Rethinking Video Interfaces for Usability and Editor's Performance

Miguel Borges Ribeiro
mbribeiro95@gmail.com

Instituto Superior Técnico, Lisboa, Portugal

October 2020

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

1. Introduction

According to Cisco data in 2020, 80% of the content consumed online will be video [3]. 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 [5].

And currently, when we discuss videos, we instantly think of online videos. The reason for such immediate though 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 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 [10].
- One-third of online activity is spent watching video. [5]
- 85% of the US internet audience watches videos online [4].

It is also worthwhile mentioning that another critical piece of marketing data also shows that: 85% of Facebook videos are watched without sound [6]. This is an astonishing amount of 8 billion views per day which is possible mostly 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.

But it is worth noticing that video captions benefit every human being [14]. Captions are especially important for the deaf and hard-of-hearing [11], they benefit children that are learning how to read [16, 12] and people that are learning a new language [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.

2. Problem statement

Companies, and video producers, keep looking for the cheapest and fastest providers of transcriptions and captions, while trying to increase the quality and reducing the costs. And since the demand keeps increasing, providers want to able to keep up with such demands 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.

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 [13, 7, 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.

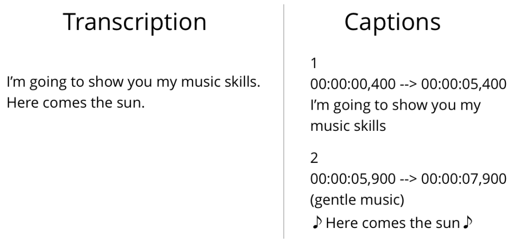
And so, the core research question that I will explore is: ***do standard video platforms allow the editor to perform their best in terms of speed and quality?*** The standard platforms tackle video as a monolithic and solid task with a single platform to do captions, without separating the complex stages into distinct phases.

3. Main Concepts and Definitions

Up to this moment, we have been using 3 main concepts, which will be clarified in this extended abstract - Transcription, Captions and Subtitles.

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

I'm going to show you my music skills.
Here comes the sun.

Captions

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

In the right-hand example of Figure 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.

3.1. Measuring Quality

We'll now look into the quality metrics used in transcription and captioning to access its quality level. These measurements are usually made by annotators which are natives or proficient linguists of the corresponding languages.

3.1.1 Word Error Rate

The Word Error Rate (WER) is a common metric of performance to measure ASR transcription systems [19]. 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 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.

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 50%. That means that the end result was 50% different then the ASR produced.

3.1.2 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.

3.2. Turnaround-time

Not related to these, is another concept which is one of the key points in this thesis, turnaround-time [18]. 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.

4. State of the art

In this section, we will see what features are usually present in transcription and captioning interfaces, and describe how ASR technologies are currently being used. We will analyze features present in both and see how they can help to achieve the best possible quality.

4.1. 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.

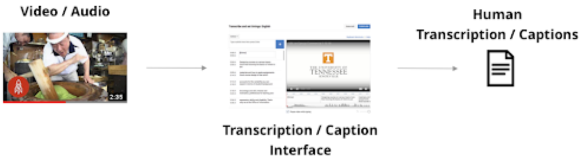

Video / Audio
Human Transcription / Captions
Transcription / Caption Interface

Figure 3: Standard workflow by transcribers and captioners

4.2. 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 [1] and multiple others have taken their approach into making these interfaces, but they all share multiple features together.

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, soundbar), 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 [7]. 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. Both playback and other keyboard short-

[1] https://www.rev.com/, https://trint.com/ ,https://www.descript.com/

cuts 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.

4.3. 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 [17]. 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.

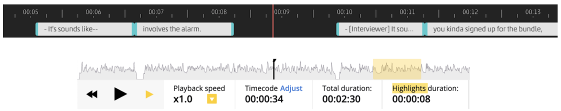


Figure 4: Rev's captioning timeline on top, and Trint timeline on the bottom with waveform

4.4. 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.

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 recognition, 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.

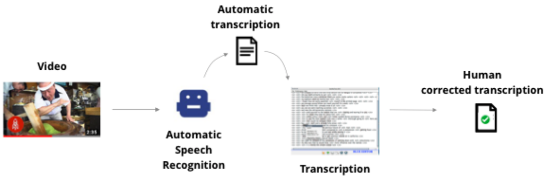


Figure 5: Proposed pipeline using ASR as a first step in [15]

A similar study was done in [19], where researcher found that a From-Scratch approach was "clearly outperformed", by ones which started with an ASR.

But, taking into account the findings in [13], it is better to start a transcription from scratch if the WER is higher than 30%. Which leads us to assume that it is not always preferable to start with an ASR baseline.

5. Unbabel's Pipeline Overview

After much research and experiments with some modified pipelines, this was the one which produced the best results:

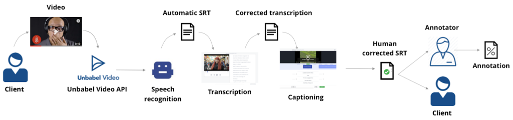
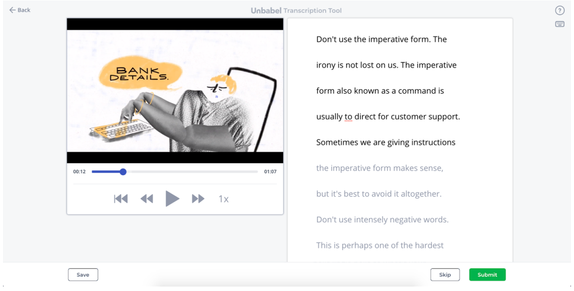


Figure 6: Unbabel captioning pipeline

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. 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.

5.1. Transcription Tool



Don't use the imperative form. The

irony is not lost on us. The imperative

form also known as a command is

usually to direct for customer support.

Sometimes we are giving instructions

the imperative form makes sense,

but it's best to avoid it altogether.

Don't use intensely negative words.

This is perhaps one of the hardest

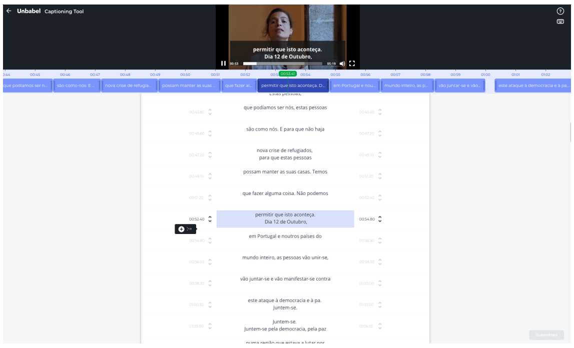
Figure 7: Unbabel transcription interface [8]

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.

5.1.1 Captioning Interface

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.



Estas pessoas,

00:43.80 que podíamos ser nós, estas pessoas 00:45.60

00:45.60 são como nós. E para que não haja 00:47.20

00:47.20 nova crise de refugiados,
para que estas pessoas 00:49.10

00:49.10 possam manter as suas casas. Temos 00:51.20

00:51.20 que fazer alguma coisa. Não podemos 00:52.40

00:52.40 permitir que isto aconteça.
Dia 12 de Outubro, 00:54.80

00:54.80 em Portugal e noutros países do 00:56.50

00:56.50 mundo inteiro, as pessoas vão unir-se, 00:58.30

00:58.30 vão juntar-se e vão manifestar-se contra 01:00.00

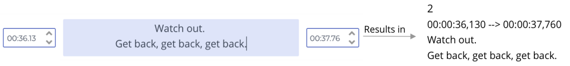
01:00.30 este ataque à democracia e à pa.
Juntem-se. 01:03.00

01:03.00 Juntem-se.
Juntem-se pela democracia, pela paz 01:06.10

numa região que estava a lutar por

Figure 8: Unbabel captioning interface [8]

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:



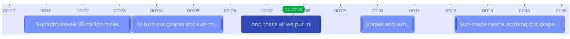
Watch out.
Get back, get back, get back.

```
2
00:00:36,130 --> 00:00:37,760
Watch out.
Get back, get back, get back.
```

Figure 9: 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.

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. A more detailed image of the timeline can be seen in the following figure.



Sunlight travels 93 million miles,

to turn our grapes into sun-m...

And that's all we put in!

Grapes and sun...

Sun-made raisins, nothing but grape...

Figure 10: Unbabel's timeline in the Captioning Tool

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.

5.2. 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 11). Simply put, one has sentences timestamped, and the other has individual words. As you can see in Figure 7, the text seems to be split on a sentence level.

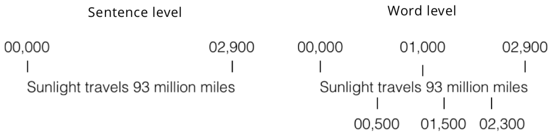


Figure 11: Visual demonstration of sentence level and word level ASR

The suggestion then, is to decrease the level of granularity of the produced ASR, so that editors can work faster. This was 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.

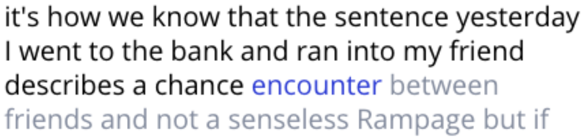
5.3. Word-Mapping

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.

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.

5.3.1 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 12. 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 12: 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.

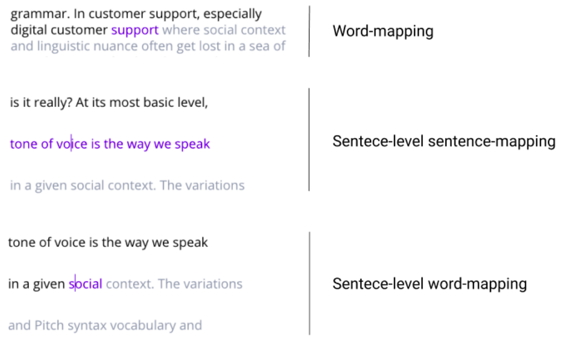
This text will then pass by an external service called Speechmatics [2], which will chunk the text into small captions taking into consideration syntax and audio. Those captions are then, to be further adjusted in the Captioning Tool.

5.3.2 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 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:

[2] https://www.speechmatics.com/



grammar. In customer support, especially digital customer support where social context and linguistic nuance often get lost in a sea of

is it really? At its most basic level,

tone of voice is the way we speak

in a given social context. The variations

tone of voice is the way we speak

in a given social context. The variations

and Pitch syntax vocabulary and

Figure 13: 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.

6. Thesis experiments

In order to test the efficiency of the tools, we'll use a famous experience research methodology: A/B testing [1]. 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 [2], with the category "General English".

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.

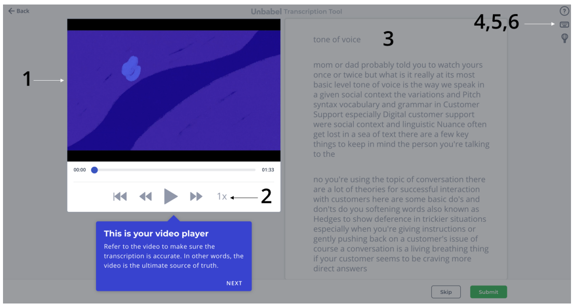
For the experiments we are running, it is mandatory to give the testers the least amount of information possible, but still, making them empowered enough to produce high level quality. For this, we had to meticulously choose the videos we were going to use. Not only to reduce the amount of guidelines these testers had to read, but we also needed to consider the level of difficulty of the video and its length.

We ended selecting a video with 11% WER, that had only 5 instances of rules to know from the official transcription guidelines.

6.1. 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:



tone of voice

mom or dad probably told you to watch yours once or twice but what is it really at its most basic level tone of voice is the way we speak in a given social context the variations and Pitch syntax vocabulary and grammar in Customer Support especially Digital customer support were social context and linguistic Nuance often get lost in a sea of text there are a few key things to keep in mind the person you're talking to the

no you're using the topic of conversation there are a lot of theories for successful interaction with customers here are some basic do's and don'ts do you softening words also known as Hedges to show deference in trickier situations especially when you're giving instructions or gently pushing back on a customer's issue of course a conversation is a living breathing thing if your customer seems to be craving more direct answers

This is your video player

Refer to the video to make sure the transcription is accurate. In other words, the video is the ultimate source of truth.

NEXT

Figure 14: Example of the first step in the Tour

The numbers on Figure 14 correspond to each highlight which in order - The video player, the playback speed option, the first line of the transcription, guidelines button, keyboard shortcuts, and finally a button to start the tour 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.

7. Results

The experiment was performed by 30 different participants.

And starting with the English test, all participants passed with ease.

But 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 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:

Average TAT	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 1: Results from average time taken (Turnaround-time)

Median TAT	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 2: Results from median time taken (Turnaround-time)

Coefficient of variation TAT	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 3: Results from coefficient of variation on time taken (Turnaround-time)

We can see from Table 1 that word-mapping has the best turnaround-time [3] 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:

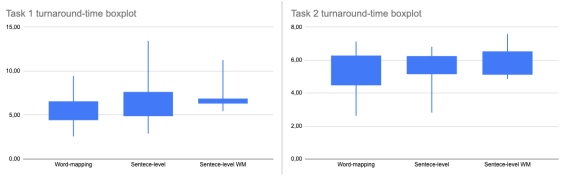


Figure 15: Turnaround-time boxplots

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.

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

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

Table 4: Results from T-student on time taken

As we can conclude from Table 4, no considerable findings can be taken from turnaround-time efficiency as the closest result to a considerable p value is not concluding enough.

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 [9]. 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 table 5.

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

[3] Time from when a user starts a task until he delivers it

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 5: Results from average BLEU scores

Coefficient of variation-BLEU	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 6: Coefficient of Variation on BLEU scores

significant variation to be taken as a considerable decrease in quality.

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.

Average - WER	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 7: Results from average WER scores

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.

But after analysing WER with BLEU, the scores gave no conclusive results.

7.1. 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.

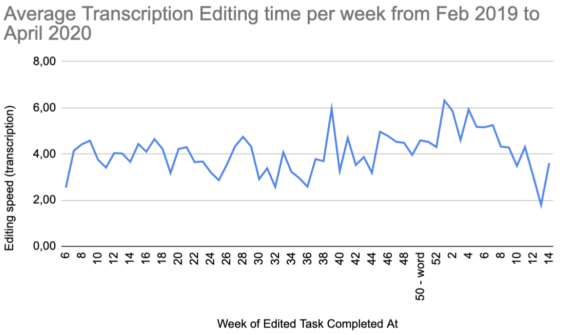


Figure 16: Average time per minute of video in Transcription per week. Notice at week 50 word-mapping of 2019 put live.

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 16 since we needed a couple more weeks to study the deviations of the editing times and see if they would maintain or increase.

8. Discussion

After building a state of the art interfaces for transcription we still wanted to pursue something better.

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 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.

In the end, it is undeniable that the results were not determinant of any meaningful results. Even so, looking into Figure 16, 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 16 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 has Word-Mapping included in their transcription interface, which could only mean that they also see value in this approach.

9. 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.

References

[1] Ab testing wikipedia definition. https://en.wikipedia.org/wiki/A/B_testing.

[2] Cambridge english test. https://tinyurl.com/yc49w7bx.

[3] Cisco vni forecast. https://tinyurl.com/yxrv4ths.

[4] comscore online video rankings. https://tinyurl.com/yybpdpwt.

[5] Hubspot marketing statistics. https://tinyurl.com/y6qepka3.

[6] No sound on facebook videos. https://tinyurl.com/y7uwdywc.

[7] transcribe.com article about how to write fast. https://www.transcribe.com/article/transcribe-at-a-fast-pace/.

[8] Unbabel interfaces, https://video.unbabel.com/editor.

[9] Vera cabarrão. https://clul.ulisboa.pt/pessoa/veracabarrao.

[10] youtube daily watching time statistic. https://tinyurl.com/y36jvrc.

[11] D. Cintas and A. Remael. *Audiovisual Translation: Subtiling.* 2014.

[12] 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.

[13] 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.

[14] M. A. Gernsbacher. Video Captions Benefit Everyone. *Policy Insights from the Behavioral and Brain Sciences*, 2(1):195–202, 2015.

[15] T. J. Hazen. Automatic Alignment and Error Correction of Human Generated Transcripts for Long Speech Recordings. *Interspeech*, pages 1606–1609, 2006.

[16] 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.

[17] S. Luz, M. Masoodian, B. Rogers, and C. Deering. Interface design strategies for computer-assisted speech transcription. page 203, 2009.

[18] U. Muegge. Fully Automatic High Quality Machine Translation of Restricted Text : A Case Study 1 . Project Background 2 . Existing Translation Environment. 2006(November), 2006.

[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.