Computing (2023) 105:1467–1495
https://doi.org/10.1007/s00607-023-01156-y

REGULAR PAPER

Synchro-Sub, an adaptive multi-algorithm framework for real-time subtitling synchronisation of multi-type TV programmes

Jose Manuel Masiello-Ruiz[1] · Belen Ruiz-Mezcua[1] · Paloma Martinez[1] · Israel Gonzalez-Carrasco[1]

Received: 23 September 2021 / Accepted: 10 January 2023 / Published online: 23 January 2023

Abstract

Subtitles are critical elements in making TV as accessible as possible for people with hearing impairment, elderly people and other end-users that could need this kind of service. Moreover, delay in live-originated TV captions or subtitles is a significant factor in the audience's perception of the absence of quality for the deaf and hard of hearing. This lack of synchronisation leads to poor quality broadcasts because of the inconsistency between images and subtitles presented to the viewer. In some cases, this situation leads to a misunderstanding of the message. Subtitling is a reactive process that starts once a minimum of audio information has been generated and ends when an insertion time, deletion time and text are allocated. Therefore, there will always be a delay associated with this process, related to the audio duration being subtitled and the subtitle production time. Moreover, in live TV programmes, unsynchronised audio and subtitles are inherent to the subtitling process because the broadcasting is performed in real-time. In this paper, the authors present the Synchro-Sub framework for synchronising subtitles with the audio-visual contents for live-captioned TV programs. For doing this, and considering the knowledge obtained in previous approaches defined by the authors, several improvements have been included in the transcription and *chronization* phases of the framework that increase the percentage of correctly timed subtitles.

✉ Israel Gonzalez-Carrasco
igcarras@inf.uc3m.es

Jose Manuel Masiello-Ruiz
jmasiell@eco.uc3m.es

Belen Ruiz-Mezcua
bruiz@inf.uc3m.es

Paloma Martinez
pmf@inf.uc3m.es

[1] Computer Science and Engineering Department, Universidad Carlos III de Madrid, Av. Universidad, 30, 28911 Leganés, Madrid, Spain

Keywords Accessibility · TV broadcasting · Synchronisation · Automatic Speech Recognition

Mathematics Subject Classification 68T10 · 68W01 · 68N01

1 Introduction

Subtitles are essential elements in making TV as accessible as possible for people with hearing impairment, elderly people and other end-users that could need this kind of service [1]. Moreover, delays in synchronisation decrease the quality of live-originated TV captions or subtitles for the deaf and hard of hearing [2]. An inadequate synchronisation leads to poor-quality broadcasts because of the mismatch between images and subtitles presented to the viewer. In some cases, this situation leads to a loss of comprehensibility of the message.

This paper proposes an improvement of Sub-Sync framework presented in [3]. Sub-Sync was focused on the process of re-situating the appearance times (insert) and deletion (erase) of the subtitles, which ensured synchronisation between them and the audio-visual elements. To achieve this synchronisation, Sub-Sync included a three-phase process. The first phase, called transcription, uses an Automatic Speech Recognizer (ASR) engine to produce a continuous word flow that corresponds with the transcription of the audio-visual's audio. The target of the second phase is to identify the correct presentation time for each subtitle. The authors called this phase chronization. For this alignment, the algorithms included were Needleman-Wunsch [4] and the inertia algorithm. Finally, the third phase harmonises subtitles and the audio-visual contents and is called synchronisation. Moreover, in the case of Sub-Sync, there were only two algorithms available, and this framework did not use the interpolation algorithm.

In this article, the authors propose a new synchronisation framework based on the insights and knowledge extracted from previous research: Synchro-Sub. Figure 1 shows the three phases of the new Synchro-Sub framework and the main improvements performed for each phase. The transcription phase has been extended and reorganised into three components: audio sampling, an ASR and the generation of the continuous stream of words with time stamps. The audio sampling component includes a fine-tuning process for configuring various parameters: format, number of channels, sample rate and chunk size. The selected configuration must be compatible with the ASR configuration. In addition, the architecture supports any ASR system on the market with an adequate level of recognition in a reasonably short processing time (typically less than one second). The framework can process a continuous stream of data with continuous real-time response. For generating the stream of words, the third component consolidates the different attempts into a single stream and assigns a time stamp to each word. The algorithm can distinguish repeated words by assigning to each repetition the instant of its first occurrence. Finally, this transcription phase has been improved by including a stability index (r). By doing this, the framework considers the words and the moment when they appear, even if they subsequently disappear in successive attempts.

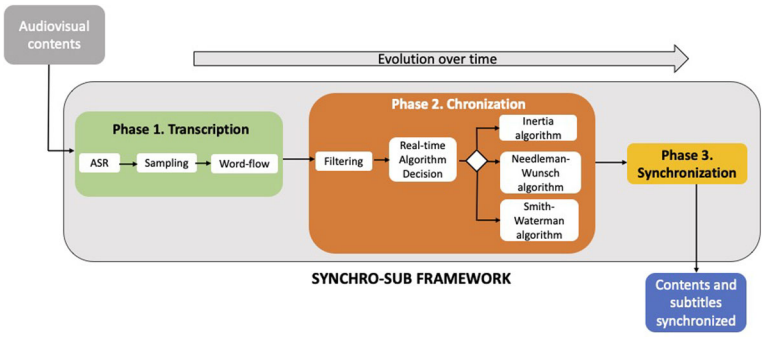

Fig. 1 Phases of the framework (blue boxes) and overview of the main improvements (white boxes)

In the new *chronization* stage of Synchro-Sub, the association between the transcript and a subtitle is determined by identifying the best alignment between the word sequences of the subtitle and the transcript. Once an alignment has been obtained, the quality of the alignment is assessed, and a frame of reference is established to decide whether the alignment obtained exceeds a minimum to consider the association valid. Finally, the part of the transcript sequence that will remain non-associated and can be used later is determined. To achieve this goal, different improvements have been included in Synchro-Sub. The first is when the synchronisation attempt is launched and the inertia algorithm is applied. Secondly, synchronisation attempts are made only on subtitles that have an instant shorter than the synchronisation instant and do not force the inertia algorithm until a certain time threshold is exceeded, which is the time corresponding to the synchronisation limit before the delay becomes excessive.

Moreover, before the alignment process, and considering the feedback obtained in the Sub-Sync framework, a new filtering process has been implemented on the texts of the transcription and the subtitles: the texts have been standardised by eliminating punctuation marks and accents and converting capital letters into lower case. In addition, the architecture makes it possible to activate a filter to remove words shorter than a given length, with the idea that shorter words are more frequent and less semantically loaded and can lead to misalignments. For the Spanish language, this length has been determined as two, but words of this length such as "es", "se", "si", "ha", "ir", "va", "he", "he", "he", etc. are incorporated as exceptions.

Regarding alignment between subtitle and transcription, two new algorithms have been included in the Synchro-Sub framework. The Needleman-Wunsch (NW) algorithm is proposed for detecting a global alignment [4]. However, because the transcript is usually much longer than the subtitle and a segment of a TV programme is generally focused on a specific topic, a sequence of words is likely repeated, which can lead to false alignments between a subtitle and transcripts that correspond to later subtitles.

Furthermore, a modification of the NW algorithm is proposed for selecting the alignments closest to the beginning of the transcript (NWA). Besides, the Smith-Waterman (SW) algorithm is proposed to detect possible local alignments [5].

The final stage of the process is to synchronise subtitles and the audio-visual content, either in broadcasting or in the reception. This stage focuses on adding the delay applied to the audio-visual to the timestamps recalculated by the chronizator.

Therefore, Synchro-Sub defines a new approach for synchronising the subtitles with the audio-visual contents for live-captioned TV programs by automatically adjusting to the time-ubication. For doing this, and taking into account the knowledge obtained in [3] and [1], several improvements have been defined in the three-phase process.

The remaining of this article is organised as follows: Section 2 summarises the necessary background on TV subtitles services, quality parameters and synchronisation algorithms. Section 3 presents the architecture of the Synchro-Sub framework focusing on the main improvements incorporated. Section 4 presents the experimentation and results obtained in five different videos and a comparison against the tool Sub-Sync. Finally, Section 5 discusses the conclusions and future lines of research.

2 Literature Review

Taking into account the world disability report [6], there are over one thousand million people with at least one type of disability. Moreover, more than 66 million people suffer from hearing impairment, and this disability causes them difficulty in understanding audiovisual content due to the loss of audio information [7]. The incorporation of subtitles/close captions in multimedia content is today the most largely used solution to this problem [8]. In this sense, subtitles are essential not only for deaf people but also for all content consumers, as it helps to improve comprehension [9, 10].

In subtitle synchronisation, the authors have been working on previous approaches for dealing with this situation. The authors compare Needleman-Wunsch and Levenshtein algorithms in [3]. Moreover, another related approach was presented for subtitle synchronisation in [1]. The authors defined a deep learning architecture based on semantic models and the Needleman-Wunsch algorithm in this case. Those approaches were based on the Needleman-Wunsch algorithm, a dynamic programming algorithm used in the pairwise global alignment of two biological sequences [11, 12]. In the past, this algorithm and its variants have been widely used in various contexts. For example, in bioinformatics, sociology, customer retention, medical diagnosis, and communication, and in recent years, have gradually become accepted by eye-tracking-based studies to analyse scanning path [13].

The Synchro-Sub framework also includes the Smith-Waterman algorithm for adaptive real-time synchronisation. The Smith-Waterman algorithm is a classical method for comparing two strings to identify highly similar sections within them [5]. It is widely used in finding good near-matches, or so-called local alignments, within biological sequences [14]. Moreover, this algorithm for sequence alignment is one of the main tools of bioinformatics. It is used for sequence similarity searches and alignment of similar sequences [15]. Furthermore, the Smith-Waterman algorithm has been used in the text plagiarism detection [16] to assess the correct level of an answer and plagiarism level of answers among subjects [17] or for information retrieval in Electronic Medical Records [18].

The subtitles in TV audiovisual content broadcasts are a challenge from the point of view of quality, especially in live broadcasting. In this context, three main indicators must be taken into account for showing readable and understandable subtitles: speed, literal subtitle (fidelity), and synchronisation [19]. Moreover, previous studies about caption perception of hearing-impaired audiences show that adjusting captions to suitable linguistic level, reading rate and synchronisation can significantly improve the information gained from captions [9].

Synchronisation implies that the subtitle is shown simultaneously with the speaker's speech. However, the delay in the availability of subtitles is disconcerting for the consumer of audiovisual content. Therefore, as synchronisation is one of the quality indicators, it is important to consider this issue in the audiovisual production process.

Television broadcasting has three types of programs: pre-recorded, live-scripted or live-improvised. Another possible scenario is a combination of different types in the same broadcasting (mixed scenario). In a pre-recorded program, subtitles can be generated during the editing period or afterwards, regardless of whether insertion is done for all the viewers (open caption) or only visible for those who select the service (closed caption).

In those programs, the synchronisation of subtitles and audiovisual content can be done carefully before broadcasting using a script. In [7], the authors present a system for automatic subtitle positioning from an associated script designed to enhance the video accessibility of deaf and hearing-impaired people to multimedia documents. Another approach is based on dynamic captioning for synchronising the scripts word-by-word with the speech and highlighting the variation of voice volume [9].

In the same way, when the audiovisual production is simultaneous with its broadcast, and the speakers' interventions are subject to a pre-established script (live-scripted), the subtitles' quality is high as long as the speakers comply with the script. In this context, the literature offers a wide range of techniques for automated synchronisation. In [20], the authors describe a framework for automatically synchronising subtitles with audio in news programs, but this system has been tested only on live-scripted types of programs. Another approach is a simultaneous subtitling system for broadcast news programs with a speech recogniser [21]. In [22], the authors propose an algorithm to synchronise live speech with its corresponding transcription in real-time at a syllabic unit. The proposed algorithm employs the syllable endpoint detection method and the syllable landmark detection method with band-limited intensity features.

However, there is another group of programs where it is impossible to have a rigorous script for producing the subtitles: e.g., in programs in which guests participate, such as debate or interview programs, or those in which the speakers have some freedom to express themselves within the topic to be discussed. This is called a live-improvised scenario: live broadcasts without a previous script.

There are three usual techniques for generating subtitles in this scenario: by typing, stenography, and re-speaking [23, 24]. The most common ones are stenography and re-speaking. Since both the stenotypist and the re-speaker have to follow the speaker, the subtitles are issued "as soon as possible" in an "unsynchronised" way (usually delayed) concerning the audio that produces them. It is necessary to add to this delay the one involved in the logic of the process, as the subtitle cannot be issued until it is complete, which will be no sooner than, in the best scenario, at the end of the

locution [3]. For this scenario, some studies have focused on the automatic generation of subtitles for live broadcast news programs. Still, the scripts are generated in real-time during the broadcast and synchronisation is not taken into account [25, 26]. Another study has been oriented to generating a transcription where multiple speakers appear in the same context [27].

The main issue of this scenario is the confusion produced when subtitles that correspond to the audio of one speaker appear when that speaker is no longer talking and/or the image shows another speaker. Finally, there is a discrepancy between the subtitle and the audiovisual information for the hard of hearing.

Table 1 summarises the main characteristics of the research analysed in this section, focusing on the synchronisation process. In the algorithm column, NW indicates Needleman-Wunsch, LEV is Levenshtein and SW is Smith-Waterman.

This paper focuses on two types of programs: improvised or mixed-live broadcasts. There is no synchronisation between audiovisual contents and subtitles because a temporal displacement is associated with their generation process.

Finally, synchronisation in live broadcasting is a problem dealt with by the authors in two previous studies [3], and [1]. The research presented in this manuscript follows the insights and the knowledge extracted from them.

3 Solution Proposed

For live TV programmes, the problem of unsynchronised audio and subtitles is inherent to the subtitling process. Subtitling is a reactive process that starts once a minimum of audio information has been generated and ends when an insertion time (t_i), deletion time (t_e) and text are allocated. As shown in Fig. 2, there will always be a delay $\Delta_d = T_m + T_p$, where T_m is the minimum audio time and T_p is the subtitle production time.

The delay (Δ_d) will depend on the solution adopted for the subtitle generation and the nature and structure of the TV programme.

In some cases, in programmes where most of the audio is relevant, such as news programmes, the subtitle is an almost verbatim transcription, so the T_m will be small. In other cases, such as entertainment programmes, debates, commercials, etc., the subtitles summarise the audio, and there are even many segments for which no subtitle is generated. In this case, the subtitling process needs more information to determine what is relevant to produce the text, and T_m will increase. This lack of synchronisation causes a low quality in the broadcasts due to the difference between images and subtitles. Additionally, in some cases, the intelligibility of the message is lost.

In this context, the time that a subtitle is delayed (Δ_i) will be equal to the subtitle generation delay (Δ_d) plus the processing time for synchronisation. To determine the value of (Δ_e) it must be considered that for commercial reasons, it must be as small as possible, for technical reasons constant throughout the broadcast and also for all the subtitles of the broadcast, it must be fulfilled:($\Delta_e > \Delta_i$).

The synchronisation process is divided into three phases (1): (i) transcription phase, in which, from the audio stream of the programme, a stream of words with their corresponding timestamps is generated, (ii) timing (*chronization*) phase in which from

Table 1 Comparison of references for subtitle synchronisation

Reference	Year	Algorithm	Scenario	Live broadcasting
[1]	2021	Deep learning and NW	Improvised or mixed-live broadcasts	Yes
[3]	2019	NW and LEV	Improvised or mixed-live broadcasts	Yes
[7]	2019	Subtitle positioning from an associated script	Pre-recorded programs	No
[9]	2010	Dynamic captioning from scripts (word-by-word)	Pre-recorded programs	No
[20]	2009	ASR	Live-scripted programs	Yes
[21]	2003	ASR	Japanese news programs	Yes
[22]	2013	Ad-hoc (syllabic unit)	Live Speech	Yes
[25]	2009	Hidden Markov Model and Viterbi	News programs	Yes
[26]	2000	ASR	Japanese news programs	Yes
[27]	2020	ASR	Multiple Speakers (conferences)	Yes

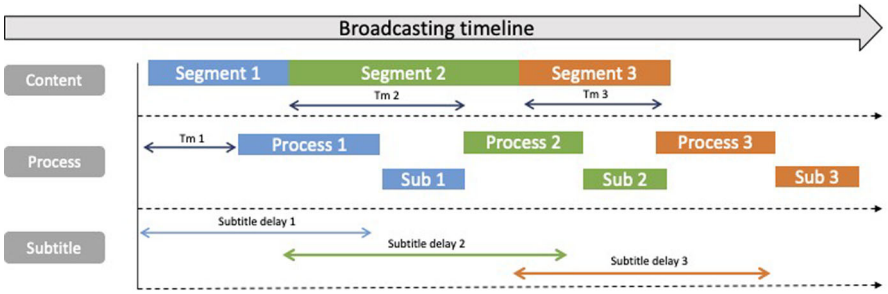

Fig. 2 The subtitle production process always introduces a delay

the transcribed word streams and the subtitles generated by the subtitling process a stream of subtitles with the timestamps is generated and (iii) synchronisation phase in which from the temporarily stored TV programme stream and the timestamped subtitles the synchronised broadcast is produced.

In the *chronization* phase, the objective is to associate, through an automatic process, fragments of transcripts to each of the subtitles and use the transcripts' time stamps to determine the subtitle's times of appearance and disappearance.

Taking the architecture defined in [3] as a reference, this paper proposes a set of improvements in the (i) transcription and (ii) *chronization* phases that improve the percentage of correctly *chronized* subtitles. These improvements are explained in the following sections.

3.1 Transcription

A module organised into three components has been designed to support this phase: audio sampling, ASR and generation of the continuous flow of timestamped words.

All the times handled by the architecture, particularly (t_s), are referenced to the start of the audio sampling. This start should be synchronised with the times of the audiovisual flow. If this synchronisation cannot be ensured by other means, a synchronisation mechanism has been implemented based on a keyword (mark). The instant of appearance in the audiovisual production of this mark is known. Therefore, this mark defines the time of the appearance in the recognition frame, and it must be located within the first seconds of the production. Finally, the keyword is set, and the synchronisation is established for the whole process.

Several parameters can be configured in the sampling process: format, number of channels, sampling rate, and chunk size. The selected configuration must be compatible with the ASR configuration. The audio and commercial products used have been configured for one channel, Int16 sample format, a sample rate of 16kHz and a chunk size of 100 ms. The chunk size will determine the maximum precision of the timing allocation by the ASR.

The architecture supports any commercial ASR product with an adequate level of recognition in a reasonably short processing time (typically less than one second), and it can process a continuous stream of data with continuous real-time response.

Commercial products that process audio in batches are not valid. The ASR must generate timestamps about the audio sample sent: the first sample sent marks instant 0, and the transcription times must be referenced to this sample.

Since most commercial products do not support audio streams without time limitation, the architecture implements a process of restarting the transcription request every so often. When this restart occurs, the ASR will start sending the timestamps referenced to the first audio block sent at that restart. The architecture ensures no loss of coherence in the timestamps of the responses belonging to different restarts. Therefore, all the times are referenced to instant 0 of the initial block of the whole process.

In the continuous flow configuration, the ASR system has a recognition start when a first audio block is sent and starts to respond with a series of frames known as transcription attempts. The incremental attempts may be modified as more audio blocks are sent until the ASR system determines a complete text. Next, the ASR system sends a response frame marked "end" and starts a new frame with the subsequent audios.

The ASR responses come in the form of asynchronous, intermediate or final frames, which contain the following information: the associated instant in milliseconds, the response text containing all the acknowledgement since the last final frame and a stability index (c), between 0 and 1, which indicates the quality of the transcription. The smaller this index is, the more likely subsequent transcriptions will modify the current one.

Figure 3 describe an example of transcription attempts by an ASR. This example shows how certain words that appear in a first attempt disappear until they appear again in the flow. For example, “estás mala” in the instant 1980 with a c=0.01 disappear until they appear again in the instant 2460. In the same way, the word “hacer” appears in 2700, disappears in 3060, and will appear again in 3300 in a stable way. All the times are synchronised with the first audio block and adjusted with the synchronisation keyword mentioned above.

For generating the stream of words, the third component consolidates the attempts into a single stream. Moreover, it also assigns the timestamp to each word (the instant of its first appearance in an attempt). The algorithm can distinguish repeated words by assigning each repetition the instant of its first occurrence.

In the case of unstable words, they appear and disappear in different attempts; this module determines whether the word is considered stable based on the index c and the stability of the subsequent words. The stability index assigned to the word is that associated with the last occurrence of the word.

In addition, the index is calculated for each word.

$$r = n \,/\, \Delta tr \tag{1}$$

where n is the number of times the word appears and Δtr is the number of frames since the first occurrence. This index is updated with each attempt and stabilises once the system gives a sequence of attempts as “final”. Table 2 shows what the output of this module would be for the example in Fig. 3.

	Text	stability
1920	small	c = 0.01
1980	estás mala	c = 0.01
2220	Málaga	c = 0.01
2460	estás mala vamos	c = 0.01
2520	estás mala vamos para	c = 0.01
2700	estás mala vamos para hacer	c = 0.01
2880	estás mala vamos para hacer es	c = 0.01
2940	estás mala vamos para hacer eso	c = 0.01
3060	estás mala vamos	c = 0.90
3120	estás mala vamos para	c = 0.90
3300	estás mala vamos para hacer	c = 0.90
3540	estás mala vamos para hacer eso	c = 0.90
3720	estás mala vamos para hacer eso hay	c = 0.90
3780	estás mala vamos para hacer eso hay que	c = 0.90
4020	estás mala vamos para hacer eso hay que ser	c = 0.90
5580	estás mala vamos para hacer eso hay que ser bastante	c = 0.90
5880	estás mala vamos para hacer eso hay que ser bastante mala	c = 0.90
6000	estás mala vamos para hacer eso hay que ser bastante mala no	c = 0.90
51240	estás mala vamos para ..... podría quedar archivada de nuevo	c = 0.95
53700	si lo	c = 0.01
53820	si lo que	c = 0.01
53940	si lo que se	c = 0.01
54000	si lo que sea	c = 0.01
54120	si	c = 0.90
54240	si	c = 0.90
54300	si lo	c = 0.90
54360	si lo	c = 0.90
54420	si lo que	c = 0.90

Fig. 3 Sequence of attempts and stability factor

As a result of this process, an object "trsASR" is generated, expanded and added as new audio. Therefore, this object contains all the sent words transcribed, temporally ordered and with their index r.

3.2 textitChronization

This phase aims to calculate the subtitles' new insertion and deletion times to improve the coherence of audio and image. Two processes are established: (1) the assignment of a set of transcribed words to a subtitle and (2) the calculation of the insertion and deletion times of the subtitle according to the associated audio time interval.

3.2.1 Time reference

All the entities involved in the process are temporally referenced to the start of the audiovisual production, particularly the timestamps of the subtitles and the words embedded in the trsASR stream.

Table 2 Example of trsASR object content

Word	Timestamp	Stability	r
small	1920	0.01	1/18
estás	1980	0.9	16/17
mala	1980	0.9	16/17
Málaga	2220	0.01	1/16
vamos	2460	0.9	15/15
para	2520	0.9	13/14
hacer	2700	0.9	11/13
es	2880	0.01	1/12
eso	2940	0.9	7/7
hay	3720	0.9	6/6
que	3780	0.9	5/5
ser	4020	0.9	4/4
bastante	5580	0.9	3/3
mala	5880	0.9	2/2
no	6000	0.9	1/1

Let (t_s) be the time elapsed since the start of the transmission, (t_s) the delay introduced in the transmission and (t_s) a safety margin set to ensure that at least no increase in the subtitle delay occurs, a maximum time is defined.

$$t_{max} = t_s - \Delta_e + \Delta_s \tag{2}$$

No subtitle can exceed t_{max} without being timed.

3.2.2 Algorithm for processing subtitles and audio sequences

The inputs to the association process are the stream of words contained in the trsASR and the subtitles supplied by the subtitling system. A set of consecutive words of the tsrASR object is called a fragment.

The model assumes that sound and subtitles follow the same temporal sequence. Therefore, the two sequential fragments assigned follow the same sequence as their associated subtitles. Moreover, all unassigned fragments (D) before the last fragment assigned (link) cannot be used for subsequent assignments. All allocation attempts are made with the unallocated fragment after the last allocated fragment (P) and the rest of the subsequent tsrARS frame.

From a timing point of view, the subtitles can be in two states:

- Pending: when the timing recalculation has not taken place. A subtitle can be in this state as long as: $t_i \leq t_{max}$.
- *chronized*: when the subtitle has been assigned to a fragment and the insertion and deletion times have been recalculated according to this assignment.

The *chronization* process follows the scheme shown in Fig. 4:

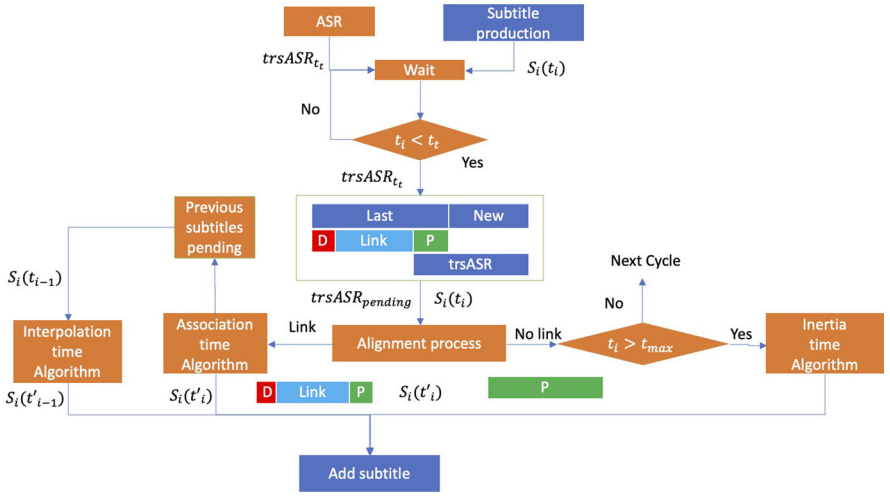

Fig. 4 Diagram of the *chronization* algorithm flow

- The pending subtitles that satisfy that $t_i \leq t_t$ are selected, where t_t is the instant of the last transcribed word.
- For each of the selected subtitles, the connection process (Sect. 3.2.3) is performed. The inputs are the last fragment of the unconnected trsASR object and the selected subtitle. The outputs are the fragment that is associated with the subtitle (link), a fragment formed by previous words and that is discarded (D), and a fragment with later words that is available to be processed with the following subtitles (P) (Fig. 5).
- In the case of an association, the subtitle is timed using the time association algorithm (Sect. 3.2.4). If there is any previous subtitle in a pending state, their respective t'_i, t'_e are calculated using the interpolation algorithm (Sect. 3.2.5).
- If no association occurs and $t_i > t_{max}$ is timed using the inertia algorithm (Sect. 3.2.6).
- In case none of the above occurs, the subtitle remains in a pending state.

This process is repeated when new information is incorporated in trsASR, new subtitles, or periodically to avoid prolonged silences that may cause the t_{max} to be exceeded. For this time, one second is proposed.

3.2.3 Algorithm for the association of audio fragments to subtitles

This algorithm aims to automatically identify and associate a set of temporally ordered words contained in the trsASR object with a given subtitle.

Figure 5 shows a diagram of the association process. The green lines mark the instants of attempted association between the subtitle and the trsASR fragment with which the association is attempted.

The proposed algorithm assumes that both the audio, and therefore the transcript, and the associated subtitle convey essentially the same message, so that: 1) the subtitle

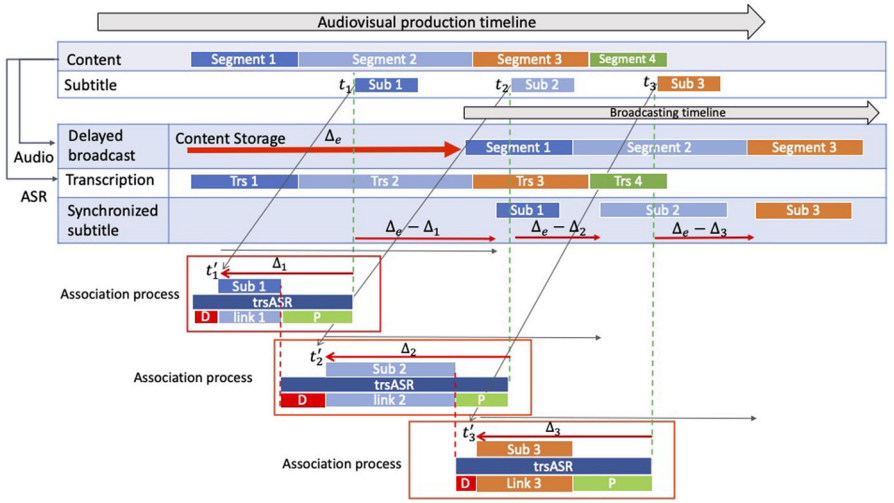

Fig. 5 Diagram of the association process between a subtitle and a transcription

and the associated transcript have some common or similar words 2) the subtitle and the transcript must be correctly from a grammatical point of view. Similar word sequences are expected to occur between fragments of the transcript and a subtitle frequently.

The association between the transcript and a subtitle will be determined by identifying the best alignment between the word sequences of the subtitle and the transcript. Once an alignment is obtained, the quality of the alignment is evaluated. Next, a frame of reference is established to decide if the alignment obtained exceeds a minimum to consider the association valid. Finally, the part of the transcript sequence that will remain non-associated and be used later is determined.

Before the alignment process, a filtering process was implemented on the transcription texts and subtitles: the texts have been normalised by eliminating punctuation marks and accents and converting capital letters into lower case. In addition, the architecture allows activating a filter to eliminate words shorter than a given length, with the idea that shorter words are more frequent and less semantically loaded and can lead to misalignment. For the Spanish language, this length has been determined as 2, but words of this length are incorporated as an exception, such as “es”, “se”, “si”, “ha”, “ir”, “va”, “he”, etc.

In general, the subtitle sequence has a shorter length than the transcription. Therefore, two alignment strategies can be established, one local and the other global. Several strategies were tried in the implemented association process, and the best quality was selected.

The Needleman-Wunch (NW) algorithm [4] is proposed to detect a global alignment. However, because the transcript is usually much longer than the subtitle and a segment of a TV program is generally focused on a specific topic, a sequence of words is likely repeated. This situation can lead to false alignments between a subtitle and transcripts corresponding to later subtitles. A modification of the NW [3] algorithm is proposed for selecting the alignments closest to the beginning of the transcript

(NWA). The Smith-Waterman (SW) [5] algorithm is presented in this framework to detect possible local alignments.

Let W be a dictionary consisting of a set of symbols plus the symbol **null** representing the absence of a symbol. Let T be the transcription formed by an ordered sequence of symbols of W.

$$\begin{aligned} T &= \{t_i / t_i \in W \sim null\} \\ L_T &= Card(T) \end{aligned} \tag{3}$$

Let S be the subtitle formed by an ordered set of W.

$$\begin{aligned} S &= \{s_i / s_i \in W \sim null\} \\ L_S &= Card(S) \end{aligned} \tag{4}$$

Let A be a vector representing an alignment between S and T with null representing the alignment jumps:

$$A = \{a_k(s_s, t_t) \in \{\{L_S \cup \{null\}\} \times \{L_T \cup \{null\}\}\}\} \tag{5}$$

Let $delta_{lv}(s, t)$ be the Levenshtein edit distance [28]:

$$\delta_{lv}(s, t) : W \times W \rightarrow R \tag{6}$$

Let $delta(s, t) : W \times W \rightarrow [0, 1]$ be a function that measures the dissimilarity of two W symbols.

$$\begin{aligned} \delta &= \frac{\delta_{lv}(s, t)}{max(lng(s), lng(t))} \\ \delta(s, t) &= \begin{cases} 0 & \text{if } (\delta < D_m) \\ \delta & \text{if } (D_m \leq \delta < D_M) \\ 1 & \text{otherwise} \end{cases} \end{aligned} \tag{7}$$

The parameters D_M and D_m allow ignoring small word variations. Let $delta_c(s, t) : W \rightarrow R$ be a cost function.

$$\delta_c(s, t) = \begin{cases} C_I + (C_D - C_I)\delta(s, t) \\ C_V \text{ if } s = null \\ C_H \text{ if } t = null \end{cases} \tag{8}$$

Where C_I is the cost of the two words being equal, C_D is the cost of inequality, C_H is the cost of inserting a null in the subtitle (4), and C_V is the cost of inserting a null in the transcription (3). The effect of inserting a null in the subtitle or in the transcription is symmetric. Hence, in general $C_H = C_V$. Depending on the relative values of (C_I, C_D, C_H, C_V), the alignment algorithms behave differently, prioritising the action that has the higher cost. Usually, C_I will be higher than any other. If $C_D > C_H$, it will prioritise aligning two unequal words, while $C_H > C_D$ will tend to insert gaps.

Let $C(A)$ be a cost function of the vector A:

$$C(A) = \sum_{\forall a_k(s_s,t_t)\in A} \delta_c(s_s, t_t) \tag{9}$$

Let $M[(L_S + 1) \times (L_T + 1)]$ be the alignment matrix of the algorithm, each cell in this table contains two values, the cell cost and a mark indicating an adjacent cell (*final, leftarrow, uparrow, diagonalarrow*) according to the filling criteria. Depending on the filling strategy of this table, a different algorithm is implemented. In any case, the objective of the algorithm is to compute the alignment A (5) that maximises its cost function $C(A)$ (9).

For the NW algorithm, the matrix M is filled in as follows:

$$\begin{aligned}
&m_{0,0} = 0,\ final\\
&m_{0,j} = m_{0,j-1} + C_H,\ leftarrow\ \forall j = 1..L_T\\
&m_{i,0} = m_{i-,0} + C_V,\ uparrow\ \forall i = 1..L_S\\
&m_{i,j} = \max \begin{cases} m_{i-1,j-1} + \delta_C(s,t_j),\ diagonalarrow\\ m_{i-1,j} + C_H,\ leftarrow\\ m_{i,j-1} + C_V,\ uparrow \end{cases}
\end{aligned} \tag{10}$$

In the case that the maximum can be reached from several cells, all valid indications are stored.

To build the alignment, the algorithm starts in the cell m_{L_S,L_T} and generates a pair for each displacement within the matrix according to the instructions contained in each cell. The pair is formed as follows:

$$a_k = \begin{cases} (s_i, t_j) \text{ if } m_{i,j} = diagonalarrownextm_{i-1,j-1}\\ (s_i, null) \text{ if } m_{i,j} = uparrownextm_{i-1,j}\\ (null, t_j) \text{ if } m_{i,j} = leftarrownextm_{i,j-1} \end{cases} \tag{11}$$

It is continued until reaching the cell $m_{0,0}$. In the case of the NWA algorithm, the matrix M is constructed in the same way, but when constructing the alignment, instead of starting in the cell m_{L_S,L_T} we start in the cell m_{L_S,L_M} which is the cell of the row L_S with the maximum value. If there are several cells with the same maximum value, we choose the one with the highest column index L_M.

For the SW algorithm, the matrix M is filled as follows:

$$
\begin{aligned}
&m_{0,0} = 0,\ final \\
&m_{0,j} = 0,\ final\ \forall j = 1..L_T \\
&m_{i,0} = 0,\ final\ \forall i = 1..L_S \\
&m_{i,j} = \max \begin{cases} m_{i-1,j-1} + \delta_C(s_i, t_j),\ diagonalarrow \\ m_{i-1,j} + C_H,\ leftarrow \\ m_{i,j-1} + C_V,\ uparrow \\ 0,\ final \end{cases}
\end{aligned} \tag{12}
$$

In the case that the maximum can be reached from several cells, all valid indications are stored. To build the alignment, we start with the cell with the lowest cost, and a pair is generated for each displacement within the matrix according to the instructions contained in each cell, the pair is formed as follows:

$$
a_k = \begin{cases} (s_i, t_j) \text{ if } m_{i,j} = diagonalarrownextm_{i-1,j-1} \\ (s_i, null) \text{ if } m_{i,j} = uparrownextm_{i-1,j} \\ (null, t_j) \text{ if } m_{i,j} = leftarrownextm_{i,j-1} \\ \text{final if } m_{i,j} = final \end{cases} \tag{13}
$$

It continues until it reaches a cell with the indication $final$. In this case, unlike NW, the alignment is not global, not all the words of the transcript and the subtitle are aligned, and it finds local alignments of the subtitle, totally or partially, within the transcript.

To determine whether an alignment is valid, the quality function $Q(A)$ is defined. Let A be an alignment according to (5) and $a_k(s_s, t_t) \in A$. Let a_i,a_f be the first and last pairs of A with aligned words. So, there is no jump in those alignment positions.

$$
\begin{aligned}
&lt(a_k) = len(t_t) \\
&ls(a_k) = len(s_s) \\
&score_k = (1 - \delta(s_s, t_t))ls(a_k) \\
&LS = \sum_{s \in S} len(s) \\
&LT_A = \sum_{\substack{a_k = a_f \\ a_k = a_i}} lt(a_k)
\end{aligned} \tag{14}
$$

Where $ls(a_k)$ and $lt(a_k)$ are the respective lengths of the aligned words of the a_k pair. LS is the sum of the lengths of the subtitle words. LT_A is the sum of the lengths of the transcript words between the first and last aligned word of A. Finally, $score_k$ is an indicator of the similarity between two words considering the length of the subtitle word.

The function $Q(A)[0, 1]$, index of alignment quality, is defined as:

$$Q(A) = 2 \sum_{a_k \in A} \frac{score_k}{(LS + LT_A)} \tag{15}$$

The proposed alignment algorithm (MT) is based on the simultaneous use of three alignment algorithms, NW, NWA and SW, configured with the same parameters (C_I, C_D, C_H, C_V) selecting as alignment A the one with the best quality index of the three:

$$Q(A) = \max(Q(A_{NW}), Q(A_{NWA}), Q(A_{SW})). \tag{16}$$

Given a valid A, the transcript is split into three parts using the first (w_0) and the last word (w_n) of the transcript T associated with a word of the subtitle S s_0 and s_n respectively.

- Fragment discarded: $T_d = \{w_i \in T : w_i < w_0\}$. Marked as "D in Fig. 5.
- Associated fragment: $T_a = \{w_i \in T : w_0 \leq w_i \leq w_n\}$. Marked as "Link" in Fig. 5.
- Pending Fragment: $T_p = \{w_i \in T : w_i > w_n\}$. Marked as "P" in Fig. 5.

3.2.4 *Chronization* by association

This calculation is applied when a valid association is achieved, and therefore a fragment T_a and a solution A are available, being (s_0, w_0) the first and (s_n, w_n) the last aligned pair. The procedure is as follows:

- The instant corresponding to w_0 is taken as time reference t_0.
- The position occupied by the word s_0 within the subtitle S is calculated: i_0.
- Assuming that, on average, the Spanish pronunciation time of a word is 0.385 s, the new insertion time of the subtitle is recalculated:

$$\begin{aligned} t'_i &= t_0 - 0.385 i_0. \\ \Delta_i &= t'_i - t_i \end{aligned} \tag{17}$$

- If $t'_i > t'_d$ from the previous subheading, t'_i is increased by 0.5 s.

3.2.5 *Chronization* by interpolation

It is applied when an association with subtitles after (subsequent subtitles) the one being processed has been achieved.

Let $S_1(t_{i1}, t_{e1}, t'_{i1}, t'_{e1})$, $S_2(t_{i2}, t_{e2}, t'_{i2}, t'_{e2})$ be the two subtitles timed by association, and $S_a(t_{ia}, t_{ea})$ be a subtitle pending to be timed such that:

$$t_{i1} < t_{ia} < t_{i2} \rightarrow t'_{i1} < t'_{ia} < t'_{i2}$$

Then, t'_{ia} is calculated by interpolation:

$$\begin{aligned} \Delta_{i1} &= t'_{i1} - t_{i1} \\ \Delta_{i2} &= t'_{i2} - t_{i2} \\ p &= \frac{(t_{ia} - t_{i1})}{(t_{i2} - t_{i1})} \\ \Delta_{ia} &= p * \Delta_{i2} + (1 - p) * \Delta_{i1} \\ t'_{ia} &= t_{ia} + \Delta_{ia} \end{aligned} \tag{18}$$

3.2.6 *Chronization* by inertia

Let S_i be a subtitle for which an association with the minimum quality ($Q(A) < Q_0$) has not been achieved, ($t_i > t_s - \Delta_e + \Delta_s$) is satisfied. Therefore, the broadcast delay will be exceeded, and a subsequent subtitle could not be associated, allowing the application of 3.2.5.

Let $S = \{S_k \; associated \; \wedge \; k < i\}$, so then:

$$C_i = \begin{cases} C_0 = \{S_k \in S : len(S_k) \leq 3\} \\ C_1 = \{S_k \in S : 3 < len(S_k) \leq 8\} \\ C_2 = \{S_k \in S : len(S_k) > 8\} \end{cases}$$

$$N_i = \begin{cases} N_0 = Card(C_0) \\ N_1 = Card(C_1) \\ N_2 = Card(C_2) \end{cases} \tag{19}$$

$$D_i = \begin{cases} D_0 = \frac{1}{N_0} \sum_{S_j \in C_0} \Delta_j \\ D_1 = \frac{1}{N_1} \sum_{S_j \in C_1} \Delta_j \quad \text{Con } \Delta_j = t'_j - t_j \\ D_2 = \frac{1}{N_2} \sum_{S_j \in C_2} \Delta_j \end{cases}$$

Where D_i in (19) are the mean delays calculated from the previous subtitles that have been timed (*chronized*) by association and segmented according to the number of words that make up the subtitle.

The new insertion time is calculated:

$$t'_i = \begin{cases} t_i + D_0 \text{ if } len(S_i) \leq 3 \\ t_i + D_1 \text{ if } 3 < len(S_i) \leq 8 \\ t_i + D_2 \text{ if } len(S_i) > 8 \end{cases} \tag{20}$$

3.2.7 Calculation of deletion times

To calculate the erasure time, the recommendation of the AENOR standard UNE 153010 [29] of 15 characters per second for reading has been used, which means that:

$$t'_e = t'_i + lenght(subtitle)/15 \tag{21}$$

Table 3 Scenarios. Characteristics

Scenario	TV channel	Program	Duration	Number of subtitles
1	General	Magazine	30:45	335
2	General	News	30:41	521
3	Sports	Live sports	29:5	335
4	General	News	10:00	187
5	General	Late night show	59:59	498

4 Experimental Results

For the validation of the hypothesis proposed in this paper, five test scenarios have been defined using fragments of different Spanish TV channels:

- Scenario 1: part of a magazine program. A good audio quality, a speaker with good pronunciation, occasional background music and long-time interruptions with advertisements.
- Scenario 2: part of a news program. A good audio quality, a speaker with good pronunciation, and live reports with different quality audio. Without advertisement.
- Scenario 3: part of a sports program with poor audio quality, several speakers' voices mixed with audio background noise.
- Scenario 4: part of a news program. Similar to scenario 2.
- Scenario 5: part of a late-night show. Several speakers speak fast simultaneously, and music background, noise, claps, and long-time interrupts with advertisements.

Scenarios 1, 2 and 3 are the same as those established in the work [3]. For all the scenarios, a synchronisation reference has been selected by manually aligning the subtitles. This mark is the one used to compare the results.

Table 3 shows the characteristics of the fragment associated with each scenario.

The chosen scenarios are sufficiently diverse to represent a good sample of programmes broadcast on Spanish TV channels. Considering the difficulty of the scenarios, it is expected a worse performance in those programmes with poor audio quality, with mixed audio and subtitles that are far from the literal audio and even with large segments of the broadcast without associated subtitles (commercials, songs, etc.).

Taking the above into account, the scenarios can be ordered in terms of complexity: 2, 4, 1, 3 and 5. Scenario 5 is particularly complex as it is a late night with many commercials, laughter, mixed conversations and subtitling that is very different from the literalness of what is being said.

The selected parameters are the same in all scenarios. The costs of the equations (10) and (12) are set to $(C_I, C_D, C_H, C_V) = (1, -1, -2, -2)$, the minimum confidence of the transcription $c = 0.9$, the minimum word length to be considered in the alignment: $lng\_word = 2$, the parameters of the dissimilarity function (7) are $D_M = 0.6$ and $D_m = 0.1$ and the minimum quality index for an alignment to be considered valid (15) is set to $Q_0(A) = 0.6$.

Table 4 Scenario 1. Data

Original				
N_Sub	335			
Media Dif.	9.771			
StdDev Dif.	2.522			
Chronized Synchro-Sub	Aligned	Inertied	Interpolated	Total
N_Sub	261	3	71	335
N_Sub%	76.24%	1.78%	21.98%	100.00%
Media Dif.	435	-157	618	468
StdDev Dif.	761	2.712	2.815	1.471
Chronized Sub-Sync				
N_Sub	213	122		335
N_Sub%	63,76%	36,24%	0,00%	100.00%
Media Dif.	303	714		453
StdDev Dif.	988	2.990		1.974

4.1 Scenario 1

Table 4 shows the average times and the standard deviation of the temporal differences between the time of appearance of the subtitle and the one established as a manual reference. Three sets of subtitles have been established: The original is formed by the subtitles generated during the programme's production. The label "Chronized Synchro-Sub" results from applying the algorithm developed in this work, and the label "Chronized Sub-Sync" is the result obtained with Sub-Sync.

Figure 6 presents the relative frequencies of the subtitles according to the original distribution of the subtitles and, after *chronization*, separated according to the algorithm used for *chronization*. Most of the subtitles, 261, representing 76.24%, have been timed by alignment. In the case of the same scenario in [3], this percentage was 63.76%. Most of those that could not be *chronized* by alignment was done using the interpolation algorithm.

Figure 7 shows, for scenarios 1, 2 and 3, a comparison of the distribution of subtitles according to the time difference with the reference for the results obtained in this work (*Chronized*) and those obtained in Sub-Sync (*ChronizedArt*) using only the asymmetric NW alignment.

4.2 Scenario 2

Scenario 2, characterised according to Table 3, is a news programme with good audio and well-defined texts, so a high percentage of *chronization* by the alignment is obtained.

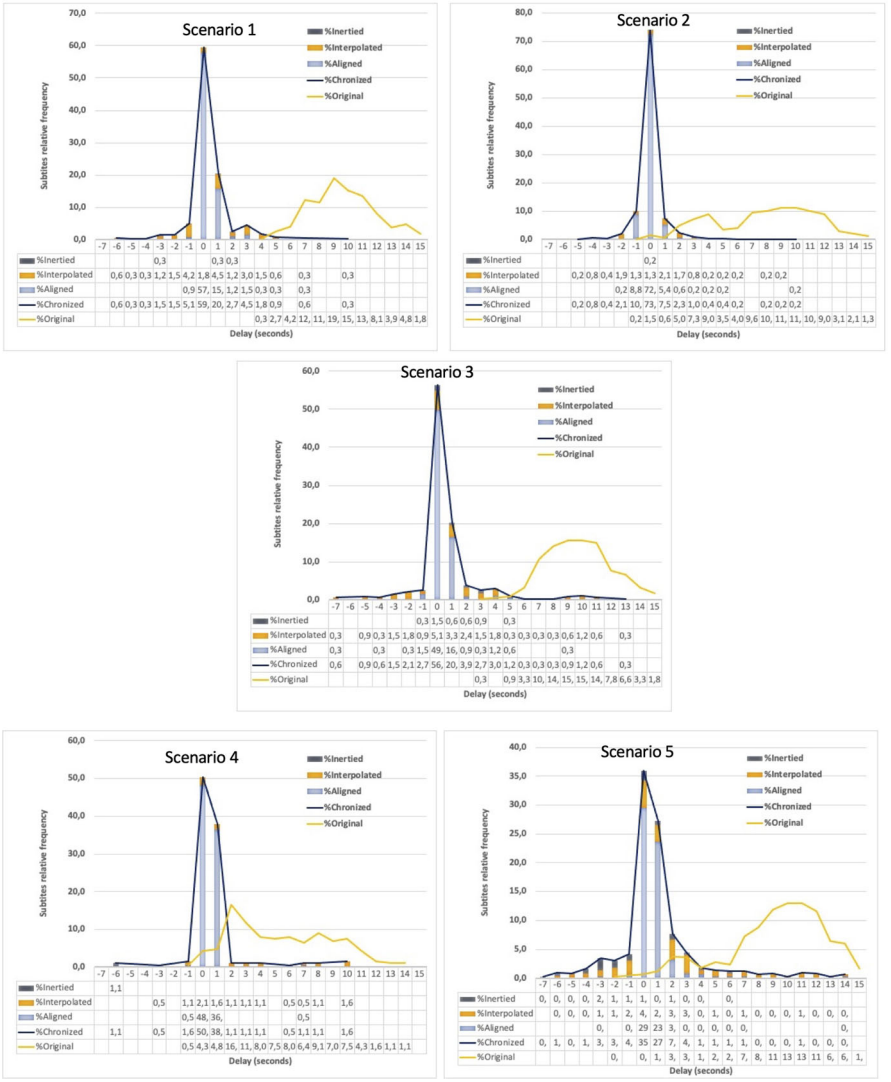

Fig. 6 Comparison of Original vs *chronized* Synchro-Sub" algorithms (all scenarios)

The results are summarised in Table 5, and a level of alignment of 88.17% is observed. Figure 6 shows the distribution of subtitles and the algorithm used for the synchronisation. Figure 7 compares the results with those obtained with Sub-Sync. In the case of Sub-Sync, only two algorithms were available, and this framework did not use the interpolation algorithm.

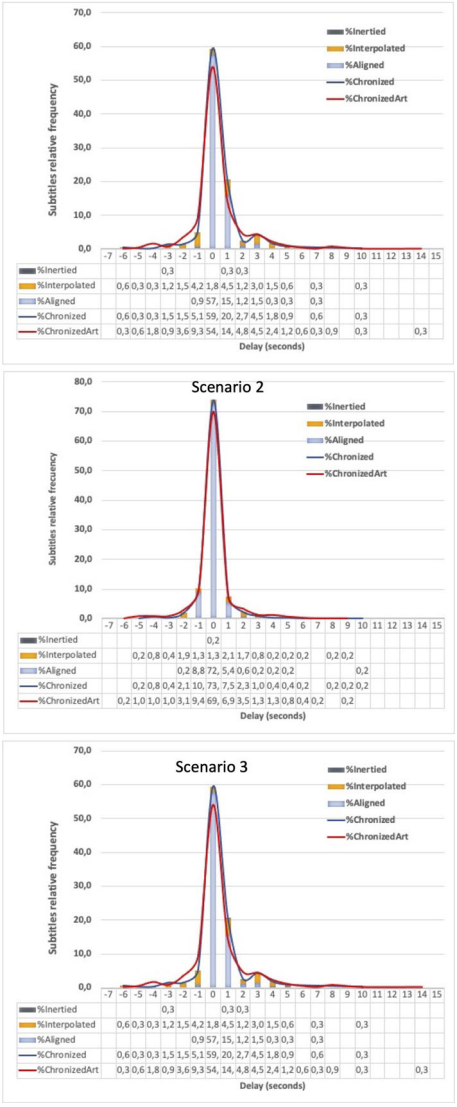

Fig. 7 Comparison of new Synchro-Sub algorithms vs previous Sub-Sync (scenarios 1, 2 and 3)

4.3 Scenario 3

Scenario 3, characterised according to Table 3, is a sports programme with bad audio, so it is expected that the *chronized* by alignment performance will decrease compared to scenario 2. A summary of the timing (*chronization*) performed by the algorithms is shown in Table 6. Figure 6 shows the distribution of subtitles and the algorithm with which they have been timed. Figure 7 compares the results with those obtained with Sub-Sync.

Table 5 Scenario 2. Data

Original				
N_Sub	521			
Media Dif.	7.967			
StdDev Dif.	3.554			
Chronized Synchro-Sub	Aligned	Inertied	Interpolated	Total
N_Sub	459	1	61	521
N_Sub%	88.17%	0.13%	11.70%	100.00%
Media Dif.	25	−355	229	48
StdDev Dif.	700		2.975	1.207
Chronized Sub-Sync				
N_Sub	407	114		521
N_Sub%	77.14%	22.86%	0.00%	100.00%
Media Dif.	−7	99		16
StdDev Dif.	593	2.746		1.384

Table 6 Scenario 3. Data

Original				
N_Sub	335			
Media Dif.	10.203			
StdDev Dif.	2.898			
Chronized Synchro-Sub	Aligned	Inertied	Interpolated	Total
N_Sub	241	14	80	335
N_Sub%	65.96%	5.92%	28.12%	100.00%
Media Dif.	444	1.321	1.862	819
StdDev Dif.	1.072	1.672	4.506	2.472
Chronized Sub-Sync				
N_Sub	179	156		335
N_Sub%	44.28%	55.72%	0.00%	100.00%
Media Dif.	173	−990		−368
StdDev Dif.	867	3.461		2.509

4.4 Scenario 4

Scenario 4, characterised according to Table 3, is a news programme and, as shown in Table 7, has a very high level of alignment (91.7%).

Figure 6 shows the distribution of the *chronized* subtitles and its comparison with the original allocation (comparison of the distribution of the *chronized* and the original

Table 7 Scenario 4. Data

Original				
N_Sub	187			
Media Dif.	5.513			
StdDev Dif.	3.539			
Chronized Synchro-Sub	Aligned	Inertied	Interpolated	Total
N_Sub	162	2	23	187
N_Sub%	91.70%	0.85%	7.45%	100.00%
Media Dif.	537	−5.680	3.205	799
StdDev Dif.	1.692	14	3.965	2.359

subtitle). Table 7 shows the details of the distribution of the *chronization* according to the algorithm used.

4.5 Scenario 5

Scenario 5 corresponds to a late-night show, so it is a scenario with a lot of noise, poor audio quality, many people talking simultaneously, and voice mixing. In addition, there are segments of very long commercials (minutes), for which there are no subtitles. In this case, the literalness of the subtitles is much smaller than in the previous scenarios. The parameters of this scenario are detailed in Table 3.

Table 8 Scenario 5. Data

Original				
N_Sub	498			
Media Dif.	10.462			
StdDev Dif.	20.406			
Chronized Synchro-Sub	Aligned	Inertied	Interpolated	Total
N_Sub	298	55	145	498
N_Sub%	58.19%	14.90%	26.90%	100.00%
Media Dif.	742	−1.363	2.164	924
StdDev Dif.	1.652	4.559	4.364	3.233

Table 8 summarises the results for this scenario. As expected, the number of subtitles *chronized* per alignment decreases. However, the average deviation from the reference remains stable at around 924 ms. Finally, Fig. 6 shows a comparison of the distribution of the *chronized* and original subtitles.

4.6 Discussion

As explained in the manuscript, five test scenarios have been defined using fragments of different Spanish TV channels to validate the hypothesis proposed in this paper. Moreover, the three first scenarios are the same that were tested with the Sub-Sync framework in previous research. Therefore, two new scenarios had not been tested before (scenarios 4 and 5).

Moreover, in the Literature Review section, different research was analysed. In this section, a comparative table is included to highlight the main characteristics of those proposals. However, the only proposal with a similar approach and scope that can be compared at the outcome level is the one presented in [3]. It is also important to note that the proposal's performance depends on the scenario where it is tested. Therefore, comparing results on similar video fragments (from TV programs) is necessary.

Table 9 summarises the results obtained for Synchro-Sub and Sub-Sync frameworks in scenarios 1,2, and 3 in the case of aligned subtitles. What can be interpreted from the results obtained is that both proposals perform equally well 53% of the time, and in the rest, there is a better performance of Synchro-Sub (97/65 in scenario 1, 112/86 in scenario 2, 125/76 in scenario 3 and 334/226 in total).

As scenarios 4 and 5 are new in this research, there are no previous results for being compared between both proposals.

Scenarios 1,2 and 3 present a high level of alignment with Synchro-Sub framework. In all the scenarios, it improves the results of Sub-Sync framework. Moreover, as expected, due to the characteristics of those scenarios, scenarios 3 and 5 obtain a lower number of subtitles *chronized* per alignment.

Table 9 Scenarios. Summary results' comparison between % of aligned subtitles in Synchro-Sub and Sub-Sync

Program	Scenario	Duration	Number of subtitles	*Better Synchro-Sub*	*Better Sub-Sync*	Equal	*% Better Synchro-Sub*	*% Better Sub-Sync*	% Equal
Magazine	1	30:45	335	97	65	173	29%	19%	52%
News	2	30:41	521	112	86	323	21%	17%	62%
Direct	3	29:5	335	125	76	134	37%	23%	40%
Total			1191	334	227	630	28%	19%	53%

5 Conclusions

One of the main problems in broadcasting subtitles on live television comes from the generation, transcription and formation of the subtitles and the insertion and presentation of these subtitles on the screen.

In [30], the author has analysed the average delay time in subtitles with a database containing more than 1,504,426,787 lines of subtitles from live broadcasts. As a result of this analysis, an average delay of 12.2 s in live programmes has been detected. Moreover, this delay depends on the solution adopted for the generation of the subtitle (stenotype, respeaking, ASR, etc.) and the genre and structure of the TV programme (news, sports, entertainment, etc.) [19].

This situation leads to a significant distortion in the understanding of the message, especially for people with residual hearing.

To address this problem, in this paper, the authors propose a new synchronisation framework, Synchro-Sub, based on the insights and knowledge extracted from previous research. Synchro-Sub defines a new approach for synchronising the subtitles with the audio-visual contents for live-captioned TV programs by automatically adjusting to both time-ubication. Different improvements have been included in the three phases of the framework: a new alignment process between the word sequences of the subtitle and the transcription, two new algorithms for the alignment between subtitle and transcription, a new filtering process for subtitles and transcriptions, etc. Moreover, the framework includes a real-time adaptive process for the subtitles' alignment. The NWA algorithm is proposed for selecting those alignments closest to the beginning of the transcript, and the SW algorithm is included in this framework to detect possible local alignments.

Those improvements have been tested in five scenarios defined using fragments of different Spanish TV channels. The chosen scenarios are sufficiently diverse to represent a good sample of programmes broadcast on Spanish TV channels. Considering the difficulty of the scenarios, a worse performance was expected in those programmes with poor audio quality, mixed audio and subtitles that are far from the literal audio and even with large segments of the broadcast without associated subtitles (commercials, songs, etc.). Therefore, one of the scenarios (scenario 5) was particularly complex as it is a late night with many commercials, laughter, mixed conversations and subtitling that is very different from the literalness of what is being said.

The best results are obtained in scenario 4, a news programme with a high level of alignment (91.7%). In scenario 5, as expected, due to the poor audio quality, the number of subtitles *chronized* per alignment decreases to 58.19%. However, the average deviation from the reference remains stable at around 924 ms. Moreover, in scenarios 1, 2 and 3, the new alignment algorithms of the Synchro-Sub framework obtain better results compared with the Sub-Sync framework. Finally, in Sub-Sync, only two algorithms were available, and this framework did not use the interpolation algorithm.

Future research will focus on increasing the signal quality used in the framework for improving the alignment results (as in scenario 5). Moreover, the authors will analyse the framework's adaptation to languages other than Spanish, focusing on moving the ASR component to different languages and customising the filtering process considering the features of each language.

Acknowledgements This work was partially supported by R&D&i ACCESS2MEET (PID2020-116527RB-I0) project supported by MCIN AEI/10.13039/501100011033/. Additionally, this work has been supported by the Madrid Government (Comunidad de Madrid-Spain) under the Multiannual Agreement with UC3M in the line of Excellence of University Professors (EPUC3M17), and in the context of the V PRICIT (Regional Programme of Research and Technological Innovation).

Funding Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature.

References

1. Martín A, González-Carrasco I, Rodriguez-Fernandez V, Souto-Rico M, Camacho D, Ruiz-Mezcua B (2021) Deep-sync: a novel deep learning-based tool for semantic-aware subtitling synchronisation. Neural Comput Appl 1–15
2. Simpson MN, Barrett J, Bell PJ, Renals S (2016) Just-in-time prepared captioning for live transmissions. In: IBC 2016 conference, pp 27 (9 .)–27 (9 .). Institution of Engineering and Technology
3. Gonzalez-Carrasco I, Puente L, Ruiz-Mezcua B, Lopez-Cuadrado JL (2019) Sub-sync: automatic synchronization of subtitles in the broadcasting of true live programs in Spanish. IEEE Access 7:60968–60983
4. Needleman Saul B, Wunsch Christian D (1970) A general method applicable to the search for similarities in the amino acid sequence of two proteins. J Mol Biol 48(3):443–453
5. Smith TF, Waterman MS (1981) Identification of common molecular subsequences. J Mol Biol 147:195–197
6. World Health Organization (2011) World disability report. Technical report
7. Mocanu B, Tapu R, Zaharia T (2019) Enhancing the accessibility of hearing impaired to video content through fully automatic dynamic captioning. In: 2019 7th E-health and bioengineering conference, EHB 2019. Institute of Electrical and Electronics Engineers Inc
8. Tamayo A, Chaume F (2017) Subtitling for d/deaf and hard-of-hearing children: current practices and new possibilities to enhance language development. Brain Sci 7(7):75
9. Hong R, Wang M, Xu M, Yan S, Chua TS (2010) Dynamic captioning: video accessibility enhancement for hearing impairment. In: MM'10 - proceedings of the ACM multimedia 2010 international conference, ACM Press, New York, USA, pp 421–430
10. Safadi B, Sahuguet M, Huet B (2014) When textual and visual information join forces for multimedia retrieval. In: ICMR 2014 - proceedings of the ACM international conference on multimedia retrieval 2014, Association for Computing Machinery, pp 265–272
11. Jararweh Y, Al-Ayyoub M, Fakirah M, Alawneh L, Gupta BB (2019) Improving the performance of the needleman-wunsch algorithm using parallelization and vectorization techniques. Multimed Tools Appl 78(4):3961–3977
12. Needleman SB, Wunsch Christian D (1970) A general method applicable to the search for similarities in the amino acid sequence of two proteins. J Mol Biol 48(3):443–453
13. Day RF (2010) Examining the validity of the Needleman-Wunsch algorithm in identifying decision strategy with eye-movement data. Decis Support Syst 49(4):396–403
14. Zhang F, Qiao XZ, Liu ZY (2002) A parallel Smith-Waterman algorithm based on divide and conquer. In: Proceedings - 5th international conference on algorithms and architectures for parallel processing, ICA3PP 2002, Institute of Electrical and Electronics Engineers Inc., pp 162–169

15. Ligowski L, Rudnicki W (2009) An efficient implementation of Smith Waterman algorithm on GPU using CUDA, for massively parallel scanning of sequence databases. In: IPDPS 2009 - proceedings of the 2009 IEEE international parallel and distributed processing symposium
16. Su Z, Ahn BR, Eom KY, Kang MK, Kim JP, Kim MK (2008) Plagiarism detection using the Levenshtein distance and Smith-Waterman algorithm. In: 3rd International conference on innovative computing information and control, ICICIC'08
17. Lemantara J, Dewiyani Sunarto MJ, Hariadi B, Sagirani T, Amelia T (2018) Prototype of online examination on MoLearn applications using text similarity to detect plagiarism. In: Proceedings - 2018 5th international conference on information technology, Computer and Electrical Engineering, ICITACEE 2018, Institute of Electrical and Electronics Engineers Inc., pp 131–136
18. Popescu M (2010) An ontological fuzzy Smith-Waterman with applications to patient retrieval in Electronic Medical Records. In: (2010) IEEE World Congress on Computational Intelligence. WCCI 2010
19. Souto-Rico M, González-Carrasco I, López-Cuadrado J-L, Ruiz-Mezcua B (2020) A new system for automatic analysis and quality adjustment in audiovisual subtitled-based contents by means of genetic algorithms. Expert Systems
20. Garcia J E, Ortega A, Lleida E, Lozano T, Bernues E, Sanchez D (2009) Audio and text synchronization for TV news subtitling based on automatic speech recognition. In: 2009 IEEE International symposium on broadband multimedia systems and broadcasting, BMSB 2009, IEEE, pp 1–6
21. Ando A, Imai T, Kobayashi A, Homma S, Goto J, Seiyama N, Mishima T, Kobayakawa T, Sato S, Onoe K, Segi H, Imai A, Matsui A, Nakamura A, Tanaka H, Takagi T, Miyasaka E, Isono H (2003) Simultaneous subtitling system for broadcast news programs with a speech recognizer. IEICE Trans Inf Syst E86–D(1):15–25
22. Lertwongkhanakool N, Punyabukkana P, Suchato A (2013) Real-time synchronization of live speech with its transcription. In: 2013 10th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, ECTI-CON 2013
23. de Castro M, Rodríguez L P, Ruiz-Mezcua B (2015) Synchronized subtitles in live television programmes. In: Audiovisual translation in a global context, Palgrave Macmillan UK, pp 51–71
24. Pražák A, Loose Z, Trmal J, Psutka J, Psutka J (2012) Captioning of live TV programs through speech recognition and re-speaking. In: Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics), vol 7499 LNAI. Springer, Berlin, Heidelberg, pp 513–519
25. Gao J, Zhao Q, Li T, Yan Y (2009) Simultaneous synchronization of text and speech for broadcast news subtitling. In: Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics), vol 5553 LNCS. pp 576–585
26. Ando A, Imai T, Kobayashi A, Isono H, Nakabayashi K (2000) Real-time transcription system for simultaneous subtitling of Japanese broadcast news programs. IEEE Trans Broadcast 46(3):189–196
27. Suzuki T (2020) Simultaneous speech subtitling systems for multiple speakers. In: Communications in Computer and Information Science, vol 1226 CCIS. Springer, pp 114–120
28. Levenshtein Vladimir (1966) Binary codes capable of correcting deletions, insertions, and reversals. Sov Phys Dokl 10(8):707–710
29. AENOR (2012) UNE 153010:2012. Subtitulado para personas sordas y personas con discapacidad auditiva
30. Souto-Rico M (2021) Estudio de la velocidad de los subtítulos para sordos en España y sus consecuencias normativas. PhD thesis