pillar where it creates a foundation to what comes next. So, after producing the captioning tool, which we personally consider is on par with all the existing tools in the market according to internal experiments made, we solely focused on trying to improve further what came as a step before.

On finishing our experiments in this thesis it was evident, that the values we managed to acquire were not conclusive enough to take any considerable result. In the end, the use of the third interface - sentence-level word-mapping - was a "curiosity" that cost us time and which ended up only having inconclusive outcomes. In retrospect, it would have been of more value to have done a 15/15 distribution between word-mapping and the sentence-level approach which, ultimately, could have given us more conclusive values.

We also overlooked the fact that WER was not going to give us conclusive results due to the fact that editors don't need to add spaces at the end of each line in sentence-level approaches. This ended up giving us a considerable different WER through word and sentence approaches, which led us to not having enough data to make a correlation. Quality on the other hand, seemed to be undisturbed all across.

In the end, it is undeniable that the results were not determinant of any meaningful results. Even so, looking into Figure 9.4, we cannot help but speculate that the decrease in the first weeks of 2020 were due to Word-mapping having been implemented.

Conclusively, we understand that for this experiment we should have had more participants and have ignored the third interface comparing only word-mapping "against" sentence-level sentence-mapping (our old approach in 2019). Not only that, we were a few weeks away from concluding from the data of Figure 9.4 if there was going to be a considerable increase or stabilization of editing times in our interface, which is just lamentable.

We cannot end up however, sharing as a personal note, that we still believe that Word-Mapping is the future of Transcription Interfaces. Trint as one of the best Transcription platforms in the market [43, 44] has Word-Mapping included in their transcription interface, which could only mean that they also see value in this approach.

# Chapter 11

## Conclusions

It is undeniable that video and audio content is increasing in volume and search across different platforms. And not only that, transcription and caption providers are looking into multiple ways of keeping up with this progressive demand from the market looking into ways of making their prices and quality as appealing as possible for customers.

In this way, in recent years, companies have looked into ways of making their editors more efficient with the use of Machine Learning technologies namely ASR. The mixture of this with transcription or captioning interfaces has seen light in the recent past, with some promising results. As such, in Unbabel we have built what we believe is at the forefront of this technology mix.

Having approaches such as the split, separating transcription and captioning jobs, showed us a considerable improvement in quality and speed to produce captions. Not only that, we take a new format of ASR technology and design an interface around it, trying to understand how far can we take editors to be empowered to transcribe. Although without conclusive results, we can see a very promising understanding of what could be a innovative approach in how to display and interact with text in a transcription interface.

As a closing remark, I feel the obligation to mention the things I acquired while I partook this project. It taught me how to build a production ready service to be used by hundreds if not thousands of users. Writing simple and efficient code with state of the art frameworks and methodologies.

To think as a team. No interface is complete without good design, back-end, and users. All of these were fundamental to the success of the project. As such, it's imperative to mention that what I learned was far from what I imagined when I started. Everything from product management, community support and acquisition, back-end, front-end, and a great deal of design. Without good design, the features can be implemented in the most efficient code, but if the end user is not on-board, the feature is useless in the end. That was probably my biggest learning.

With these teaching, I undoubtedly became a way better professional and communicator. And that is what I take.

# Bibliography

- [1] Cisco vni forecast. [https://www.cisco.com/c/m/en\\_us/solutions/service-provider/vni-forecast-highlights.html](https://www.cisco.com/c/m/en_us/solutions/service-provider/vni-forecast-highlights.html).
- [2] Hubspot marketing statistics. [https://www.researchgate.net/publication/300855510\\_Introduction\\_Audiovisual\\_translation\\_comes\\_of\\_age](https://www.researchgate.net/publication/300855510_Introduction_Audiovisual_translation_comes_of_age).
- [3] J. Diaz-Cintas. *Introduction: Audiovisual translation comes of age*, pages 1–9. 01 2008. ISBN 978-90-272-1687-8. doi: 10.1075/btl.78.02dia.
- [4] IndieWire. Subtitles vs. dubbing: The big business of translating foreign films in a post-‘parasite’ world. 2020.
- [5] youtube daily watching time statistic. <https://youtube.googleblog.com/2017/02/you-know-whats-cool-billion-hours.html>.
- [6] outbrain marketing content. <https://www.outbrain.com/content-marketing/>.
- [7] comscore online video rankings. [https://www.comscore.com/Insights/Press-Releases/2012/1/comScore-Releases-December-2011-US-Online-Video-Rankings?cs\\_edgescape\\_cc=US](https://www.comscore.com/Insights/Press-Releases/2012/1/comScore-Releases-December-2011-US-Online-Video-Rankings?cs_edgescape_cc=US).
- [8] codefuel marketing statistics. <https://www.codefuel.com/blog/video-marketing-statistics-for-2015-the-next-big-thing-is-here/>.
- [9] No sound on facebook videos. <https://digiday.com/media/silent-world-facebook-video/>.
- [10] M. A. Gernsbacher. Video Captions Benefit Everyone. *Policy Insights from the Behavioral and Brain Sciences*, 2(1):195–202, 2015. ISSN 23727330. doi: 10.1177/2372732215602130.
- [11] D. Cintas and A. Remael. *Audiovisual Translation: Subtiling*. 2014. ISBN 9781900650953.
- [12] P. S. Koskinen, R. M. Wilson, and C. J. Jensema. Using Closed-Captioned Television in the Teaching of Reading to Deaf Students. *American Annals of the Deaf*, 131(1):43–46, 2013. doi: 10.1353/aad.2012.0751.
- [13] B. Dallas, A. McCarthy, and G. Long. Examining the Educational Benefits of and Attitudes Toward Closed-Captioning Among Undergraduate Students. *Journal of the Scholarship of Teaching and Learning*, 16(2):56, 2016. ISSN 1527-9316. doi: 10.14434/josotl.v16i2.19267.



- [14] B. Kothari and T. Bandyopadhyay. Same Language Subtitling of Bollywood Film Songs on TV: Effects on Literacy. *Information Technologies & International Development*, 10(4):31–47, 2014. ISSN 15447529. URL <http://itidjournal.org/itid>.
- [15] Y. Gaur, W. S. Lasecki, F. Metze, and J. P. Bigham. The effects of automatic speech recognition quality on human transcription latency. pages 1–8, 2016. doi: 10.1145/2899475.2899478.
- [16] transcribe.com article about how to write fast. <https://www.transcribe.com/article/transcribe-at-a-fast-pace/>.
- [17] S. Luz, M. Masoodian, B. Rogers, and C. Deering. Interface design strategies for computer-assisted speech transcription. page 203, 2009. doi: 10.1145/1517744.1517812.
- [18] K. Zechner and A. Waibel. Minimizing Word Error Rate in Textual Summaries of Spoken Language. *Proceedings of the 6th Applied Natural Language Processing Conference and the 1st Meeting of the North American Chapter of the Association for Computational Linguistics*, pages 186–193, 2000.
- [19] M. Sperber, G. Neubig, S. Nakamura, and A. Waibel. Optimizing Computer-Assisted Transcription Quality with Iterative User Interfaces. *Language Resources and Evaluation (LREC)*, pages 1986–1992, 2016.
- [20] E. S. Ristad and P. N. yianilos. Learning string-edit distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(5):522–532, 1998. ISSN 01628828. doi: 10.1109/34.682181.
- [21] H. Uszkoreit and A. Lommel. Multidimensional Quality Metrics : A New Unified Paradigm for Human and Machine Translation Quality Assessment. 2012. URL <http://www.qt21.eu/launchpad/sites/default/files/MQM.pdf>.
- [22] U. Muegge. Fully Automatic High Quality Machine Translation of Restricted Text : A Case Study 1 . Project Background 2 . Existing Translation Environment. 2006(November), 2006.
- [23] M. Wald. Captioning multiple speakers using speech recognition to assist disabled people. 2020.
- [24] H. Nanjo, Y. Akita, and T. Kawahara. Computer Assisted Speech Transcription System for Efficient Speech Archive. *Western Pacific Acoustics Conference (WESPAC)*, pages 1–7, 2006.
- [25] L. Wang, T. Hu, P. Liu, and F. Soong. Efficient handwriting correction of speech recognition errors with template constrained posterior (tcp). pages 2659–2662, 01 2008.
- [26] T. J. Hazen. Automatic Alignment and Error Correction of Human Generated Transcripts for Long Speech Recordings. *Interspeech*, pages 1606–1609, 2006.
- [27] Y. Pan, T. Yao, H. Li, and T. Mei. Video captioning with transferred semantic attributes. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017-January: 984–992, 2017. doi: 10.1109/CVPR.2017.111.

- [28] P. Pan, Z. Xu, Y. Yang, F. Wu, and Y. Zhuang. Hierarchical Recurrent Neural Encoder for Video Representation with Application to Captioning. page 33, 2015. URL <http://arxiv.org/abs/1511.03476>.
- [29] Rev tutorial video. <https://www.youtube.com/watch?v=96YMV49MNFk>.
- [30] Respeaking bbc. <http://www.intralinea.org/specials/article/Respeaking-for-the-BBC>.
- [31] Crowdsourcing correction of speech recognition captioning errors. <https://eprints.soton.ac.uk/272430/1/crowdsourcecaptioningw4allCRv2.pdf>.
- [32] A productivity test of statistical machine translation post-editing in a typical localisation context. <https://www.degruyter.com/downloadpdf/j/pralin.2010.93.issue-1/v10108-010-0010-x/v10108-010-0010-x.pdf>.
- [33] Vue website, <https://vuejs.org/>.
- [34] Unbabel interfaces, <https://video.unbabel.com/editor>.
- [35] Ab testing wikipedia definition. [https://en.wikipedia.org/wiki/A/B\\_testing](https://en.wikipedia.org/wiki/A/B_testing).
- [36] Cambridge english test. <https://www.cambridgeenglish.org/test-your-english/general-english/>.
- [37] Zoom.us website. <https://zoom.us/>.
- [38] Fullstory website. <https://www.fullstory.com/>.
- [39] Helena moniz profile. <https://unbabel.com/research/people/helena-moniz/>.
- [40] Vera cabarrão. <https://clul.ulisboa.pt/pessoa/veracabarrao>.
- [41] E. Y. D. S. Craig S. Miller, Richard C Omanson. Comparison of mouse and keyboard efficiency. 2010.
- [42] R. C. O. Craig S. Miller, Svetlin Denkov. Categorization costs for hierarchical keyboard commands. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2011.
- [43] We tried audio transcription software trint. here's what we found , <https://multimedia.journalism.berkeley.edu/blog/audio-transcription-software-trint-review/> .
- [44] Trint review, <https://www.youtube.com/watch?v=96ymv49mnfk> .

