I. González-Carrasco, L. Puente, B. Ruiz-Mezcua and J. L. López-Cuadrado, "Sub-Sync: Automatic Synchronization of Subtitles in the Broadcasting of True Live programs in Spanish," in IEEE Access, vol. 7, pp. 60968-60983, 2019

DOI: 10.1109/ACCESS.2019.2915581

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.Doi Number

Sub-Sync: Automatic synchronization of subtitles in the broadcasting of true live programs in Spanish

I. González-Carrasco[1], L. Puente[2], B. Ruiz-Mezcua[3] and J. L. López-Cuadrado[4]

[1]I. González-Carrasco, Universidad Carlos III de Madrid, Computer Science Department, Leganés, Madrid, igcarras@inf.uc3m.es
[2]L. Puente, Spanish Center for Captioning and Audiodescription, Leganés, Madrid, lpuente@cesya.es
[3]B. Ruiz-Mezcua, Universidad Carlos III de Madrid, Computer Science Department, Leganés, Madrid, bruiz@inf.uc3m.es
[4]J. L. López-Cuadrado, Universidad Carlos III de Madrid, Computer Science Department, Leganés, Madrid, jllopez@inf.uc3m.es

Corresponding author: I. González-Carrasco (e-mail: igcarras@inf.uc3m.es).

ABSTRACT Individuals with sensory impairment (hearing or visual) encounter serious communication barriers within society and the world around them. These barriers hinder the communication process and make access to information an obstacle they must overcome on a daily basis. In this context, one of most common complaints made by the Television (TV) users with sensory impairment is the lack of synchronism between audio and subtitles in some types of programs. In addition, synchronization remains one of the most significant factors in audience perception of quality in live-originated TV subtitles for the deaf and hard of hearing. This paper introduces the Sub-Sync framework intended for us in automatic synchronization of audio-visual contents and subtitles, taking advantage of current well-known techniques used in symbol sequences alignment. In this particular case, these symbol sequences are the subtitles produced by the broadcaster subtitling system, and the word flow generated by an Automatic Speech Recognizing procedure. The goal of Sub-Sync is to address the lack of synchronism that occurs in the subtitles when produced during the broadcast of live TV programs or other programs that have some improvised parts. Furthermore, it also aims to resolve the problematic interphase of synchronized and unsynchronized parts of mixed type programs. In addition, the framework is able to synchronize subtitles even when they do not correspond literally to the original audio and/or the audio cannot be completely transcribed by an automatic process. Sub-Sync has been successfully tested in different live broadcasts, including mixed programs, in which synchronized parts (recorded, scripted) are interspersed with desynchronized (improvised) ones.

INDEX TERMS Accessibility, TV broadcasting, Algorithm Design and Analysis, Automatic Speech Recognition

I. INTRODUCTION

Individuals with sensory impairment (hearing or visual) encounter serious communication barriers within society and the world around them [1]. These barriers hinder the communication process and make access to information an obstacle that they must overcome on a daily basis. Furthermore, this situation generates social exclusion and is one of the reasons that presents the sensory-impaired from achieving equal rights. These rights are of such importance that the organization of the United Nations recognizes this situation in its International Convention on the Rights of Persons with Disabilities adopted on 13 December 2006 [2]. Noteworthy is article 30, which states emphatically that States should take all appropriate measures to ensure that persons with disabilities have access to television programs, films, theatre and other cultural activities, in accessible formats.

Taking into account the world disability report [3], there are over one thousand million people with at least one type of disability. Accordingly, there is increasing demand by individuals and associations of disabled persons to make cultural activities accessible, and in particular, for television programme contents to be subtitled, and furthermore, for this subtitling to be of high-quality [4]. This situation has led governments and regulatory entities to establish mandatory requirements for broadcasters to ensure that television has

subtitling and sign language services for the deaf ,and audio description services for the blind or visually- impaired [5].

At the Spanish level, the "Informe de seguimiento del subtitulado y la audiodescripción en la TV", translated to English as "Follow-up report of the subtitles and audio description in TV" [6] states that one of the main complaints made by the TV´s users is the lack of synchronism between audio and subtitles in some types of programmes Moreover, delay remains one of the most significant factors in the audience's perception of lack of quality in live-originated TV captions or subtitles for the deaf and hard of hearing [7].

In this context, this research focuses on the process of re-situating the appearance times (insert) and deletion (erase) of the subtitles, which ensure synchronization between them and the audio-visual elements. We must take into account that the quality and even the actual possibility of the production of the subtitles by the television channel will depend on the type of program. In this regard, this research is aimed at synchronizing the subtitles with the audio-visual contents for live-captioned TV programs by making an automatic adjustment to the time-ubication of both.

The framework Sub-Sync has been designed to produce an automatic synchronization of audio-visual contents and subtitles relying on techniques used in the alignment of two symbol sequences, which are; the subtitles produced by the broadcaster's subtitling system; and the word flow generated by an Automatic Speech Recognition (ASR), regardless of the method used for this last one. To achieve this synchronization, this framework requires a three-phase process as described in Figure 1 below. The first phase called transcription, uses an ASR engine to produce a continuous word flow which corresponds with the transcription of the audio-visual's audio [8]–[10]. The results of this phase are stored in a sequential mode in the framework's component called transcription storage. Each word saved in this memory has an associated timestamp that indicates the precise time in which the sound it represents was produced.

The target of the second phase is identifying the correct time in which each one of the subtitles will have been presented. The authors have called this phase chronization. This chronization phase is comprised of two stages. In the first one, words contained in the transcription storage and the subtitles are compared in order to establish similarities between them, trying to create pairs of similar words. In the second stage, these similarities are processed to establish the best alignment between the words of the subtitle and transcription storage. For this alignment the algorithm proposed by Needleman-Wunsch [11]–[14] is used, which is useful because of its effectiveness in the alignment of symbol sequences.

The best possible alignment will be one that minimizes a given cost function (C). Notwithstanding the foregoing, it shall be taken into consideration that, due to reasons described later in this paper there are some subtitles for which a reasonable alignment may not be found. In order to determine this situation, a maximum cost limit has been established in Sub-Sync. If this limit is exceeded, it shall be assumed that no reasonable alignment between the subtitle and the transcription and thus, an alternative strategy, will be applied in order to establish the subtitle presentation and deletion times. In such a case, this paper suggests an inertia algorithm designed by the authors in order to estimate those times.

Finally, the third phase of the Sub-Sync process is to construct the synchrony between the subtitles and the audio-visual contents, which is called synchronization. In the previous phase, the original delay has been increased in the time needed to estimate the correct times, thus, it is necessary to store and delay the broadcasting of the audio-visual contents until the subtitles are available. This delay enables insertion of the subtitles in the frame indicated by the chronization process.

Figure 1. Different phases in the Sub-Sync framework

The Sub-Sync framework has been successfully tested in various live TDT broadcasts, including as well mixed programs in which synchronized parts (pre-recorded or scripted, see subsections A and B of section I) and unsynchronized parts (improvised, see subsection C of section I) are interspersed. It is in the transitions of the mixed programs where serious issues of synchronization arise between subtitles and audio-visual contents.

Therefore, the goal of the Sub-Sync framework is to correct the lack of synchronism that arises in the subtitles when these are produced during the broadcast of live TV programs or other programs that have some improvised parts. In addition, it also aims to resolve the problematic interphase of synchronized and unsynchronized parts of mixed type programs.

The main users for this proposal are individuals with hearing impairment, pre-lingual (prior to speech acquisition), post-lingual (posterior) or individuals with poor quality hearing. However, there are other users who can directly benefit from this system: children, the elderly, persons with intellectual disabilities, people learning languages, since the possibility of improving reading and writing skills is an added value to subtitling services. In addition, the general public can also benefit from subtitling in certain noisy environments (public transport, commercial areas, etc.), and their knowledge sensitizes and prepares them for possible age-associated hearing loss. In addition, using this technology for

distribution in television or streaming on the Internet can be a key factor for the education sector. The framework can be used to improve learning for persons with disabilities or for example for individuals with reading difficulties. I can also be used as a basic tool for learning the local language for foreigners.

Finally, a tool with these characteristics will greatly help to improve the accessibility of audiovisual content, not only for its broadcast on television, but also for its transmission through digital media platforms (via streaming, HbbTV or IPTV).

The paper consists of seven sections and is structured as follows. Section II presents the problem domain and the issues associated with the use of accessibility services based on subtitles. Section III reviews the relevant literature. Section IV exposes the current situation for the synchronization of subtitles in the broadcast of audiovisual content. Section V discusses the main features of Sub-Sync, including the architecture of the framework, algorithms and phases. Section VI presents the experimentation and results obtained, and finally Section VII discusses the conclusions and future lines of research.

II. BACKGROUND

Television subtitle services can be classified according to the following categories:

- When are they generated (before or during the broadcasting)?
- How are they generated (typing, stenography, re-speaking)?
- When are they introduced (in a predefined way, live synchronized, as soon as possible)?
- How are they broadcasted (incrusted, linked, synchronized or unsynchronized)?
- What type of program are they used in (pre-recorded, scripted, live improvised, mixed)?

Because of its importance in the context of this research, a description is given below of the characteristics of the different types of TV program along with associated issues regarding the use of accessibility services, such as subtitles.

A. PRE-RECORDED PROGRAMS

There are many programs broadcasted on television that are pre-recorded. Some of them already have subtitles since they were generated during the editing period. At other times, the subtitles generation and their insertion in the video are done afterwards, regardless of whether insertion is done in a visible way for all the viewers (open caption) or only visible for those who select the service (closed caption).

Since there is time enough between the production and the broadcast of the audio-visual content, it is possible to carefully produce the subtitles and review them “before the broadcast”. In a manual or semi-automatic way locutions may be transcribed, split them in subtitles and synchronize them with the audio [4].

B. LIVE-SCRIPTED PROGRAMS

There are other scenarios in which, although the audio-visual production is simultaneous with its broadcast, the interventions of the speakers are subject to a pre-established script that is rigorously followed. This circumstance allows "editing" the subtitles in advance "before the broadcast" from the script. Thus, its quality can be very high as long as the speakers comply with said script.

The subtitles are broadcasted in a "synchronized" form, through a procedure that is carried out "live" at the same time of the broadcast. Usually, a person manually inserts the subtitles following the pace of the speakers. For programs with simple environments such as news programs, the literature offers a wide range of techniques for the automation of synchronization [4], [15]–[17].

C. LIVE-IMPROVISED PROGRAMS

This section includes all the programs for which it is not possible to have a rigorous script from which to produce the subtitles. Included in this group are two types of programs; programs in which guests participate, such as debate or interview programs; and also, those in which the speakers have some freedom to express themselves within the topic to be discussed. Thus, they are improvised broadcasts as they are not bound to a pre-established script.

In these circumstances, it is not possible to know in advance what the necessary subtitles will be, so they will have to be generated in real time "during the broadcast" [4]. This necessarily implies a loss of quality, as the systems that produce them: human, machine or mixed, are not error-free and the possibility of correcting them is very limited in time. There are three usual techniques for the generation of subtitles in this scenario: by typing, stenography and re-speaking. The most common ones are the stenography and re-speaking. Typing is an alternative to stenography for those languages that have not developed their own stenographic system.

Stenography is a technique similar to typing, but which uses a special reduced keyboard that allows considerable increase in writing speed. With this technique and tool or, at least, by means of common typing, it is possible to generate the transcription of the locutions consecutively. Recent adaptations enable insertion of colour subtitles for speaker identification.

The "re-speaking" technique is one of the most commonly used in TV. It uses speech recognition technology to obtain a transcription of the locution. It is not feasible under the current state of the art of this technology, to use it directly on the audio of the event [10]. There are several reasons for this. One is that these recognition systems improve their performance significantly when they have been trained for a specific person; this is something that is not controllable in the scenarios to which we refer [18]. However, the most important reason may be that the performance of these systems decrease exponentially according to the level of the background noise (voice, music, other sounds), and is exacerbated when this noise is similar to speech, such as the other speakers' voices on the set [19].

Thus, the alternative and widely-extended solution is that a professional speaker, in a room fitted for this purpose, repeats (re-speaks) what is said on the stage. This repetition is sent to an ASR that has been trained in recognizing this professional speaker. It must be borne in mind that, unlike the stenography typist, whose work focusses on producing literal content, re-speakers, because of time reasons, simplify and summarize the original phrase.

Obviously, since both the stenotypist and the re-speaker have to follow the announcer and cannot anticipate him, the subtitles are issued "as soon as possible" in an "unsynchronized" way (usually delayed) with respect to 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.

This situation is dysfunctional for multiple reasons. One reason is the confusion created for the hearing-impaired by the lack of correlation between what they are reading and the speaker´s lip movements. Another is the confusion produced when subtitles that correspond to an audio of one speaker appear when that speaker is no longer talking and / or the image shows another speaker. Finally, for the hard of hearing, there is the discrepancy between the subtitle and the audio-visual information.

D. MIXED PROGRAMS

The previous sections have referred to three types of programs, but instead they should have referred to types of program parts, since it is usual for these programs to be composed of two or three of these types. For example, news programs, which at first glance would be considered live scripted programs, but once analyzed, it can be observed that they also incorporate interviews on the set, outset connections and/or pre-recorded reports.

As is natural, in order to optimize the subtitling quality, it is necessary to use all the techniques described above. This is an additional problem for the subtitle generation and synchrony that arises in the transitions between different types of program.

Mixed programs are interspersed with synchronized (pre-recorded, scripted) and unsynchronized parts (improvised). In the synchro-unsynchronized transition, there is a pause in the subtitles broadcasting, while waiting for them to be available, which causes confusion for the viewer. Conversely, in the unsynchro-synchronized transition, there is a "time loss" since while the first subtitles have not yet been fully broadcasted, the second ones start. In order to recover this lost time, the production control accelerates their release until the natural rhythm of the program is reached. This leads to very short subtitle persistence[1] times, which may be too short, even reaching persistence times of zero seconds.

III. LITERATURE REVIEW

The automatic synchronization of subtitles and audio has been a subject of study for previous researchers, who have produced a good deal of studies in this area.

Commonly, these synchronization proposals are considered as a component-tool for the process of automatic subtitle generation. In [20], the authors propose an integrated framework of automatic bilingual subtitle (Chinese and English) generation for lecture videos, especially for MOOCs. The framework consists of Automatic Speech Recognition (ASR), Sentence Boundary Detection (SBD), and Machine Translation (MT). Moreover, the BSD component is based on word vectors and Deep Neural Networks (DNN).

In [21], the authors present a multi-hierarchy semantic information descriptive model based on video-segmentation and extraction technology. The aim of their research is to do the mapping of video semantic information from the low-level feature to high-level while carrying out subtitle translation studies based on a technical level.

Furthermore, there are also some proposals that suggest the use of web services for synchronization in batch processes. For example, [22] presents a web-based platform that enables the customization and synchronization of subtitles on both single- and multi-screen scenarios. The platform enables the dynamic customization of the subtitles' format (font family, size, colour, etc.) and position according to the users' preferences and / or needs.

The synchronization of audio and text is also investigated and developed for applications other than subtitling. For example, in [17], an algorithm to synchronize live speech with its corresponding transcription in real time at the syllabic unit is proposed. The goal of this research is to apply

[1] persistence = subtitle erase time – subtitle presentation time

the algorithm for generating audio books in unstructured language like Thai from live speech.

In [16] a system developed for the Aragonese public television in Spain is presented. This system automatically synchronizes subtitles with audio in news programs. Specifically, the Speech-Text Alignment module (STA) connects with the Text Retrieval (TR), which interacts with the News Redaction Computer (NRC) to obtain the texts that are to be presented on the screen. The STA, which is based on an ASR engine, receives these texts and performs a temporary alignment with the incoming audio signal. This system has been tested only on "live scripted" types of programs. The recognition engine is based on Hidden Markov Models (HHM) formed by Gaussian Mixture Models (GMM), with feature vectors formed by Mel Cepstrum Coefficients, Normalized Energy, Speeds and Accelerations..

In [15] presents a system for subtitling news programs on Japanese TV. In this research, in order to simultaneously broadcast subtitled Japanese news programs, the authors implemented a simultaneous subtitling system relying on speech recognition. The system consisted of a real-time speaking recognition system that handled the transcription of broadcast news and a recognition-error correction system that corrected mistakes in the recognition result within a short delay time. Another related proposal is presented in [23]. The system is based on a hybrid automatic speech recognition system that switches input speech between the original program sound and the rephrased speech by a "re-speaker".

There is a general demand by society at large and governments alike that is propelling TV broadcasters to increase the amount and diversity of subtitled programs. The ratio of subtitled TV programs in China is still low. For live programs, such as broadcast news, the ratio is even lower. [4] presents a system for subtitling news programs on Chinese television. The system is formed by two main modules. The first one, the "text processor" module, prepares the transcripts of the news before the announcers read them at the microphone. This module has two functions; normalizing the text; and dividing it into appropriate sentences for its presentation on the TV screen. The "text-speech synchronizer" module explores the speech in order to determine the starting and ending times of the sentences so that the subtitles are presented when their speech is made. This is done by using an extension of an alignment algorithm based on Viterbi [24].

Taking into account the broadcasting of the signal, in Hybrid Broadcast Broadband TV (HbbTV), different proposals have sought to deal with the synchronization of the audiovisual content and the subtitles. In [25] a new application for checking the subtitle and video content synchronization is proposed. Moreover, media synchronization (e.g. broadcast video with subtitles or alternative audio received via the Internet) is receiving renewed attention with ecosystems of connected devices enabling novel media consumption paradigms [26].

Another goal is to make content available to all in order to overcome the existing access barriers to content for users with specific needs, or else to adapt to different platforms, hence making content fully usable and accessible. [27] presents an interactive system for automatically generating video summaries and performing subtitles synchronization for persons with hearing loss using Natural Language Processing (NLP). In [28] a video subtitling system that enables the customization, adaptation and synchronization of subtitles across different devices and multiple screens is used, allowing access to these contents to more people anywhere.

In [29], also for live scripted programs, an ASR is used for the synchronization of the predefined subtitles. The ASR generates transcription of the program's audio and includes the time stamp of each word. Later, by means of an alignment algorithm based on dynamic programming techniques, both sequences (ASR generated and predefined subtitles) are aligned and each word's time is defined.

In [7], the authors present an automated solution based on ASR, context-tuned models, and the practical application of Machine Learning across large corpora of data – namely many hours of accurately captioned English broadcast news programs. The main goal of the research is to create recognition and punctuation models to transform raw automated transcription into broadcast captions for introducing the technology into a live production environment.

Finally, regarding ASR techniques, usually top speech recognition systems rely on sophisticated $f1$s composed of multiple algorithms and hand-engineered processing stages. Traditional speech systems use many heavily engineered processing stages, including specialized input features, acoustic models, and Hidden Markov Models (HMMs). To improve these pipelines, domain experts must invest a great deal of effort into fine-tuning their features and models.

Today, the introduction of deep learning algorithms in ASR components has improved speech system performance and robustness [8], [9], [30]–[32]. While this improvement has been significant, deep learning still only plays a limited role in traditional speech pipelines. As a result, to improve performance in a task such as recognizing speech in a noisy environment, one must laboriously engineer the rest of the system for robustness. Therefore, the further challenge of dealing with noisy background conditions can be met by using state-of-the-art deep learning techniques.

As indicated above, the link between the subtitle and the locution is a case of a classical issue: the alignment of symbol sequences. In [11], Needleman and Wunch propose a technique to find the best alignment between two sequences of symbols. Although it was originally developed for the localization of amino acid sequences in proteins, the irrelevance of the type of symbol to be located has allowed it to be used in multiple fields, among others, the location of similarities between sequences of words. The Needleman and Wunch algorithm is used today in different domains as a tool for a punctuation prediction model for conversational speech [33], as a support technique for large-scale computerized text analysis in political science [34] and for helping in automatic corpus creation for Wikipedia [35].

Considering that n and m are the lengths of the two sequences to be compared, it can be shown that both, the time and space resources consumed are O (n · m). In some uses such as bioinformatics, in which the length of the sequences is extremely long, memory consumption is prohibitive, and therefore, optimizations have been proposed such as the Hirschberg algorithm [36] which is able to reduce the space up to O (n + m) but, at the expense of a computation time increment. Other proposals include the Levenshtein distance for synchronizing the videos, minutes and text transcripts, of the Basque Parliament plenary sessions [37], for aligning text with speech audio signals with lengths of up to several hours [38], for automatic bilingual subtitle generation for lecture videos [20] or even for automatic face annotation in TV series by video/script and subtitles alignment [39].

IV. CURRENT SITUATION

As has been seen in the previous section, synchronization of subtitles has been focused on resolving the automatization of their broadcast only for live-scripted programs.

Contrary to the foregoing, this paper focuses on other types of programs; improvised or mixed-live broadcasts, in which there is no synchronization between audio-visual contents and subtitles because there is a temporal displacement (Δ_t) associated with their generation process.

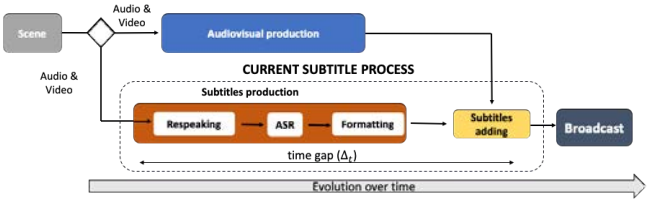

Figure 2. Diagram representing usual subtitles generation by re-speaking in live or mixed type events.

As stated above, for the generation of accessibility services associated with audio-visual content in live and mixed broadcasts, the usual tendency is to follow the process shown in Figure 2. First, a re-speaking process is carried out by means of an ASR component. Afterwards, the transcription from the ASR is formatted in order to assure it meets the current standards [1], [40], [41]. Finally, the subtitles are inserted in the DTT channel. By this time, the audio to which these subtitles correspond has already been broadcasted a while ago.

These steps, as mentioned before, take a time (Δ_t) and prevent proper synchronism between the subtitles and the contents. On the one hand, the re-speaker needs a sentence to be completed or at to be least very advanced, in order to start its repetition (simplified and summarized). On the other hand, the ASR's own internal process makes it impossible to validate a sentence until it has been able to verify its correctness (syntactic and / or semantic depending on the case), since it substitutes words that are acoustically more likely for other which are grammatically more likely. Obviously, this substitution can only be definitive once the sentence is finished or at least, once it is near to its end. These extrinsic delays are directly related to the duration of the locutions and not so much to the speed of human or computational process.

In [42] the performance of the total delay Δ_t is studied. According to the analyzed programming, the average delay was 3.8 seconds, with frequent peaks that reach 15 seconds. However, the most remarkable feature of these Δ_t is its high fluctuation. Thus, to move the process forward a fixed value (i.e. 3.8 seconds) is not a feasible solution. Should this solution be applied, half of the subtitles will be ahead of time and the other half delayed, which in no way represents an improvement.

Therefore, and bearing in mind the current situation described above for improvised or mixed broadcasts, this article presents a solution for obtaining synchronized subtitles through the application of ASR, the use of a temporary reassignment system and the dynamical delay of the signal.

The Sub-Sync framework has been designed to achieve this automatic synchronization, proposing a live-time readjustment between the audio-visual content and subtitles.

V. SOLUTION PROPOSED: SUB-SYNC FRAMEWORK

According to the state of art review included in this paper, nowadays it is not possible to produce the subtitles of improvised live programs in a synchronous way with the audio because the transcription of a previously unknown audio cannot be done in real time. The main cause for this type of delay is the fact that the transcribers, regardless of whether they are human or machines, work on a reactive basis. This delay must be increased by the additional time required by each transcription method. When it comes to practical application, this delay is very relevant, since it profoundly affects the intelligibility of the message by the

users of the subtitles due to: the loss of the correlation between the speaker and the audio; and the lack of congruence between the audio, the image and the subtitle.

Inevitably, to overcome this problem, it is necessary to delay the audio-visual until the corresponding transcription is available and to determine the precise times for the subtitles' presentation and deletion.

This research focuses on the generation of correct synchronization of subtitles as a key element to guarantee a high-quality accessibility service for people with hearing disabilities. For doing this, the Sub-Sync framework enables an automatic synchronization of audio-visual content and subtitles, using speech recognition tools and symbol alignment algorithms based on dynamic programming.

Figure 3 shows the different phases and stages of the process developed by the Sub-Sync framework responsible for the synchronization of the subtitles in parallel with the subtitle transcription process.

The first phase, called transcription, uses an ASR engine to obtain a continuous flow of words corresponding to the audio-visual transcription. This type of tool is widely developed and present both in the scientific literature and in the technological market. For the purpose of this paper, any commercial ASR can be used as long as it provides, along with each transcript word, the timestamp that indicates at what moment the sound corresponding to the particular word was produced. Obviously, the better the recognition accuracy, both in the identification of the word and in the timestamp precision, the better the results produced by the framework. In short, whichever it is, the recognizer provides a flow of words labelled with their temporary mark information that is kept in the transcription storage as long as it is necessary, which will cease to be so once the subtitle to which it corresponds has been processed.

In phase 2, the aim is to identify what would have been the correct moment in which each of the subtitles should have been presented, a process that the authors have called chronization. The two available sources of information are used: the transcription storage and the flow of subtitles provided by the production of the program. The objective here is to discover which words of the transcription storage correspond to the subtitle and from them and their timestamps determine their presentation moment. Erasure will be established according to the length of the subtitle.

It is assumed, given the circumstances described above, that when the subtitle is received, the words that correspond to it are already in the transcription storage. Therefore, a subtitle memory is not implemented, which would further delay its emission. This phase is the core of the Sub-Sync framework, and thus a more detailed description of this phase will be given later herein.

Due to the process of a typical ASR system, along with the definitive transcription, all the different attempts of partial transcription that the machine proposes while the locution is taking place are obtained. The framework also uses these attempts, as they are valuable and early information that gives robustness to the process.

The objective is to determine, when possible, the best alignment between the subtitles and the sequence of words of the transcription storage. Once this alignment is established, it will be possible to infer the correct presentation and deletion times of the subtitle. The algorithms presented in the literature are used for subtitles that fit perfectly with the audio when there is good sound quality [4], [15], [16], [29]. When these circumstances change, it is not possible to find a word-to-word correlation between subtitles and audio. This leads to the fact that sometimes there is no similarity between the recognized audio and the subtitle text. Among the reasons are: it has not been possible to recognize them, they have not been said, variations have been said, or words have been said that are not in the subtitle.

Despite the foregoing, it is common for the same (or at least similar) words to exist in the recognized text and the subtitle. These words will be more useful the longer their length is. The algorithm used must consider this in its design, since using equivalences between very short words, such as prepositions, articles, etc., is risky because these similarities can appear anywhere. Finding correspondences in short texts, such as subtitles, within much longer other texts, such as the transcriber's memory, presents additional difficulties, because, under equal conditions, the algorithms used tend to prefer certain temporary spaces.

Finally, the third phase of the process, called synchronization, corresponds to the construction of the synchrony between the subtitles and the audio-visual content, whether broadcast or reception. For this phase, it is necessary to store the audio-visual material for a long enough time for the subtitles to be available, and when appropriate, to associate them temporarily to the new broadcast timings. This phase can be implemented in the broadcast header or in the user's tuner.

The Sub-Sync framework has been tested successfully in different live broadcasts on DTT, including mixed programs where synchronized parts (recorded, scripted) are interspersed with desynchronized (improvised). It is in the transitions of the mixed programs that serious synchronism problems appear between the subtitles and the audio-visual contents.

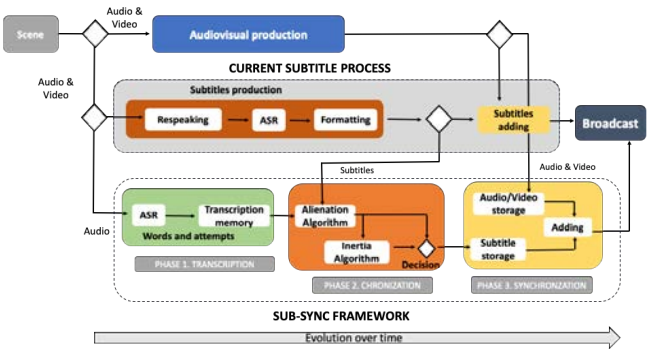

Figure 3. Synchronization of subtitles using the Sub-Sync framework.

A. PHASE 1. TRANSCRIPTION

1) UNDERSTANDING THE ASR OUTPUT

A typical ASR provides a set of acoustic approaches to the original locution. These approaches have been called *attempts* in this study Attempts are sequences of words that correspond to the part of the sentence issued. These attempts provide variations on words that are sometimes similar to those expressed by the speaker in the traditional system analyzed in Section IV and shown in Figure 3. The words that constitute the attempts correspond to the most probable transcription at a given time. At the beginning of the sentence, the attempts are primarily phonetic approximations, but as the sentence progresses, the grammatical component becomes more relevant. Thus, when the sentence is concluded and the final transcription is issued, there will be words that have been modified throughout the transcription process.

The framework, for each position (word) of the transcription, rejects the repetitions in successive attempts, but stores the different alterations in the memory. This results in several words occupying the same temporary place, which in no way affects the chronization.

```
1  I'll know
2  I'll never
3  I'll never be
4  I'll never be home
5  I'll never be hungry
6  I'll never be hungry at
7  I'll never be hungry again
8  I'll never be hungry again no
9  I'll never be hungry again no no
10 I'll never be hungry again no
11 I'll never be hungry again no nor
12 I'll never be hungry again no North
13 I'll never be hungry again no Nora
14 I'll never be hungry again no Norton
15 I'll never be hungry again no nor any
16 I'll never be hungry again no nor any of
17 I'll never be hungry again no nor any of my
18 I'll never be hungry again no nor any of my phone
19 [...] no nor any of my phone
20 [...] no nor any of my fault
21 [...] no nor any of my folks
22 [...] no nor any of my folks if
23 [...] no nor any of my folks if I
24 [...] no nor any of my folks if I have
25 [...] no nor any of my folks if I have to
26 [...] no nor any of my folks if I have to lie
27 [...] no nor any of my folks if I have to lie still
28 [...] no nor any of my folks if I have to lie steal
29 [...] if I have to lie steal cheat
30 [...] if I have to lie steal cheat or
31 [...] if I have to lie steal cheat or kill
32 [...] if I have to lie steal cheat or kill as
33 [...] if I have to lie steal cheat or kill as go
34 [...] if I have to lie steal cheat or kill as God
35 [...] if I have to lie steal cheat or kill as God is
36 [...] as God is my
37 [...] as God is my witness
38 [...] as God is my witness I'll
39 [...] as God is my witness I don't
40 [...] as God is my witness I'll know
41 [...] as God is my witness I'll never
42 [...] as God is my witness I'll never be
43 [...] as God is my witness I'll never be home
44 [...] as God is my witness I'll never be hungry
45 [...] as God is my witness I'll never be hungry at
46 [...] as God is my witness I'll never be hungry again
```

Figure 4. Sequence of generation of attempts.

Figure 4 shows an example of the output of an ASR. As the locution progresses, the recognizer incorporates new words and / or modifies the previous ones so that the final transcription is grammatically correct.

2) INFORMATION CONTAINED IN THE SUBTITLES

There are many formats in which the subtitles can be transmitted to the addition stage. For the purposes of this study, XML is very propitious because it easily allows the additional incorporation of transparent information into the various stages and is useful for the evaluation of the system's performance.

A subtitle contains the information of, not only the attributes of the text and the text itself, but also the instants of presentation (insertion) (t_i) and of erase (erase) (t_e). However, in the proposed scenario it must be assumed that these data are not correct, since it is very likely that they do not match precisely with the moment in which the corresponding locution has taken place. It is necessary to

estimate other times (t_i' *and* t_e') that locate the subtitle at its appropriate time. There is a delay between the phrase and its estimated subtitle; $\Delta = t_i - t_i' \geq 0$, which must be compensated in the synchronization process.

3) AUDIO-VISUAL INFORMATION STORE

As previously anticipated, this compensation needs to store the audio-visual in a memory for as long as it is necessary. As this time cannot be variable, the most relevant parameter to be evaluated, in the development of the solution is the quantification of the constant delay (Δ_v) (discarding the audio-visual timing).

The value of Δ_v must ensure that the recalculated subtitles will be available at the time when its corresponding audio-visual fragment is reproduced.

Considering Δ_i as the individual delay of a subtitle and that the synchronization is done by the broadcaster (before the broadcast), Δ_v should not be less than the maximum delay ($\Delta_/ = max\{\Delta_i\}$), since in the necessary time for the synchronization process (T_s) must be added in. T_s includes the processing time of the ASR.

$$\Delta_v = \Delta_m + T_: \quad (1)$$

Depending on the architecture, it will be necessary to consider other delays, such as the transmission time required between the headend and the computer where resynchronization takes place; or the propagation time of the audio-visual in the production system.
The lower the value chosen for Δ_v, the lower the disruption caused by the delay, but the probability that a subtitle will not be available at its due time is increased.
It is necessary to design a strategy to address the circumstance of the unavailability of the subtitle when the broadcast of its audio-visual fragment takes place; the simplest option is to ignore it and ignore its presentation. When this situation is unlikely, an alternative is to recover the gap little by little, slightly reducing the duration of the following subtitles and taking advantage of the breaks.
If the synchronization is done by the receiver, Δ_v is affected by the propagation time through the network of the subtitles (T_{ps}) and the audiovisual (T_{pa}).

$$\Delta_: = \Delta_m + T_: + T_{p:} - T_{pa} \quad (2)$$

In the case of TV, experience indicates that (T_{ps} - T_{pa}) can take negative values, which in some cases could, *de facto*, produce a reproduction without delay.

B. PHASE 2. CHRONIZATION

As stated earlier, the chronization process is the core of the framework. Its aim is to identify the correct (past) time in which each of the subtitles will have been presented. The solution proposed in this article for this part of the process is based on dynamic programming techniques.

The inputs to this phase, which are received in real time, are the subtitles obtained by the traditional system together with the words saved in the transcription storage. The objective is to develop an algorithm that:

- Finds the best possible alignment between the words that form the subtitle and those of the sequence of words contained in the transcription storage.
- Considers that there will be omitted, additional and modified words.
- Establishes an assessment method that compares the different alignment alternatives, giving more importance to the alignment of long words rather than short ones.
- It also considers the alignment of similar words.
- It gives more weight to the early alignment attempts rather than the later ones.

Taking into account these needs, an alignment algorithm that is based on the one proposed in 1970 by Needelman and Wunch [8], [9], [11] has been chosen. This algorithm was originally created for the identification of amino acid chains in proteins but has demonstrated its efficiency for the alignment of symbol sequences, whatever they may be, such as finding similarities between texts.

The solution proposed by the authors uses an adaptation of this algorithm to obtain as a result the best match between the words of the subtitle and the words stored in the transcription storage (considering the score calculated for that alignment). Figure 5 shows an example of this pairing process.

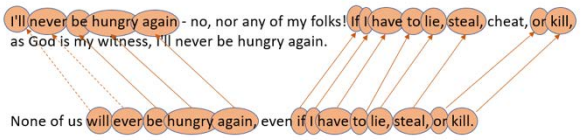

Figure 5. Example of text alignment.

In the following sections, a description of the algorithm is presented along with the adaptation of the original algorithm to the circumstances of the framework.

1) DEFINITIONS AND ALGORITHM

Let W be the set of symbols (words) available to construct the sequences of the problem and the symbol *null* represents "no symbol".

Let S be a subtitle built by an orderly sequence of symbols "s_i" that belong to the dictionary W (excluding null) and that has an overall length L_s:

$$S = \{s_i | s_i \in W - \{null\}\}$$
$$L_s = Card(S) \quad (3)$$

Let T be a transcription built by a sequence of words "t_i" that belong to the dictionary W (excluding null) and that has an overall length L_t:

$$T = \{t_i | t_i \in W - \{null\}\}$$
$$L_s = Card(S) \quad (4)$$
$$L_t \gg L_s$$

Let δ be a distance function that evaluates the dissimilarity between two symbols of W:

$$\delta : W \times W \Rightarrow \mathbb{R} \quad (5)$$

Let A be a vector of pairs of symbols contained in W:

$$A = \{a_S = (w_{S1}, w_{S2}) \in WxW\} \quad (6)$$

The vector A can be constructed to represent an alignment between symbols of S and symbols of T, each pair will indicate the position and the correspondence between both symbols (including the association with *null*).

Consider a Table M, with size (L_s+1)x(L_t+1), where each column, starting from the 1, corresponds to a symbol of T, and each row, starting from 1, corresponds to one of S, as the example shown in Figure 6.

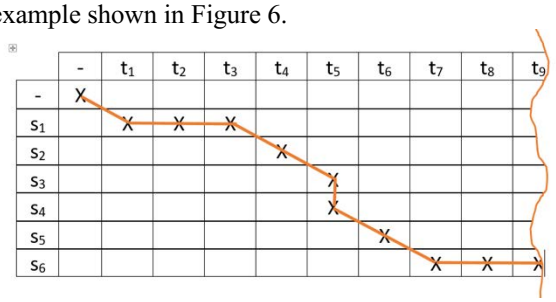

Figure 6. Alignment Table M

In this table an alignment would correspond to a sequence (represented by the continuous path) of cells (represented by the X's). In order to be valid for the purposes of this work, the sequence must start at $m_{0,0}$ and end at $m_{Ls,Lt}$ without there being any steps backwards:

$$\left.\begin{array}{l} a_S = (s_i, t_j) \\ a_{S'} = (s_{i'}, t_{j'}) \\ k' = k+1 \end{array}\right\} \Rightarrow \begin{cases} i' = i+1 & j' = j+1 \\ i' = i+1 & t_{j'} = null \\ s_{i'} = null & j' = j+1 \end{cases} \quad (7)$$

In Table M, a horizontal step means that a symbol of T could not be matched to a symbol of S, and a vertical step represents that a symbol of S could not be matched to a symbol of T.

Based on the path described by the alignment, a cost function C can be defined as the sum of the costs of each of its steps and these as the distance between the members of the pair.

$$C(A) = \sum_S C(a_S) = \sum_S \delta(s_{S1}, t_{S2}) \quad \forall a_S \in A \quad (8)$$

Although other solutions could have been considered, for this work, a constant value of the distances to the *null* symbol has been accepted.

$$\delta(s_i, null) = C_s$$
$$\delta(null, t_j) = C_t \quad (9)$$

2) NEEDLEMAN-WUNCH ALGORITHM

The aim of the algorithm is to find the vector A that minimizes the cost function.

$$A_{odt} = minarg(C(A)) \quad (10)$$

Table 1 shows the interpretation of the previous example.

Table 1. Example of the costs of an alignment

A	S	T	Cost
a_1	s_1	t_1	$C_1 = \delta(s_1, t_1)$
a_2		t_2	$C_2 = C_T$
a_3		t_3	$C_3 = C_T$
a_4	s_2	t_4	$C_4 = \delta(s_2, t_4)$
a_5	s_3	t_5	$C_5 = \delta(s_3, t_5)$
a_6	s_4		$C_6 = C_S$
a_7	s_5	t_6	$C_7 = \delta(s_5, t_6)$
a_8	s_6	t_7	$C_8 = \delta(s_6, t_7)$
...			...

The Needleman-Wunch (NW) algorithm proposes to fill Table M in the following way:

- Consider value zero in cell (0,0).
- Fill in row 0 by increasing each cell by C_T, additionally an arrow pointing to the left is written on the cell (left).
- Fill in column 0 by increasing each cell by C_S, additionally an arrow pointing upwards is written on the cell (up).
- Fill in the rest of the cells with the value that minimizes reaching to it (see Equation 11). Additionally, an arrow is written on the cell that

indicates the direction of the minimum cost (left, up, diagonal).

$$\begin{gathered} m_{0,0} = 0 \\ m_{0,j} = m_{0,j\mathrm{g}\,1} + C_{\mathrm{h}}\ \forall\, j = 1 \ldots L_s\ (left\ arrow) \\ m_{i,0} = m_{i\mathrm{g}\,1,0}\ C_{\mathrm{h}}\ \forall\, i = 1 \ldots L_t\ (up\ arrow) \\ m_{i,j} = min \begin{cases} m_{i\mathrm{g}\,1,j\mathrm{g}\,1} + \delta(s_i, t_j)\ (diagonal\ arrow) \\ m_{i\mathrm{g}\,1,j} + C_{\mathrm{h}}\ (left\ arrow) \\ m_{i,j\mathrm{g}\,1} + C_s\ (right\ arrow) \end{cases} \end{gathered} \tag{11}$$

Thus, the cell m_{L_s,L_t} contains the minimum cost needed to reach it from $m_{0,0}$ and the arrows indicate the path followed to obtain that cost.

To determine the best route (A_{opt}) it will begin with m_{L_s,L_t} and progress backwards following the route indicated by the arrows.

As an example, consider the sequence S of length 6 and the sequence T of length 19. A very simple function of distance (δ) represented in Figure 7, where $C_H = C_T = 2$.

	-	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	t_{11}	t_{12}	t_{13}	t_{14}	t_{15}	t_{16}	t_{17}	t_{18}	t_{19}
-																				
s_1		0	2	2	2	2	2	0	2	2	2	2	2	0	2	2	2	2	2	2
s_2		2	2	2	0	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
s_3		2	2	2	2	0	2	2	2	2	2	2	2	2	2	2	2	2	2	2
s_4		0	2	2	2	2	2	0	2	2	2	2	2	0	2	2	2	2	2	2
s_5		2	2	2	2	2	0	2	2	2	2	2	2	2	2	2	2	2	2	2
s_6		2	2	2	2	2	2	0	2	2	2	2	2	2	2	2	2	2	2	2

Figure 7. Distance function

With this information, Table M presented in Figure 8 can be constructed, where the shaded cells represent the best possible path obtained by the algorithm. The total cost is 30.

	-	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	t_{11}	t_{12}	t_{13}	t_{14}	t_{15}	t_{16}	t_{17}	t_{18}	t_{19}
-	0	2	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34	36	38
s_1	2	0	2	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34	36
s_2	4	2	2	4	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34
s_3	6	4	4	4	6	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32
s_4	8	6	6	6	6	6	6	6	8	10	12	14	16	18	20	22	24	26	28	30
s_5	10	8	8	8	8	8	6	8	8	10	12	14	16	18	20	22	24	26	28	30
s_6	12	10	10	10	10	10	8	6	8	10	12	14	16	18	20	22	24	26	28	30

Figure 8. Example of Table M associated with the algorithm

3) VARIATION OF THE NW ALGORITHM

In the context of this research, as it has been exposed in the previous section, the algorithm does not consider some characteristics that are specific to the alignment between subtitles and transcriptions. Although other approaches such as the Hirschberg algorithm [30] reduce the complexity of the algorithm, they increase the computation time. In the context of this research, the time is a critical factor. For this reason, Needleman-Wunsch [11] [12]–[14] was applied in order to avoid delays in the computational time.

One of these characteristics is that the length of the transcriptions is much greater than those of the subtitles ($L_t \gg L_s$). The algorithm, in its original form, considers symmetry in the process. However, in the present case, what is intended is to find the best position of the subtitle in the transcription and not the other way around.

Another specific characteristic is that the keywords associated with the topic of the TV program appear repeated and these repetitions become systematic in the transcription as they correspond to a long period of time (Δ_t). The algorithm in its original formulation tends to find associations, preferably at the end of the transcription, when it does not really correspond to the current subtitle but to a future one.

As can be seen in Figure 8, the cell *(S_6, T_7)* with a distance of symbols: *$\delta(S_6, T_7)=0$* and cost *$C(S_6, T_7)=6$*, which is the lowest total cost of the row S_6. This means that at this point, the algorithm has already found the end of the best path for the alignment of all the words of the subtitle (S_1 to S_6), although ignoring some of the transcription's words (T_8 to T_{19}).

To incorporate the absence of symmetry and the preference of early associations into the algorithm, the authors have opted to search in the row L_s of the matrix M, the best cost and use the first cell in which it is found instead of *(L_s, L_t)* for later, rebuilding backwards the best path.

An additional consideration is that, as mentioned above, for each position the system provides several attempts. Thus, it will be usual for an element of the transcript to have no match with any element of the subtitle. Symmetrically, since the subtitles do not literally correspond to the locutions, it will not be unusual for an element of the subtitle not to correspond to any element of transcription. In both cases, the circumstance will not have an important impact on the cost function.

4) ALIGNMENT QUALITY

It will not always be possible to find a good alignment between subtitle and transcription, which is defined as an alignment with a cost below a certain value or predefined threshold. The optimal value of this parameter depends basically on the style with which the repeater performs his duty. Fortunately, the range of acceptable values of this maximum cost is very wide, and therefore, the application of a reasonable minimum value results in a good behaviour.

When the alignment found is acceptable, the framework determines the presentation and deletion times based on the calculated alignment, otherwise, another method is

necessary to estimate the temporal marks, for which the authors have opted for the "inertia algorithm".

5) ESTIMATED PRESENTATION TIME BASED ON ALIGNMENT

The result of the alignment process produces a vector *A* that consists of the pairs of symbols of *S (subtitles)* and *T* (transcripts) where the $t_j \in T$ have an associated timestamp provided by the ASR that indicates when the word has been said.

The estimation of the presentation time of the subtitle (t'_{ps}) can be made from the moment of appearance of the word in the transcription storage (*t*) aligned with the best cost and from the position (*k*) of this word in the subtitle. Assuming that, on average, a Spanish speaker pronounces a word every 0.385 s, the presentation time can be calculated as:

$$t'_{ps} = t - k \cdot 0.385 \qquad (12)$$

Table 2. Example of the results of an alignment (part I)

Index	**0**	**1**	**2**	**3**	**4**	**5**	**6**
Time				**2.00**	**2.22**	**2.53**	**2.67**
Transcr.				I'll	never	be	hungry
Subtitle	none	of	Us	will	ever	be	hungry
Distance	1	1	1	0.25	0.8	0	0
Score	0	0	0	1	3.6	2	6

Table 3. Example of the results of an alignment (part II)

Index	**7**	**8**	**9**	**10**	**11**	**12**	**13**
Time	**3.06**	**4.40**	**4.73**	**4.93**	**4.06**	**4.20**	**4.53**
Transcr.	again	no	nor	any	Of	my	folks
Subtitle	again						
Distance	0	1	1	1	1	1	1
Score	5	0	0	0	0	0	0

In the example proposed in Table 2 and Table 3, the score has been calculated as:

$$score_S = (1 - distance_S) \times l \qquad (13)$$

where l is the length of the word of the subtitle.

In that case "hungry" is the word with the best score (6), the transcription time for it is *t*= 2.67 and its position within the subtitle is *k* = 6

$$t'_{ps} = t - k \cdot 0.385 = 0.67 - 0.385 * 6 = 0.36 \qquad (14)$$

6) INERTIA ALGORITHM (CHRONIZATION BY INERTIA).

As previously mentioned, when a re-speaker is used, his locution is characterized by the lack of literalness, which means that in many cases there is no alignment with enough quality to be used. In these cases, the estimation of the presentation time is made based on the following assumption:

"The delay introduced by the re-speaking process should be similar to the delay introduced to the most recent subtitles."

The algorithm that implements it, designed by the authors, has been called "Inertia Algorithm". The calculation of presentation time and inertia can be expressed for the i-nth subtitle, depending on whether there is alignment or not, as:

$$\forall i \mid C(\square, T) \leq t\square \begin{cases} R_s = t_{ps} - t'_{ps} \\ I_s = \frac{1}{2}(R_s + I_{sg\,1}) \end{cases} \qquad (15)$$

$$\forall i \mid C(\square, T) \square\ t\square \begin{cases} I_s = I_{sg\,1} \\ t'_{ps} = t_{ps} + I_s \end{cases} \qquad (16)$$

t_{ps}*: original presentation time,* $t'_{ps}\square$ *presentation time obtained by alignment when possible,* R_s*: delay of the subtitle,* I_s*: value of inertia calculated for the subtitle, th: maximum permitted cost.*

When the cost exceeds the maximum, the subtitle is advanced with respect to its original time the inertial value *(I)* which is to be calculated using only the aligned subtitles.

7) ERASE TIME CALCULATION

Once the presentation time has been established, it is necessary to determine the deletion time. One possibility is to use the original duration, that is:

$$\begin{cases} \square = t_{bs} - t_{ps} \\ t'_{bs} = t'_{ps} + \square = t'_{ps} + (t_{bs} - t_{ps}) \end{cases} \qquad (17)$$

d: duration of the subtitle, t_{ps}, t_{bs}*: original presentation and deletion times,* t'_{ps}, t'_{bs}*: chronized presentation and deletion times.*

However, these durations may not correspond to reasonable values, since, as previously mentioned, in the transitions of improvised (live) to scripted, the TV channels reduce the subtitles' duration to recover the lost time.

The solution proposed is to calculate the duration considering the maximum reading speed, which, in the case of Spain, that recommended by the standard AENOR UNE-153010 [40] is 15 characters per second. Thus, while time-variables are expressed in seconds, it can be calculated as:

$$t'_{bs} = t'_{ps} + \square = t'_{ps} + len\square t\square(\square ubtitle)\square 15 \qquad (18)$$

C. PHASE 3. SYNCHRONIZATION

The final stage of the process is the construction of the synchrony between the subtitles and the audio-visual either in the broadcast or in the reception.

This stage focuses on adding the delay that has been applied to the audio-visual to the timestamps recalculated by the chronizator.

There are two possible points where the synchronization is performed: in broadcast or in reception.

In broadcast implies that the delay of the audio-visual must be made by the program's broadcaster. Under those conditions, all programs should be delayed by the amount of time that the authors have proposed of 20s. In [33] a time of 15s was proposed but has been increased to 20s in order to cover most of the cases without delaying the broadcast excessively. This solution would allow standard tuners to be used, without the adaptation described in the following paragraph, but there is a certain reticence on the part of broadcasters to implement this type of audio-visual manipulation.

In reception implies that it is the DTT-tuner that performs, on one hand, the accumulation of the audio-visual during the stipulated time, and, on the other hand, the presentation of the subtitles at the correct time. This solution leads to two problems. The first one is that when changing a channel, there will be 20 seconds where it is not possible to present a correct subtitle. The second one is that the determination of the time of insertion of the subtitle is affected by an additional parameter: the propagation time of the signal, which is variable and difficult to determine precisely.

VI. EXPERIMENTAL RESULTS

When estimating the validity of the hypotheses set out in this article, the chronization described above has been tested on three programs of three different Spanish TV channels, captured directly on a DTT antenna, for both the audio-visual and the subtitles:

- Two parts of two current magazine-programs broadcasted by two different TV channels, and
- A part of a sports program (live broadcast of a football match) of a sports channel, which is a complicated matter for an ASR due to the poor audio-quality, in which, as any live sport event, the speakers' voice is mixed with audio background noise.

A manual alignment of the subtitles was carried out for the three tests, thereby establishing the reference values of both the presentation and deletion times. These values allowed the framework's performance to be evaluated.

In all cases, the ASR used was Google. Google has shown good results in speech to text tasks with low training time. Google Speech API also allows inclusion of context words in order to increase the quality of the results The Google Cloud Speech-to-text services allows streaming speech recognition providing text alternatives to the speech recognized with a percentage of confidence. As mentioned, these alternatives are called attempts in the Sub-Sync framework (see subsection A of section V). The alternatives recognized by the Google API are considered in the transcription process.

The distance function between words (δ) has been chosen considering the characteristics of the language of the locutions (Spanish). Other languages will need other designs depending on their characteristics. The function is expressed in Equation 19, where *l* is the largest length of characters of the words and *n* is the length of the largest common initial sequence.

$$\begin{gathered} \delta(w_1, w_2) = 1 - {}^{n}/_{l} \\ l = \max(length(w_1), length(w_2)) \\ n = length(common\_radix(w_1, w_2)) \end{gathered} \qquad (19)$$

As an example:

w1 = "olvidado"
w2 = "olvidó"
length(w1) = 8;
length(w2) = 6
l = max(8,6) = 8
common_radix = "olvid"
leng(common_radix) = 5
δ = 1- 5/8 = 0.375

The cost threshold for a valid alignment was established similarly in the three cases: $th = 0.58$.

A. Scenario 1

For the evaluation of the Sub-Sync framework, different scenarios were studied. For each scenario, one fragment of audiovisual content was analysed. In Table 4, the main characteristics of Fragment 1 are detailed.

Figure 9 shows the percentage of subtitles (in y-axis) with a determined delay in seconds (in x-axis). The delay time indicates the difference in seconds between the audio broadcast until the appearance of the subtitle on the screen. In this scenario, fragment 1, the subtitles provided by the TV channel appeared with respect to the audio with an average delay of 9.771 seconds and a standard deviation of 2.522 seconds. The distribution of the delays is shown in Figure 9, which indicates a mode of 9 seconds.

Table 4. Fragment's 1 characteristics.

TV channel:	General
Program:	Magazine
Duration:	30:45
Number of subtitles:	335

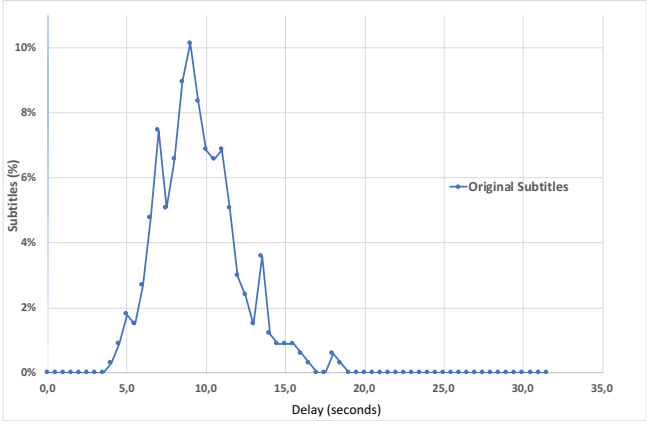

Figure 9. Distribution of the delay of the original fragment's 1 subtitles (original subtitles).

In Figure 10, a comparison between the original broadcasting and the chronized subtitles of Sub-Sync is shown. In this case, the subtitles obtained as an output of Sub-Sync have an average delay of 0.453 seconds with a standard deviation of 1.974 seconds. The distribution of the delays is shown in Figure 10, which indicates a mode of 0 seconds. As can be seen, sometimes the synchronized subtitles come a bit ahead (because of the accuracy of the inertia algorithm, which averages the placement of subtitles that do not match the ASR).

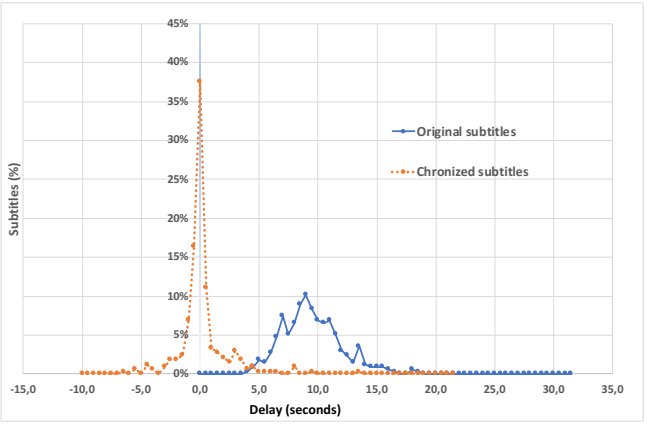

Figure 10. Original vs chronized subtitles comparison (fragment 1).

Data breakdown according to the type of algorithm used for the subtitles alignment (NW algorithm or "aligned" and Inertia Algorithm or "inertied") has been carried out and both distributions results are shown in Figure 11. The aligned ones were 213 subtitles with an average of 0.303 seconds and a deviation of 0.988 seconds. The "inertied" were 122 subtitles with an average of 0.714 seconds and a deviation of 2.990 seconds.

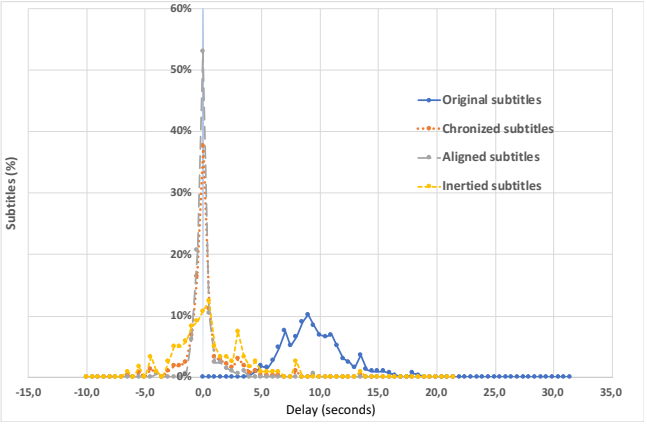

Figure 11. Algorithms comparison for fragment 1.

B. Scenario 2

For the second scenario, the main characteristics of the audio-visual fragment are detailed in Table 5.

Table 5. Characteristics of fragment 2

TV channel:	General
Program:	Magazine
Duration:	30:41
Number of subtitles:	521
Average delay	3.000 seconds
Deviation	4.146 seconds
Mode	2.500 seconds

Figure 12 shows the comparison between original and chronized subtitles.

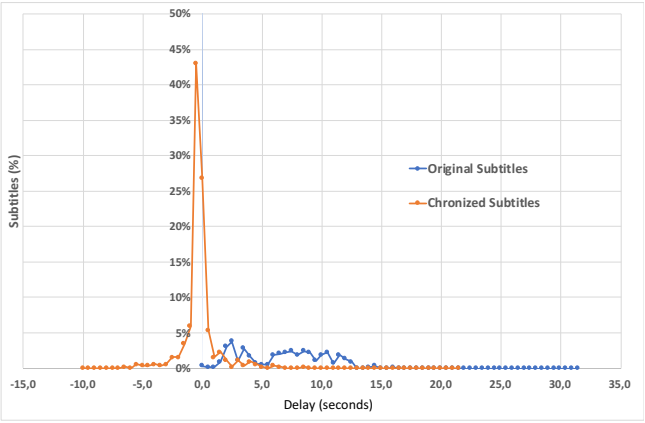

Figure 12. Original vs chronized subtitles comparison (fragment 2)

Table 6. Fragment 2. Data for chronized subtitles.

	Global	Aligned	Inertied
Number of subtitles	521	407	114
Average delay (seconds)	0.016	-0.007	0.099
Deviation (seconds)	1.384	0593	2.746
Mode (seconds)	-0.500	-0.500	-1.500

The subtitles obtained as an output of Sub-Sync are detailed in Table 6. Figure 13 shows a comparison between the different algorithms for this scenario.

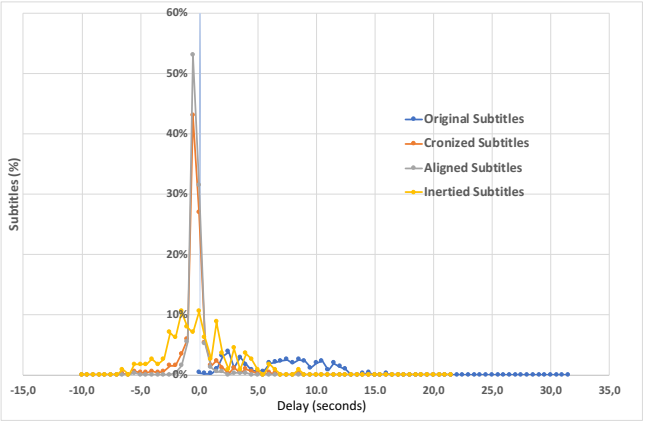

Figure 13. Algorithms comparison for fragment 2.

C. SCENARIO 3

For the last scenario of the experimentation, the main characteristics of fragment 3 are detailed in Table 7.

Table 7. Characteristics of fragment 3

TV channel:	Sports
Duration:	29:59
Number of subtitles:	355
Average delay	10.203 seconds
Deviation	2.898 seconds
Mode	10.000 seconds

Figure 14 shows the comparison between original and chronized subtitles.

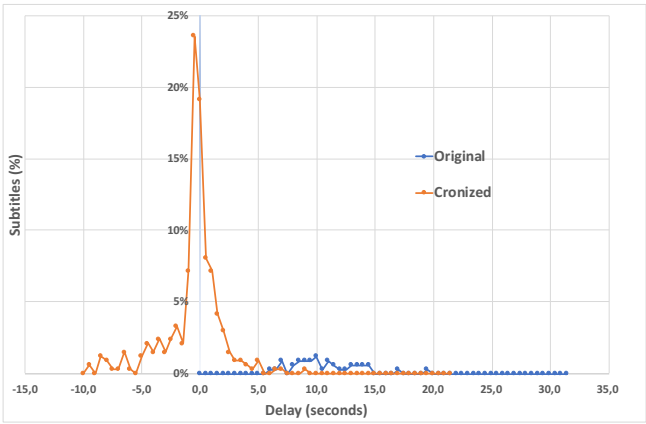

Figure 14. Original versus chronized comparison. Fragment 3

The subtitles obtained as an output of Sub-Sync are detailed in Table 8. Figure 15 shows a comparison between the different algorithms and their distribution.

Table 8. Fragment 3. Data for chronized subtitles.

	Global	Aligned	Inertied
Number of subtitles	335	179	156
Average delay (seconds)	-0.368	0.173	-0.990
Deviation (seconds)	2.509	0.867	3.461
Mode (seconds)	-0.500	-0.500	-1.000

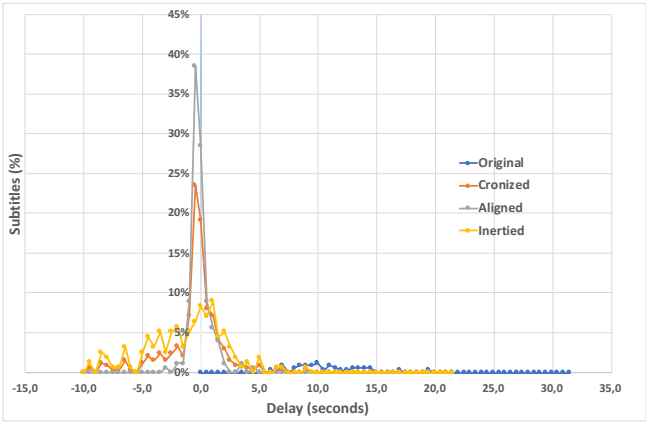

Figure 15. Algorithms comparison for fragment 3.

D. COMPARISON BETWEEN NEEDLEMAN-WUNSCH AND LEVENSHTEIN ALGORITHMS

As has been exposed in Section III, another typical algorithm for measuring distance between words is the Levenshtein algorithm. The Levenshtein distance is a string metric for measuring difference between two sequences. The Levenshtein distance between two words is the minimum

number of single-character edits (i.e. insertions, deletions or substitutions) required to change one word into the other.

In this section, two different comparison are performed. In the first one, the performance obtained from the Needleman-Wunch algorithm in scenario 1 is compared with the Levenshtein algorithm (see Table 9).

Table 9.Results obtained in scenario 1 for the Needleman-Wunch and the Levenshtein algorithms.

Algorithm	Output	Average Delay (sec)	Standard deviation (sec)
Re-speaking		9,771	2,522
Manual Synchro.		0,453	1,974
Needleman-Wunch	Aligned	0,303	0,988
	Inertied	0,710	2,990
Levenshtein	Aligned	0,233	2,543
	Inertied	1,228	3,015

In the Table 10, the performance obtained from the Needleman-Wunch algorithm in scenario 2 is compared with the Levenshtein algorithm.

Table 10. Results obtained in scenario 2 for the Needleman-Wunch and the Levenshtein algorithms.

Algorithm	Output	Average Delay (sec)	Standard deviation (sec)
Re-speaking		8,459	3,540
Manual Synchro.		0,160	1,384
Needleman-Wunch	Aligned	0,007	0,593
	Inertied	0,099	2,746
Levenshtein	Aligned	0,939	2,100
	Inertied	0,150	3,000

The comparison of the NW and the Levenshtein algorithms indicates that the performance of both is similar (in both cases the average delay is reduced to less than 1 second, although the standard deviation is greater with the Levenshtein algorithm). Thus, in the future research could be done in this direction. Both algorithms obtain better performance than re-speaking and the Needleman-Wunch algorithm is quite similar to the manual synchronization.

E. DISCUSSION

Regarding the experimental results, the framework proposed allows us to synchronize subtitles even if they do not correspond literally with the original audio and/or the audio is not one hundred percent susceptible to being transcribed automatically. In addition, the proposed system works in any scenario, regardless of how the subtitles are generated.

The Sub-Sync framework has been successfully tested in different live broadcasts on DTT, including mixed programs where synchronized parts (deferred, scripted) are interspersed with desynchronized (improvised). As pointed out previously, it is in the transitions of the mixed programs where serious synchronism problems appear between the subtitles and the audio-visual contents.

At the beginning of the research, the Needleman-Wunch algorithm was selected because this measure enables sharing of character sequences allowing "gaps", which is what happens in the re speaking, since the audio is normally summarized (and therefore it seemed initially more optimal).

When comparing it with the Levenshtein algorithm (that it less complex), the results are similar. Thus, the authors consider that it is another way to approach the research since it is based on a simpler algorithm that can be reinforced with the characteristics of the problem (for example, you have to look for the most probable subtitle in the following subtitles in a small time interval).

Finally, the experimental results show that the performance of the framework is linked to the quality of the transcription. In very noisy environments, such as fragment 3, when the ASR finds more problems converting the discourse into text, the Sub-Sync presents slightly worse results than in more controlled scenarios, but even so, it is a much higher quality service than the original one.

VII. CONCLUSIONS AND FUTURE LINES

One of the main complaints made by the TV's users is the lack of synchronism between audio and subtitles in some types of programs. Moreover, delay remains as one of the most significant factors in the audience's perception of quality in live-originated TV captions or subtitles for the deaf and hard of hearing.

In addition, according to the state of art included in this paper, today it is not possible to produce subtitles for improvised live programs in a synchronous way with the audio because the transcription of a previously unknown audio cannot be done in real time. The main cause for this type of delay is the fact that the transcribers, regardless of whether they are human or machines, work in a reactive basis. This delay is increased by the additional time required by each transcription method.

When it comes to practical application, this delay is very relevant, since it profoundly affects the intelligibility of the message for the users of the subtitles due to: the loss of the

correlation between the speaker and the audio; and the lack of congruence between the audio, the image and the subtitle.

Inevitably, to overcome this problem, it is necessary to delay the audiovisual until the corresponding transcription is available and determine the precise times for the subtitles' presentation and deletion.

Accordingly, the idea behind this research is tackling the main technical problem faced by accessible television for overcoming communication barriers that affect a large number of people, thereby generating an inclusive solution for all. In this context, the subtitling of live and mixed television programs constitutes the main current technological challenge for the accessibility of programs that are broadcast in real time on television networks. Improvement in quality is currently the main demand from users of subtitles. In most of these programs, there is no possibility to anticipate the transcription to text of the audio program for its conversion to subtitles and broadcast on the television signal. Considering these issues, this paper presents a framework that is able to synchronize subtitles even when they do not correspond literally to the original audio and/or the audio cannot be fully transcronrf by an automatic process.

To reflect upon the possible scope of this proposal in a numerical way, it is worth noting that, if all of the people with sensory disabilities lived in the same country, it would be the world's third most populated country, with approximately 375 million inhabitants, only behind China and India.

The proposed system works for any scenario, regardless of how the subtitles are generated. The Sub-Sync framework has been successfully tested in different live broadcasts on DTT, including mixed programs, in which synchronized parts (recorded, scripted) are interspersed with desynchronized (improvised) ones. It is in the transitions of the mixed programs where there are serious synchronism issues between the subtitles and the audio-visual content.

According to the tests results, the framework's performance depends first upon the quality of the transcriptions, because if the ASR is not able to transcribe the sound, the system will not match the subtitles properly. For example, in very noisy environments, as is the case of the third test, the ASR has more problems in transcribing the speeches into text. Accordingly, Sub-Sync presents slightly poorer results than in more controlled scenarios. Even so, the results are much better than the original. Other scenarios in which ASR quality decreases are related to the grammaticality or the speed of the speaker, among other reason.

As a future research line, for certain situations, such as when changing the program or the TV channel, it may be interesting to incorporate a module that detects some type of signal that indicates whether the subtitles are already synchronized by the broadcaster in order to use or not the information generated by the chronizator. In addition, it currently the framework's adaptation to other languages than Spanish is currently being evaluated. In this case, the tests to be performed include checking the capacity of the ASR component within the framework, the evaluation of the attempts obtained by the ASR, as well as how to reassess the equations of the chronization process associated with the inertial and alienation algorithms Further research could be focused on testing other ASR systems and their impact on the performance of the Sub-Sync. In this sense, the process starts once the ASR provides the transcriptions. Different ASR systems could be applied and their impact on the transcription time will be analyzed. In a regular scenario, this time is included in the time for the synchronization process (T_s). In this research, we have considered the regular scenario in which the transcription is available in a few seconds. Once the first transcription has been received, the next ones are received as a pipeline. Future research will include an analysis of the impact of the ASR time as well as a threshold in which if no response has been received from the ASR the inertia algorithm will show the subtitles.

REFERENCES

[1] J. Díaz Cintas, P. Orero, and A. Remael, *Media for all: subtitling for the deaf, audio description, and sign language*, vol. 30. Rodopi, 2007.

[2] United Nations, "Convention on the Rights of Persons with Disabilities.," *Eur. J. Health Law*, vol. 14, no. 3, pp. 281–98, 2007.

[3] W. B. WHO, "The world report on disability," *Disabil. Soc.*, vol. 26, no. 5, pp. 655–658, 2011.

[4] J. Gao, Q. Zhao, T. Li, and Y. Yan, "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)*, 2009, vol. 5553 LNCS, no. PART 3, pp. 576–585.

[5] BOE, "Ley 7/2010 general de la comunicación audiovisual," Madrid, 2010.

[6] CESyA, "Informe de seguimiento del subtitulado y la audiodescripción en la TDT 2014," Madrid, 2015.

[7] S. Renals, M. N. Simpson, P. J. Bell, and J. Barrett, "Just-in-time prepared captioning for live transmissions," in *IBC 2016 Conference*, 2016, p. 27 (9 .)-27 (9 .).

[8] D. Yu and L. Deng, *Automatic speech recognition: A deep learning approach*. London: Springer-Verlag London, 2014.

[9] B. Hinton, G., Deng, L., Yu, D., Dahl, G., Mohamed, A., Jaitly, N., … Kingsbury, "Deep Neural Networks for Acoustic Modeling in Speech Recognition.," *IEEE Signal Processing Magazine*, no. November, pp. 82–97, 2012.

[10] J. Li, L. Deng, R. Haeb-Umbach, and Y. Gong, "Fundamentals of speech recognition," in *Robust Automatic Speech Recognition*, 2015, pp. 9–40.

[11] C. D. Wunsch and S. B. Needleman, "A general method applicable to the search for similarities in the amino acid sequence of two proteins," *J. Mol. Biol.*, vol. 48, no. 3, pp. 443–53, 1970.

[12] W. H.Gomaa and A. A. Fahmy, "A Survey of Text Similarity Approaches," *Int. J. Comput. Appl.*, vol. 68, no. 13, pp. 13–18, 2013.

[13] R. Yu, Z. Luo, and Y. Y. Chiang, "Recognizing text in historical

maps using maps from multiple time periods," in *Proceedings - International Conference on Pattern Recognition*, 2017, pp. 3993–3998.

[14] S. Yang *et al.*, "Duration-aware alignment of process traces," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2016, vol. 9728, pp. 379–393.

[15] A. Ando *et al.*, "Simultaneous subtitling system for broadcast news programs with a speech recognizer," *IEICE Trans. Inf. Syst.*, vol. E86–D, no. 1, pp. 15–25, 2003.

[16] J. E. Garcia, A. Ortega, E. Lleida, T. Lozano, E. Bernues, and D. Sanchez, "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*, 2009.

[17] N. Lertwongkhanakool, P. Punyabukkana, and A. Suchato, "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*, 2013.

[18] M. J. F. Gales, "Adaptive training for robust ASR," in *2001 IEEE Workshop on Automatic Speech Recognition and Understanding, ASRU 2001 - Conference Proceedings*, 2001, pp. 15–20.

[19] A. Maas, Q. Le, T. O'neil, O. Vinyals, P. Nguyen, and A. Ng, "Recurrent neural networks for noise reduction in robust ASR," 2012.

[20] X. Che, S. Luo, H. Yang, and C. Meinel, "Automatic Lecture Subtitle Generation and How It Helps," in *Proceedings - IEEE 17th International Conference on Advanced Learning Technologies, ICALT 2017*, 2017, pp. 34–38.

[21] W. Xuemei, "The Study of Subtitle Translation Based on Multi-Hierarchy Semantic Segmentation and Extraction in Digital Video," *Humanit. Soc. Sci.*, vol. 5, no. 2, p. 91, 2017.

[22] M. Montagud, F. Boronat, J. González, and J. Pastor, "Web-based Platform for Subtitles Customization and Synchronization in Multi-Screen Scenarios," in *dl.acm.org*, 2017, pp. 81–82.

[23] S. Homma, A. Kobayashi, T. Oku, S. Sato, T. Imai, and T. Takagi, "New real-time closed-captioning system for japanese broadcast news programs," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2008, vol. 5105 LNCS, pp. 651–654.

[24] H.-L. Lou, "The viterbi algorithm," *Proc. IEEE*, no. September, 1995.

[25] D. Kedacic, M. Herceg, V. Pekovic, and V. Mihic, "Application for Testing of Video and Subtitle Synchronization," in *2018 International Conference on Smart Systems and Technologies (SST)*, 2018, pp. 23–27.

[26] M. O. van Deventer, H. Stokking, M. Hammond, J. Le Feuvre, and P. Cesar, "Standards for multi-stream and multi-device media synchronization," *IEEE Commun. Mag.*, vol. 54, no. 3, pp. 16–21, Mar. 2016.

[27] I. Cuzco-Calle, P. Ingavelez-Guerra, V. Robles-Bykbaev, and D. Calle-Lopez, "An interactive system to automatically generate video summaries and perform subtitles synchronization for persons with hearing loss," in *2018 IEEE XXV International Conference on Electronics, Electrical Engineering and Computing (INTERCON)*, 2018, pp. 1–4.

[28] A. Rodriguez-Alsina *et al.*, "Subtitle Synchronization across Multiple Screens and Devices," *Sensors*, vol. 12, no. 7, pp. 8710–8731, Jun. 2012.

[29] C.-W. Huang, W. Hsu, and S.-F. Chang, "Automatic Closed Caption Alignment Based on Speech Recognition Transcripts," 2003.

[30] A. Hannun *et al.*, "Deep Speech: Scaling up end-to-end speech recognition," *arxiv.org*, 2014.

[31] G. E. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *IEEE Trans. Audio, Speech Lang. Process.*, 2012.

[32] H. Lee, P. Pham, Y. Largman, and A. Ng, "Unsupervised feature learning for audio classification using convolutional deep belief networks.," in *NIPS, Proc. of Advances in Neural Information Processing Systems*, 2009, pp. 1–9.

[33] P. Zelasko, P. Szymanski, J. Mizgajski, A. Szymczak, Y. Carmiel, and N. Dehak, "Punctuation prediction model for conversational speech," in *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, 2018, vol. 2018–Septe, pp. 2633–2637.

[34] J. Wilkerson and A. Casas, "Large-Scale Computerized Text Analysis in Political Science: Opportunities and Challenges," *Annu. Rev. Polit. Sci.*, vol. 20, no. 1, pp. 529–544, May 2017.

[35] K. Arne, F. Stegen, and T. Baumann, "Mining the Spoken Wikipedia for Speech Data and Beyond," in *LREC 2016*, 2016, pp. 4644–4647.

[36] D. S. Hirschberg, "A linear space algorithm for computing maximal common subsequences," *Commun. ACM*, vol. 18, no. 6, pp. 341–343, 2002.

[37] G. Bordel, M. Penagarikano, L. J. Rodríguez-Fuentes, A. Álvarez, and A. Varona, "Probabilistic kernels for improved text-to-speech alignment in long audio tracks," *IEEE Signal Process. Lett.*, vol. 23, no. 1, pp. 126–129, Jan. 2016.

[38] I. A. Karpukhin and A. S. Konushin, "Constructing a speech audio–video corpus by aligning long segments of speech and text," *Moscow Univ. Comput. Math. Cybern.*, vol. 41, no. 2, pp. 97–103, 2017.

[39] Y. Zhang, Z. Tang, C. Zhang, J. Liu, and H. Lu, "Automatic face annotation in TV series by video/script alignment," *Neurocomputing*, vol. 152, pp. 316–321, Mar. 2015.

[40] AENOR, "UNE 153010:2012. Subtitulado para personas sordas y personas con discapacidad auditiva," 2012.

[41] J. Neves, "Interlingual subtitling for the deaf and hard-of-hearing," in *Audiovisual Translation: Language Transfer on Screen*, 2008, pp. 151–169.

[42] M. De Castro, D. Carrero, L. Puente, and B. Ruiz, "Real-time subtitle synchronization in live television programs," in *IEEE International Symposium on Broadband Multimedia Systems and Broadcasting, BMSB 2011 - Conference Programme*, 2011.