Enhancing Accessibility of Parliamentary Video Streams: AI-Based Automatic Indexing Using Verbatim Reports

Daniele Bertillo*[1], Andrea de Donato[1], Carlo Marchetti[2] and Paolo Merialdo[1]

[1]*Roma Tre University, Italy*
[2]*Senato della Repubblica Italiana*

Abstract
The increasing availability of documents and multimedia contents published by public Institutions and Administrations over the Internet pushes investments for improving their accessibility and navigability. The Italian Senate has been broadcasting video streams of its plenary sittings for the last two decades, but only since 2016 each video has been indexed according to the table of contents of the corresponding verbatim report, allowing citizens for accessing videos at the moment of each specific event indexed in the report. However, the elaboration of the augmented indexes necessary for achieving this kind of access requires a considerable effort. In this paper, we present a prototype system that automatizes the production of augmented video indexes for the plenary sittings not currently indexed. We exploit artificial intelligence technologies, such as Speaker Diarization and Speech2Text models, to transcript each sitting and cross-reference the results with sentences in the verbatim reports to create meaningful indexing files, named Video Table of Contents (VTOC). We evaluated our system against sittings of the 15th Italian term obtaining encouraging results.

Keywords
Video Automatic Indexing, Speech2Text, Semantic Textual Similarity, Speaker Diarization

1. Introduction

Over the last two decades, public Institutions and Administrations have significantly increased interest and investments for improving accessibility and navigability of their documentary and multimedia contents over the Internet. As an example, in the parliamentary context, it is nowadays quite common to provide citizens with the possibility of watching plenary sittings as video streams cast over the Internet. However, a single sitting can last several hours, making it difficult for users to address their informative needs, e.g. finding a specific statement of a specific Member of Parliament (MP). Hence, one of the efforts typically put in place by parliamentary Administrations is not only aimed at producing detailed verbatim reports of each debate, but

LIRAI 2023: First Workshop on Legal Information Retrieval meets Artificial Intelligence, September 4–8, 2023, Rome, Italy
*Corresponding author.
All the authors contributed equally.
daniele.bertillo@uniroma3.it (D. Bertillo*); and.dedonato@stud.uniroma3.it (A. de Donato); carlo.marchetti@senato.it (C. Marchetti); paolo.merialdo@uniroma3.it (P. Merialdo)
https://github.com/dedo99/Automatic_indexing_videoStreams_plenarySittings_ItalianSenate (A. de Donato)
0000-0002-3852-8092 (P. Merialdo)

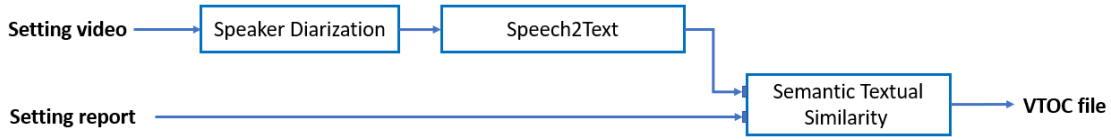

Figure 1: Architecture of the system developed.

also to produce tools and interfaces to facilitate the fruition of these streams, which, in turn, requires a considerable effort both in terms of qualified personnel and time.

The Italian Senate broadcasts plenary sittings over the Internet through a specific website (webtv.senato.it) since 2004, and therefore this site allows accessing an archive composed by thousands of recordings of the Senate plenary sittings, organized by parliamentary terms. Moreover, since April 2016, each video of each plenary sitting is indexed according to the table of contents of the corresponding verbatim report, i.e., a few working days after a sitting has ended, users can watch to the sitting stream while reading the corresponding report, and, vice-versa, they can select a specific item of the table of contents of the report and have the video stream automatically ”jump” to the specific corresponding moment of the sitting. This feature is achieved, in summary, by augmenting the XML index of the report of each sitting with a video-offset attribute for each element of the index; the video-offset attribute is properly set according to time elapsed since the beginning of the video for each event reported in the index.

In this paper we present a prototype system to automatize the production of these augmented XML indexes, by applying state of the art artificial intelligence technologies to the video streams and verbatim reports produced and managed by the Italian Senate for each sitting, with the objective of creating augmented indexes for the plenary sittings not currently indexed. i.e. those in the range 2004-2016. Indeed, the availability of verbatim reports (or, in British gergo, hansards) makes it possible to creatively integrate the results of current open-source artificial intelligence solutions and to obtain good-quality indexing files.

The indexing files used by the Senate of the Italian Republic are named *Video Table of Contents* (VTOC). In order to generate VTOC files we first use Speaker Diarization models to obtain a chronological ordered list of speakers' speeches, and then we use those information in conjunction with a Speech2Text model to transcript each sitting. The results of the transcriptions are linked and crossed with information in the verbatim reports to create meaningful indexing files, that allow users to navigate the video content by topic or speaker through an interface developed by the Senate.

In the following sections we present the system architecture, achieved results, related work, and end with conclusions and future work.

2. Architecture

In this section we provide an overview of the system architecture, composed by three main steps. A representation of the architecture is shown at Figure 1. Here a brief introduction of the individual steps:

- Speaker Diarization: performs speech processing and audio analysis to automatically partition an audio recording into segments based on the identity of different anonymous speakers.
- Speech2Text: converts audio content extracted from videos into text form using speech recognition technologies.
- Semantic Textual Similarity: matches the portions of text extracted from the speech with the portions of texts extracted from the report.

System's inputs are the videos and the verbatim reports of the Senate's sittings. Sitting videos are the actual recording of each sitting, while the reports contain the human transcription of the sitting enriched with speakers' and topics' information.

In the following, we will illustrate the detail of each step and a detail examination of VTOC files.

2.1. Speaker Diarization

Speaker Diarization (or diarisation) is the task of dividing an audio stream that contains human speech into segments based on the identity of each speaker [1]. The process is composed by several tasks, for example voice detection, clustering and segmentation.

For the Speaker Diarization task we used *pyannote.audio* [2], a python open-source library based on the popular machine learning framework PyTorch [3]. *pyannote.audio* provides pre-trained models covering a wide range of domains for voice activity detection, speaker change detection, overlapped speech detection, and speaker embedding, offering good open-source qualitative solutions.

The Speaker Diarization model receives as input the AAC file of the Senate sitting and outputs an RTTM file. The AAC file must be converted to a WAV-formatted file since the AAC format is not supported by the model offered by *pyannote.audio*. The RTTM (Rich Transcription Time Marked) file is a text file that contains fields delimited by spaces, where each record refers to a different speech. Each record contains the following 10 fields, some of them are *<NA>* if the model does not output a value for the relative field and they are not relevant for us:

- Type: segment type; it should always be SPEAKER.
- File ID: name's file; registration's name without extension.
- Channel ID: actual channel; it should always be *1*.
- Turn Onset: start of the speech (in seconds counting from the start of the registration).
- Turn Duration: length (in seconds) of the speech.
- Orthography Field: *<NA>*.
- Speaker Type: *<NA>*.

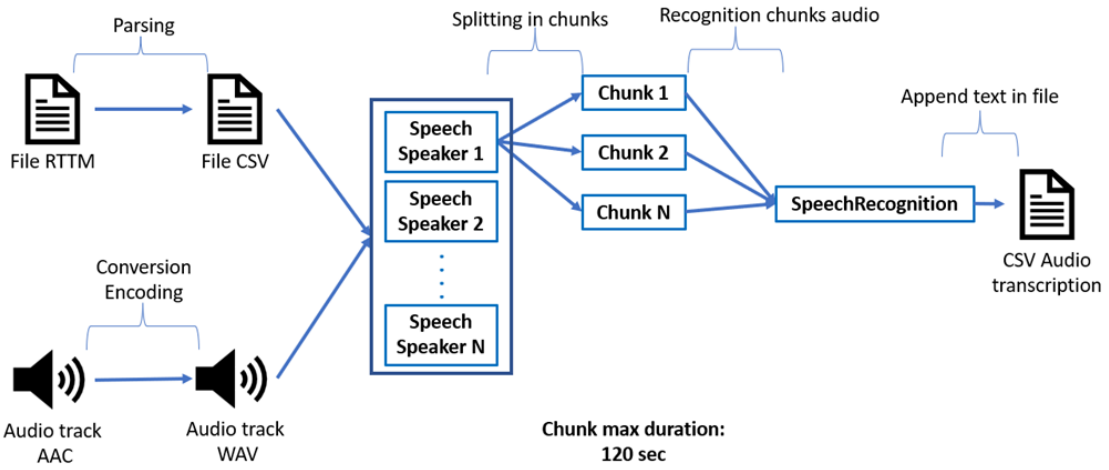

Figure 2: Speech2text architecture.

- Speaker Name: name of the senator; it must be unique in the current file.
- Confidence Score: system's confidence about the information produced; *<NA>*.
- Signal Lookahead Time: *<NA>*.

We use RTTM's fields Turn Onset, *Turn Duration* and *Speaker Name*, which corresponds to speech's offset, speech's length and to the ID associated with the speaker, to produce a CSV file.

2.2. Speech2Text

Speech2Text is the task of transcribing an audio file in a text file. We want to transcript each Senate's sitting as a CSV file with records of each speech. A representation of the Speech2Text task architecture is shown in Figure 2.

For the Speech2Text task we compared the performances of two popular libraries: SpeechRecognition [1] and Whisper [4]. We used Whisper's Medium model and SpeechRecognition *Recognizer Google Web Speech API* [2]. While the performances in terms of word recognition errors are very similar, Whisper took more than twice of the time to process the same speech consisting of about 100 words: Whisper took 3 min 44 sec compared to 1 min 43 sec taken by the SpeechRecognition library. Therefore, we decided to use SpeechRecognition solution.

The Speech2Text receives as input the CSV file generated by the Speaker Diarization step and the audio file of the sitting in AAC format. From the CSV file we extract Turn Onset, Turn Duration and Speaker Name attributes, and then collapse all adjacent records that are related to the same speaker into a single speech by appropriately recalculating the speech lengths of each speaker. This aggregation is needed because the Speaker Diarization model split speeches on pauses rather than on change of speakers' voices. The AAC file is converted to WAV format because the model performing the Speech2Text step does not support the format provided as input (AAC).

[1] https://github.com/Uberi/speech_recognition
[2] https://wicg.github.io/speech-api/

Word transcription is performed separately for each speaker using the data in the CSV file obtained by the Speaker Diarization. For each speaker, the offset of the start of the speech is used to determine the precise time of the speaker debate. Speaker speeches are split into chunks if they are longer than 120 seconds because the model API does not support the processing of overly extended audio tracks. For example, a 260-second speech is divided into 3 chunks, where the first two are 120 seconds while the third has a duration of 20 seconds. After extracting the text for each speaker speech, a punctuation insertion model is used on the extracted text. The model used for this purpose is Deepmultilingualpunctuation [5]. We used this model to enrich the punctuation inserted by the SpeechRecognition library; in fact, the latter tends to insert less punctuation than is needed. A better punctuation improves the results of the semantic similarity step between the sentences extracted from the audio and those extracted from the reports. The output provided by this step is a CSV file consisting of records with four attributes:

- id_speaker: identifier assigned to the speaker during the Speaker Diarization step.
- text: extracted text from the speech.
- offset_video: Start of the speech in seconds from the beginning of the recording.
- duration: length in seconds of the speech.

The output of this step is then used for the Semantic Textual Similarity step to identify speakers' names and speeches' topics.

2.3. Semantic Textual Similarity

Semantic Textual Similarity is the task of comparing two sentences to obtain a similarity score in terms of meaning. We used a Semantic Textual Similarity model to match the sentences extracted from the audio tracks with the sentences extracted from the reports. This step is needed because sitting reports are not the exact transcription of speeches, they are rather revised by Senate employees, and therefore the matching between sentences in transcriptions and in reports can not be determined on the basis of the results of a word-to-word comparison. Hence, we evaluated to what extent a Semantic Textual Similarity solution between sentences of the two sources could be a meaningful way to tackle the problem.

The model used for this step is *SBERT* [6], a variant of *BERT* [7], that generates *sentence embeddings*. Sentence embeddings natural language processing (NLP) techniques that allow the meaning of sentences to be represented as a vector that can be ingested by machine learning models. They capture the meaning and context of a sentence and can be used in tasks such as text classification, sentiment analysis, text generation and sentence similarity.

The Semantic Textual Similarity step receives two files as input: the CSV file obtained from the Speech2Text step containing the text of each speech divided into sentences, and the sitting verbatim report, processed to extract speakers and their speeches as a list of sentences linked to each speaker and topic.

In addition, we defined two parameters to optimize the performances of the semantic similarity process:

- THRESHOLD_SINGLE_SENTENCES: threshold used to determine when two sentences are in match based on similarity score.

- DISPLACEMENT_THRESHOLD: threshold used to determine within what range of distance sentences in the speech can be compared with those in the report; it avoids, for example, comparing sentences spoken at the beginning of the audio track with sentences written at the end of the report.

Sentence comparison is made using the *AllvsAll* approach, in which all embedded speech sentences are compared with the embedded sentences in the report by calculating cosine similarity distances, respecting the constraints imposed by the DISPLACEMENT_THRESHOLD threshold. Consequently, the number of comparisons is reduced according to the value assumed by this threshold.

Next, we associated the IDs identified by the Speaker Diarization step with the name of the speaker. Each speaker ID is mapped to the speaker's name using a dictionary, where the key is the ID and the value is the name (ID -> name). Each key (ID) is associated with the name that has received the highest score so far. In case matches with better scores are encountered it is substituted. For each speech belonging to a speaker we assign the topic that has the highest semantic similarity with it, from the list of topics extracted from the report. In the verbatim reports, topics are the paragraph titles, resulting in an easy extraction.

2.4. VTOC File

Finally, for the generation of the VTOC files of each sitting we used the results of the Semantic Textual Similarity. The VTOC files are XML-like files, which lead us to use the python module xml.etree.ElementTree [3] to generate them. This module allows to parse XML, browse XML, create, edit and write XML elements.

The VTOC file is a marker language-based file, and its main markers are:

- *BGTDOC*
- *WEBTV*
- *Proprieta* (Property)
- *Protocollo* (Protocol)
- *Seduta* (Sitting)
- *InizioSeduta* (Start time of the sitting)
- *Intervento* (Speech)
- *Discussione* (Discussion)
- *Titolo* (Title)

For each marker there are several attributes. The markers that are relevant for us are Intervento (Speech) and Discussione (Discussion), and their attributes are:

- *Discussione* (Discussion):
  - *ID*: discussion's ID.
  - *inIndice*: discussion's topic.

[3] https://docs.python.org/3/library/xml.etree.elementtree.html

  - *VideoOffset*: start timestamp of the discussion.
- *Intervento* (Speech):
  - *Gruppo*: speaker's group affiliation.
  - *idPolitico*: speaker's ID.
  - *inIndice*: speaker's lastname.
  - *Organo*: speaker's political body.
  - *progPers*: speech's ID.
  - *videoOffset*: start timestamp of the speech.

The structure of the VTOC file requires that for each Discussion marker there is at least one Speech marker nested within it. The resulting VTOC file act as the index for the related sitting and is injected in the Senate's system to allow a dynamic navigation of the video content.

3. Evaluation

In this section we present the evaluation methodology and the preliminary results obtained so far. We evaluated the results both the Semantic Textual Similarity and VTOC files, but we decided not to evaluate the transcriptions from the Speech2Text step for two main reasons: in the indexing files speeches' texts are not reported, and the semantic textual similarity mitigates minor mistakes made by the transcription model, hence the impact of such errors is reduced. Evaluating the transcription would have also been a time-consuming activity which was not possible at this stage.

3.1. Evaluation Methods

Results has been evaluated based on three performance metrics: precision, recall and F-measure on both the semantic textual similarity and the generated VTOC files. Precision is defined as the the fraction of relevant instances retrieved (True Positives) on all the instances retrieved (True Positive + False Positives). Recall is defined as the fraction of the relevant instances retrieved (True Positives) on all the actual relevant instances (True Positives + False Negatives). F-Measure is the harmonic mean of precision and recall.

The analysis of correct matches was performed differently based on the task:

- *Semantic Textual Similarity*: since transcription sentences and verbatim report sentences are not in one-to-one correlation, we defined a match successful if two sentences are semantically equal and they belong to the same speaker and speech; sentences not matched even if the corresponding sentence was present were considered false negatives.
- *VTOC files*: we evaluated the speeches identification in the sitting; each speech has been considered individually taking into account their start timestamps, speech's lengths and speaker ID; if one of the latter attributes was incorrect the element of the VTOC file would have been considered a false positive; finally, we considered false negatives the speeches not included in the VTOC at all, which for example can happen when, during a debate, the President gives the floor to a different speaker and its brief notification is not detected by one the previous steps.

Legislation	Sitting	Precision	Recall	F-measure
15	14	0.94	0.96	0.95
15	15	0.84	0.92	0.88
15	18	0.97	0.92	0.94
15	22	0.97	0.96	0.96
15	33	0.94	0.96	0.95

Table 1
Results of the Semantic Textual Similarity by precision, recall and F-measure metrics. Precision and recall are calculated considering whether each pair of sentences, one extracted from the audio track and one extracted from the report, has been matched correctly as semantically similar on the basis of a similarity score.

Legislation	Sitting	Length (min)	Precision	Recall	F-measure
15	14	96	0.79	1	0.88
15	15	110	0.72	0.92	0.81
15	18	44	0.88	1	0.93
15	22	89	0.95	0.72	0.82
15	33	168	0.95	0.89	0.92

Table 2
Results of the generated VTOC files by precision, recall and F-measure. Precision and recall are calculated considering if each speech of the sitting has been reported correctly in the VTOC file as start timestamp, length and speaker ID attributes.

Regarding the thresholds mentioned at 2.3 Section, the optimal values we determined are 0.8 and 0.3 respectively for THRESHOLD_SINGLE_SENTENCES and DISPLACEMENT_THRESHOLD.

3.2. Results

We evaluated the system on five different sittings from the fifteenth legislation on the Semantic Textual Similarity task and VTOC generated files.

Table 1 shows the results of the Semantic Textual Similarity. Overall, the model demonstrates a strong ability to identify semantically similar sentence pairs between the verbatim report and transcription, and appears to be more robust on identifying all the relevant matches.

Table 2 reports the results of the VTOC generated files. The results look very promising, as they show a precision of 0.85 and recall of 0.90, suggesting that the system has the potential to be quite robust. The system seems to be more reliable on the identification of all the relevant speeches. An F-measure of 0.87 suggests good overall performances, but experiments on more sittings are required.

4. Related Work

Some research has been conducted concerning parliamentary sessions transcriptions, indexing and retrieval of information from such transcriptions. *Onyimadu et al.* [8] presents a novel sys-

tem to retrieve information from transcripts using semantic search technologies. [9] expounds upon the indexing standards developed by her team for Canadian parliamentary proceedings. This publication not only addresses the challenges encountered but also offers insights into potential directions for future work. *Szaszak et al.* [10] developed an automatic audio indexing system designed to work in a bilingual environment.

5. Future Work

Future developments present opportunities for improving the performance of all architecture steps. Improving each one of the steps will improve the overall performance of the system, as the steps are intricately linked, and errors or issues in the early steps can propagate across the system, potentially amplifying their impact on the performances.

Regarding the Speech2Text step, it is crucial to continue to develop and train increasingly accurate speech recognition models. In this case we employed a library with an open-source speech recognition engine, but we expect better performance to be achieved using proprietary speech recognition engines. Further analysis and investigation should be dedicated in the accurate evaluation of the transcription results.

Speaker Diarization can benefit from advanced clustering algorithms and deep learning models. The model we used for Speaker Diarization can be improved as it tends to make errors when overlapping voices and noise are present in Senate's sittings.

Finally, the Semantic Textual Similarity step can be improved by advanced text representation models. The evolution of models based on BERT offers opportunities to further refine the semantic comparison between sentences extracted from videos and from the verbatim reports, improving the performance.Also, further tests to evaluate the impact of some decisions we made, for example the thresholds used in the experiments, are required to investigate the actual impact they may have had on results. The use of Large Language Models could also improve comparison results, but at the moment these solutions are limited by the size of the context that can be provided as input to the models.

Addressing the matching problem as a standalone translation problem from the transcription's sentences to verbatim report's sentences, which are slightly different because of the refine work of human operators, is also an interesting research subject, as it would enable the automated transcription of the sittings according to the current Senate's standards.

Overall, this preliminary experiments show encouraging results, but further experimental evaluation is left as future work.

Acknowldgement We express our gratitude to Roberto Battistoni and Giovanni Lalle, domain experts at the Italian Senate, for their assistance in the successful completion of this project.

References

[1] X. Anguera, S. Bozonnet, N. Evans, C. Fredouille, G. Friedland, O. Vinyals, Speaker diarization: A review of recent research, IEEE Transactions on Audio, Speech, and Language Processing 20 (2012) 356–370. doi:10.1109/TASL.2011.2125954.

[2] H. Bredin, R. Yin, J. M. Coria, G. Gelly, P. Korshunov, M. Lavechin, D. Fustes, H. Titeux, W. Bouaziz, M.-P. Gill, pyannote.audio: neural building blocks for speaker diarization, in: ICASSP 2020, IEEE Int. Conference on Acoustics, Speech, and Signal Processing, 2020.

[3] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, S. Chintala, Pytorch: An imperative style, high-performance deep learning library, in: Advances in Neural Information Processing Systems 32, Curran Associates, Inc., 2019, pp. 8024–8035. URL: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

[4] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, I. Sutskever, Robust speech recognition via large-scale weak supervision, 2022. arXiv:2212.04356.

[5] O. Guhr, A.-K. Schumann, F. Bahrmann, H. J. Böhme, Fullstop: Multilingual deep models for punctuation prediction (2021). URL: http://ceur-ws.org/Vol-2957/sepp_paper4.pdf.

[6] N. Reimers, I. Gurevych, Sentence-bert: Sentence embeddings using siamese bert-networks, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, 2019. URL: https://arxiv.org/abs/1908.10084.

[7] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, 2018. URL: http://arxiv.org/abs/1810.04805, cite arxiv:1810.04805Comment: 13 pages.

[8] O. Onyimadu, K. Nakata, Y. Wang, T. Wilson, K. Liu, Entity-based semantic search on conversational transcripts semantic, in: Semantic Technology, Springer Berlin Heidelberg, 2013, pp. 344–349. URL: https://doi.org/10.1007%2F978-3-642-37996-3_27. doi:10.1007/978-3-642-37996-3_27.

[9] S. Bilodeau, The parliament of canada: indexing the work of the senate committees, ????

[10] G. Szaszak, M. Cernak, P. N. Garner, P. Motlicek, A. Nanchen, F. Tarsetti, Automatic speech indexing system of bilingual video parliament interventions, 2013.