2016

# AUTOMATIC RECOVERY AND VERIFICATION OF SUBTITLES FOR LARGE COLLECTIONS OF VIDEO CLIPS

Mike Armstrong

BBC Research & Development, UK

## ABSTRACT

This paper describes an experimental system that can create good quality subtitle files for video clips derived from broadcast content. The system is designed to run automatically without the need for human verification. The approach utilises existing metadata sources, an off-air broadcast archive and an archive of original subtitle files along with audio fingerprinting and speech-to-text technology to identify the source programme. It then locates the position of the video clip, verifies the match between the video clip and the subtitles and create a new subtitle file.

This paper also reports on the results of the work using a large corpus of over 7,000 video clips and further, smaller sets of clips from different television genres, and explores where improvements might be made. It also looks at the limitations of the current approach discussing alternative methods for providing subtitles for video clips.

## INTRODUCTION

In the UK the BBC provides subtitles for 100% of its television programmes on all of its main television channels. Subtitles are also provided on all these same programmes when on the BBC's video on demand service, iPlayer. However, the situation is very different for video clips that the BBC provides on its websites. At present, a small proportion of clips are subtitled manually, but the majority do not have subtitles. As many websites become more reliant on video content the need to provide subtitles for these clips is increasing.

Our understanding of the use of subtitles is also improving. Audience surveys have indicated that around 10% of our adult TV audience use subtitles daily and around 6% use them most of the time (1) but we have no data on television viewing with subtitles by children. However, with iPlayer the BBC can begin to record accurate data on the use of subtitles on a per programme basis. Currently, verified data is only available for iOS devices, but early indications reveal high levels of subtitle use. A sample from a week in March 2016 indicated that overall subtitle use is around 18%, while usage levels on tablets is higher at over 20%. However, the most interesting figures are for the BBC's children's services where subtitle usage is around 30% and content classified as 'Learning' where use is around 35%. Further work is on going to understand these patterns of subtitle use for on-demand content and to look across other platforms. However, it is clear that for our on-line audience, subtitles are an important part of their viewing experience and this helps to motivate investigatory work to try and extend subtitle availability on-line.

## BACKGROUND

The BBC has many thousands of video clips on its websites and the number is growing every day. Until now, finding subtitles for video clips has been a manual process, either by retrieving subtitles from the original programme or by authoring new ones. However, most video clips provided on the BBC's website are either derived from, or closely related to a broadcast programme. Where a clip is taken directly from a broadcast television programme, subtitles should exist to cover the duration of the clip. If it is possible to locate the broadcast programme and the associated subtitle file and then to identify the timing of the clip, then a new subtitle file can be created for the clip.

At NAB last year we described our initial work, focusing on providing subtitles for clips on the BBC News website (2). Because most news programming is broadcast live, the work included a user interface to enable manual correction and retiming of the subtitles. Towards the end of last year we were asked to look at video clips from general programmes, where the subtitles are mostly pre-prepared and so of high quality. The request was to provide a subtitle search that could be triggered by a video clip being uploaded though the BBC's iBroadcast publication interface and to provide subtitles without the need for an additional user interface. Also required was a batch processing version which could find subtitles for clips already published. In both cases the search needed to be automatic and the quality of the subtitles had to be verified without the need for human oversight.

Initial test data for this work was supplied by BBC Knowledge & Learning who provided a list of all the video clips and programme identifiers (PIDs) from their Bitesize website, `http://www.bbc.co.uk/education`, a learning resource for school children. From this list of PIDs, a total of 7,509 audio/video files were downloaded, amounting to nearly 500 hours of widely varying video content. Viewing all these clips in normal working hours would take one person over 3 months, so this large set of files provided a clear proof-of-concept challenge in terms of subtitling without human intervention.

## THE OVERALL PROCESS

The task breaks down into three basic components: identifying the programme (or several versions of the programme) from which the content could have been derived; locating the section of the programme from which the clip was lifted; and verifying that the subtitles are a good match to the speech across the length of the clip. In order to increase the chance of the system being able to run without any human intervention it was agreed that we would initially limit the scope of the work to 'straight lift' clips where no editing had taken place after the clip had been taken from its original programme.

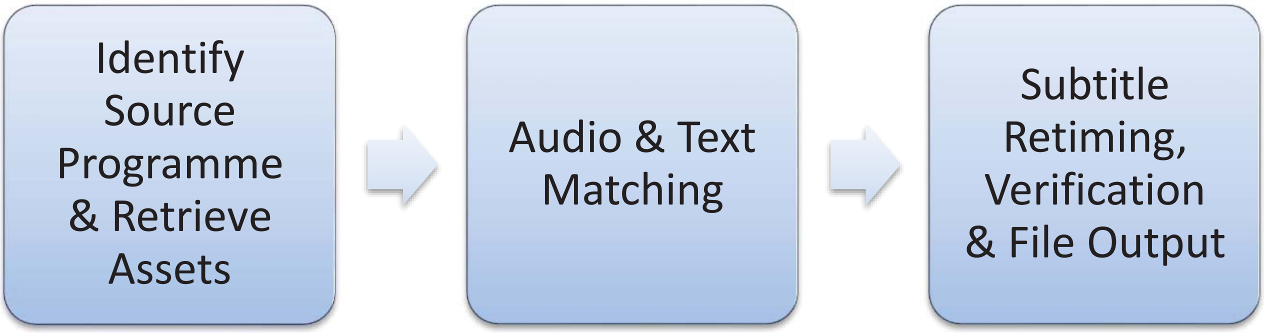
Identify Source Programme & Retrieve Assets
Audio & Text Matching
Subtitle Retiming, Verification & File Output

Figure 1 Subtitle Recovery Process Outline

## IDENTIFYING THE SOURCE PROGRAMME

Two approaches are used to identify the parent programme for video clips. The direct method, which uses metadata to identify the parent programme and to locate it along with its subtitles. And a more indirect method, which is required where metadata is not available or is broken. This involves processing the clip with a speech-to-text system and using the output text strings to search an archive of subtitle files.

### Metadata sources

The BBC's metadata system is known as PIPs[1] (Programme Information Platform). This holds the PID values for all items and their associated metadata and is used to build a number of different systems and services containing metadata. The subtitle search utilises the BBC's programmes service as the initial metadata source. Each clip can be viewed in its unique webpage, so the clip entitled *Dive into a black hole* has a PID value of `p01bybb7` and can be viewed at the URL `http://www.bbc.co.uk/programmes/p00bjs5b`. Data about the clip can be obtained in other formats by adding a file extension to the URL. The .xml and .json versions both provide structured data about the clip and its origins. The key items of data for this task are the PID value for the "parent" programme "episode" and the "series" or "brand" title. In the case of the clip `p01bybb7`, the parent episode *Swallowed by a Black Hole* has PID value `b036bv0z` and brand title *Horizon*. Other useful information is available, such as their durations and the first broadcast date.

### Off-air Archive

The PID value for the episode is used to locate a copy of the broadcast episode in the BBC's off-air archive, BBC Redux[2]. This is achieved via a search engine built on top of Redux called BBC Snippets[3]. Snippets can locate programme items via a text search of the subtitles, or by using the episode PID. The search returns a list of recordings with their Redux disk references and other information, including a flag which identifies live broadcasts. The subtitle search selects the first broadcast of the programme, unless it was live, in which case a repeat is selected, as that will have pre-recorded subtitles. The disk reference is then used to retrieve a copy of the broadcast audio and subtitles from Redux.

[1] http://www.bbc.co.uk/blogs/bbcinternet/2009/02/what_is_pips.html
[2] http://www.bbc.co.uk/blogs/bbcinternet/2008/10/history_of_the_bbc_redux_proje.html
[3] http://www.bbc.co.uk/rd/projects/snippets

The subtitles in Redux are available in XML format. However, these have been recovered from the broadcast DVB subtitles using OCR and since both audio and subtitles have been through playout, transmission and off-air reception, they may contain errors and the time alignment is often imperfect. Also, Redux has only been archiving broadcasts since 2007 so programmes broadcast before this date are not available.

### Subtitle Archive

Separate from Redux, BBC R&D also has an internal archive of all pre-recorded subtitle files that have been broadcast by the BBC along with some more recent subtitles created live. This includes many programmes broadcast prior to Redux and the data is not subject to transmission and playout errors. This archive is indexed by what is known as the 'material reference' and this has no direct mapping back to the episode PID. So this archive is searched using the series or brand title and a set of search strings created from speech-to-text processing of the clip.

### Creating Search Strings

The audio for the clip is processed using an open source speech-to-text engine with an English language data set. This returns data containing each word along with its start and end times, punctuation and a confidence rating. Strings of words become far less common as more words are added, and strings of 5 or more words (n-grams) will usually produce single matches from a large language corpus (3). Search strings are derived from the speech-to-text output containing 4 to 7 consecutive words that have a confidence rating of 100%. The subtitle archive is searched with each search string in turn, along with the programme title. The best match is determined by counting the number of matches for each string. Fractional values are assigned where multiple matches occur.

Problems can arise when the programme brand title is not available to restrict the search. Multiple matches occur with programmes containing poetry and commonly reworked plays, particularly Shakespeare. Here a good text match may be found in the wrong programme, and while the subtitles may contain the correct words, their timings will be wrong. So if no title is available a higher threshold is used to eliminate incorrect matches. If no match is found then the system defaults to using the off-air subtitle file from Redux.

## SEARCHING WITHIN THE PROGRAMME

Two approaches are used to locate the section of the programme that matches the clip. The first is an audio match using an audio fingerprinting technique building on our earlier work described in (2). The second is a set of of text-matching techniques. Both can identify whether a clip is: derived from the programme as a straight lift, whether it appears to have been edited or whether there is clearly no match. Edit detection using the text match, however, is far less reliable than with the audio match.

### Audio Matching

The system makes use of an open source audio fingerprinting tool 'Chromaprint', via its command line tool `fpcalc` (4). This produces 8 overlapping fingerprints per second, each covering 2 seconds of audio which enables an alignment with an error of less than 0.15s. Fingerprints are created for the programme and clip audio, and the clip is then compared with the programme at each location using a cross-correlation algorithm. Because the clip audio may fade in and out, the first and last 2.4 seconds are omitted from the clip fingerprint. The clip fingerprint is also divided into sections, each around 12 seconds in length, to enable the detection of edits; this ensures that the error value produced is more consistent. The comparison produces a list of matching locations for each of the sections and the corresponding error value for the match. If all the locations in the list are within one or two samples of each other and the error values are low then an unedited match is considered to have been found. If however, the list contains two or more different locations with low error values then the clip is considered to have been edited. If the error values are all high then no match has been found.

This audio match can be used to extract the subtitles directly from a programme if the timing of the audio and subtitles are a good match. However in practice, the timing of the subtitle file in our off-air archive has been found to be inconsistent with errors of several seconds in places, so further work is needed to check the timing. Also, because the original subtitle file has a completely different time reference (programme time code) from the off-air file, a further step is needed to accurately locate the subtitles for the clip.

### Initial Timing

A successful text search will match subtitles to one or more of the search strings. These results can be used to calculate a timing offset between the words in the transcript and the matching words in the subtitles. The first and last words of a subtitle relate directly to its start and end times, so the different timings for these words are used to estimate the position of the clip in the subtitle file. However, subtitle timings are often timed to coincide with other factors, such as shot boundaries, so this timing is only approximate.

## SUBTITLE RETIMING

The final stage in the match attempts to correlate the speech-to-text transcript with the subtitle files retrieved. In practice, there will often be large differences between the subtitles because the speech-to-text transcript will contain errors and omissions, particularly where there is background noise beneath the speech or the speaker has an unusual accent. A robust approach is needed to avoid false matches and to recover accurate timings for the subtitles while ensuring that edits and mismatches are detected. This is particularly important if no audio comparison was possible.

Text matching is also more difficult in the case of older programmes, particularly children's programmes, because early subtitling guidelines instructed subtitlers to omit or substitute words, and even rewrite whole sentences to shorten or simplify the language (5). In recent years it has become more usual for subtitles to be provided verbatim.

### First Text Timing Comparison

The first comparison is performed using a sequential maximum length string match, looking for the longest string of words that occur in both the subtitles and in the speech-to-text transcript. The match uses the clip transcript and the section of the programme subtitles identified as matching, with a buffer at the start and end to allow for errors in the timing estimate. This approach can create false matches, but only with short strings, so the main body of matches should be correct. In practice the system begins at an upper length of 60 words and works downwards until a match is found. It then divides the subtitles and transcript at that point and repeats the test recursively until all possible matches are found.

Once a match is achieved, the timing of the words in the transcript can be used to create new timings for each of the subtitles where a sufficient number of words are matched. If a sufficient proportion of subtitles can be retimed, then a new timing offset can be calculated for the clip.

### Second Text Timing Comparison

A second comparison is now carried out in order to verify that the correct subtitles have been located and that the clip has not been edited. This time the transcript text is matched to only the section of subtitles identified by the new timing. A different matching algorithm is used, this time based on unique matches. This avoids false matches and prevents words being given incorrect timings. Where possible, new timings are created for subtitles and these are checked to see if they are consistent with a single match, or whether they indicate that the clip has been edited. Since edits are most often found at the beginning or end of the clip, a further check is made on whether any matching words have been found for the first and last subtitles. The text match is capable of aligning the subtitles to within ±0.2 of a second depending on the consistency of the original subtitles and the number of subtitles matched.

## VERIFICATION

A number of factors are taken into account to verify that the subtitles identified are a good match to the clip. Thresholds for acceptance were arrived at by trial and error examining clips that were borderline and adjusting until the mismatched clips were eliminated. If a good audio match has been found without edits then the subtitles are accepted if more than 30% of the words have been matched in the final comparison and timings have been recovered for over 20% of the subtitles. If no audio was available then the subtitles are accepted if over 40% of the words have matched and timings have been found for over 40% of the subtitles. Also for a text-only match, the timings for all the subtitles must be consistent and the first and last subtitles need to contain matched words. The system will reject some good matches where the speech-to-text produced poor results, but this ensures that those clips that are matched to subtitles have sets of subtitles which are fit-for-purpose and as good as the subtitles broadcast with the original programme. A small number of subtitle files will contain additional words where a clip starts or ends part-way through a subtitle, and very occasionally an additional subtitle, but this is preferable to missing subtitles.

### Output

The final stage is to write a new subtitle file to accompany the clip. For convenience during testing an SRT file is written to enable the results to be previewed using the VLC video player. An XML Timed Text Mark-up Language version is then written for compatibility with the BBC video players. The ambition is to move to writing files that match the BBC's subtitle guidelines (6). The system also writes a log file with data about each clip and details of the match for later analysis.

## RESULTS

Of the 7,509 clips from the Bitesize corpus, the system can currently create subtitles for 3,508 of these clips, a 46.7% success rate. Of these 36.1% were subtitled using audio and text matching and 10.6% were subtitled using text matching only as no audio was available. Of those that failed to match, 27.2% had no audio available and the text match failed, 19.0% failed the audio match and in 7.1% of cases a successful audio match was found, but the speech-to-text failed. The majority of these clips contained non-English speech or had no speech content.

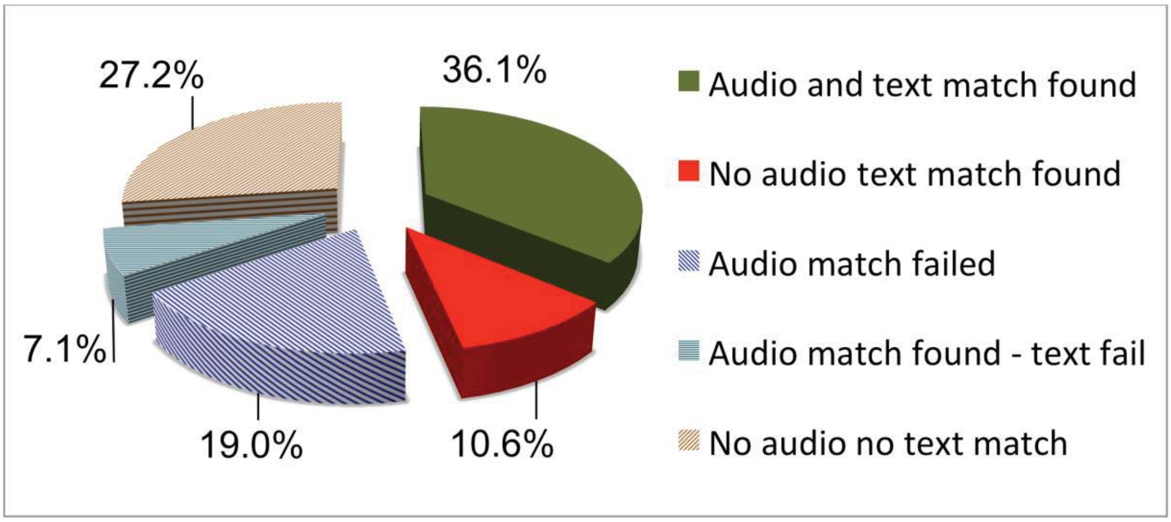

Figure 2 Results of subtitle recovery for the Bitesize corpus.

In order to see how this system could perform with mainstream content, it was also run against sets of clips from a number of television series - see results in Table 1. The large differences in the results reflect the different types of clips created and the different types of content. Where a high level of success is achieved, most clips were lifted from the programmes with no additional production. Where the match failed the clip was usually a trailer or an edited clip. By contrast, programmes like *Doctor Who* have highly produced clips, many made specially for the web and many of the clips for *Later... with Jools Holland* are interviews that were not included in the broadcast or pieces of music.

| Programme | Type | Clips processed | Clips subtitled | Success rate |
|---|---|---|---|---|
| Horizon | Science | 347 | 211 | 68% |
| Timeshift | Archive history | 181 | 137 | 76% |
| Coast | Geography | 149 | 89 | 60% |
| Great British Bake Off | Cookery competition | 181 | 136 | 75% |
| Question Time | Political debate | 33 | 23 | 70% |
| Have I Got News for You | Topical comedy quiz | 85 | 30 | 35% |
| Doctor Who | Sci-Fi | 282 (sample) | 32 | ~11% |
| Later... with Jools Holland | Live music | 220 | 5 | 2% |

Table 1 - Results of subtitle matching across a number of BBC TV programme brands

## DISCUSSION

The approach taken in this work, using metadata recovery and archived content, was taken because it requires almost no change to current production workflows and could be implemented with minimum changes to installed systems and services. There is scope for increasing the yield of subtitled clips by detecting simple edits and combining the subtitles from each section. It is also possible to re-time and re-use live subtitles if the inherent errors are deemed acceptable. Direct access to data from the BBC's main metadata and content systems would also increase the range of programme metadata and content that could be used in the search process. Timing could be improved by the use of shot detection because subtitles often respect shot boundaries. However, the reliance on speech-to-text conversion to verify a match, limits the ability to verify difficult audio. Further work is required to explore the practicalities and risks of deploying an automated process to provide subtitles as part of a live content system.

While this approach will never be able to produce subtitles for 100% of video clips, it could form a useful part of a wider ecosystem for generating subtitles for clips in a cost-effective manner. Other approaches might include capturing metadata at the video edit where the clips are created or passing subtitles through the editing process as well as automated script-to-subtitle conversion alongside the traditional manual creation of subtitles. Some of the techniques discussed in this paper could also be used in the development of an automated quality control of subtitles and audio description.

## CONCLUSIONS

This paper has outlined an automated archival search approach to providing subtitles for video clips that works well for straight-lift clips where speech-to-text conversion can recover sufficient information to verify the results. Excluding news and sport content, early indications are that this approach has the potential to provide subtitles for at least ⅓ and possibly up to ½ of the video clips currently available through the BBC's programmes website. It is one of a number of potential approaches to providing subtitles for video clips.

This work has also shown how combining multiple approaches to a media processing task can improve performance and reduce the need for human intervention, with the potential to increase the reach of a service and limit costs. It also demonstrates how value can be recovered from archived content that has well structured metadata, and how subtitle files provide a very effective tool for media searching.

## REFERENCES

1. Armstrong, M., et al 2015. Understanding the Diverse Needs of Subtitle Users in a Rapidly Evolving Media Landscape. IBC 2015 Conference, 2015

2. Hughes, C.J. & Armstrong, M., 2015. Automatic Retrieval of Closed Captions for Web Clips from Broadcast TV Content. NAB Broadcast Engineering Conference Proceedings. April 2015. pp. 318-324. http://www.bbc.co.uk/rd/publications/whitepaper293

3. Jurafsky, D & Martin, JH, 2009. "Speech and Language Processing", Second Edition, Pearson Education International.

4. Yan, K, Derek, H, and Sukthankar, H, 2005. “Computer Vision for Music Identification”, Computer Vision and Pattern Recognition, IEEE Computer Society, 1 (1).

5. Independent Television Commission, 1999. ITC Guidance on Standards for Subtitling.

6. BBC Subtitle Guidelines, Version 1.1.1, 2016. http://bbc.github.io/subtitle-guidelines/

## ACKNOWLEDGEMENTS

This work has been undertaken at BBC Research and Development with the help and support of Simon Smith & Richard Kurzik from BBC Television, Nigel Megitt, Eyal Lavi, Gareth Ford Williams, Andrew McParland, Rachael McLoughlin, Dong Liu & Robert Lee from BBC Design & Engineering and Edward Teather from BBC Policy. It has been based on previous work carried out at BBC R&D with Chris Hughes who now works at the University of Salford.