V-SAT: Video Subtitle Annotation Tool

Arpita Kundu
LTIMindTree, India
kunduarpita2012@gmail.com

Joyita Chakraborty
LTIMindTree, India
joyita.ckra@gmail.com

Anindita Desarkar
LTIMindTree, India
aninditadesarkar@gmail.com

Aritra Sen
LTIMindTree, India
aritra_sen@outlook.com

Srushti Anil Patil
LTIMindTree, India
srushtipatil7418@gmail.com

Vishwanathan Raman
LTIMindTree, India
vishwanathanraman@gmail.com

Abstract

The surge of audiovisual content on streaming platforms and social media has heightened the demand for accurate and accessible subtitles. However, existing subtitle generation methods—primarily speech-based transcription or OCR-based extraction—suffer from several shortcomings, including poor synchronization, incorrect or harmful text, inconsistent formatting, inappropriate reading speeds, and the inability to adapt to dynamic audio-visual contexts. Current approaches often address isolated issues, leaving post-editing as a labor-intensive and time-consuming process. In this paper, we introduce V-SAT (Video Subtitle Annotation Tool), a unified framework that automatically detects and corrects a wide range of subtitle quality issues. By combining Large Language Models (LLMs), Vision-Language Models (VLMs), Image Processing, and Automatic Speech Recognition (ASR), V-SAT leverages contextual cues from both audio and video. Subtitle quality improved, with the SUBER score reduced from 9.6 to 3.54 after resolving all language mode issues and F1-scores of ~0.80 for image mode issues. Human-in-the-loop validation ensures high-quality results, providing the first comprehensive solution for robust subtitle annotation.

Keywords

Video Subtitle Annotation, Automated Subtitle Quality Detection, Subtitle Correction, Large Language Models (LLMs), Vision-Language Models (VLMs), Automatic Speech Recognition (ASR)

1 Introduction

The amount of audiovisual content has surged dramatically with the rapid growth of online platforms such as YouTube, Amazon Prime, and Netflix [1], etc. Subtitles, which are snippets of timed text displaying spoken content on screen, have become an essential feature accompanying this content. Recent surveys [2] confirm this growing trend: for example, 85% of Netflix users (54% for Amazon Prime and 37% for Disney+) report opting to watch content with captions, and younger viewers under 30 are particularly likely to enable subtitles. Subtitles not only support viewers with hearing impairments but also benefit audiences in noisy environments, with low-quality audio, or when clarity of accents and specialized content is required. This growing demand underscores the importance of high-quality subtitle generation and presentation.

Despite these benefits, current subtitle generation and extraction methods face significant challenges that hinder user experience. Speech-based approaches [9, 14], which rely on transcribing audio, and OCR-based methods [15, 20], which extract text from video frames, often suffer from inconsistencies. Common problems include inaccurate start and end timestamps, poor synchronization with spoken dialogue, and failure to represent non-speaker audio such as background noise, silence, or music. Generated subtitles may also contain incorrect content, including spelling errors, grammatical mistakes, mistranslations, or harmful words. Furthermore, formatting inconsistencies—such as irregular font size, color, or positioning—can reduce readability, sometimes overlapping with important on-screen visuals. Additional issues include inappropriate reading speed, measured by Characters Per Line (CPL) and Characters Per Second (CPS), and the lack of contextual adaptation to dynamic audio-visual content. These limitations collectively increase viewers' cognitive load and disrupt immersion.

Existing literature offers a range of subtitle-generation models—some focus on speech transcription [16], while others use OCR for text extraction. Some works [11, 13, 17] attempt to solve specific problems, such as improving synchronization or optimizing subtitle positioning. Others address linguistic quality by reducing spelling and translation errors. However, these solutions are often narrow in scope, targeting one or two isolated issues rather than approaching the problem holistically [4, 5]. Moreover, current methods rarely integrate contextual information from both audio and video, leading to subtitle texts that fail to dynamically adapt to scene changes. Subtitle post-editing today is still largely manual, involving labor-intensive corrections by human editors, which is both time-consuming and costly [2, 3, 10, 12]. To date, there is no robust solution that simultaneously tackles the broad spectrum of challenges in subtitle generation and quality assurance.

Key Contributions: In this work, we propose V-SAT (Video Subtitle Annotation Tool), a unified system designed to address all major challenges in subtitle generation and post-editing. Given a raw video and subtitle file, V-SAT automatically detects inconsistencies across timing, content accuracy, formatting, positioning, and reading speed. Importantly, it not only flags these issues but also suggests automated corrections, leveraging the combined potential of Large Language Models (LLMs), Vision-Language Models (VLMs), Image Processing (IP) techniques, and Automatic Speech

[1] https://www.insiderintelligence.com/insights/ott-video-streaming-services/
[2] https://idealinsight.co.uk/infographics/netflix-captions

CODS'25, Pune, India

ACM ISBN 978-1-4503-XXXX-X/2018/06
https://doi.org/XXXXXXX.XXXXXXX

Recognition (ASR). By integrating multimodal contextual cues from both audio and video, our tool provides more accurate and adaptive subtitle generation. Finally, V-SAT includes an interactive human-in-the-loop validation step. In addition to automatically detecting and correcting issues, the tool also enables users to manually annotate content even when no issues are detected, thereby broadening the scope of correction. Subtitle quality was evaluated using SUBER for language mode issues and F1-score for image mode issues. The SUBER score improved from 9.6 to 3.54 after corrections, while subtitle positioning and font color issues achieved F1-scores of 0.82 and 0.80. To the best of our knowledge, this is the first comprehensive framework that attempts to jointly solve the wide range of subtitle quality issues in a single automated pipeline.

2 Related Work

Prior works on subtitling mainly falls into two streams– subtitle generation [14] and subtitle extraction [20]. This categorization mainly reflects speech-related studies that derive subtitles from audio, and vision or OCR-related studies that recover on-screen text from video frames—each bringing different strengths and limitations. Below, we organize prior work along these dimensions and highlight targeted studies that address specific subtitle-quality issues (synchronization, positioning, compression), as well as gaps that motivate our approach.

Generation methods treat subtitling as Speech Translation (ST) problem [16], using either cascaded Automatic Speech Recognition + Machine Translation or direct end-to-end models [14, 18], often based on Conformer encoders [7]. While direct models reduce error propagation, practical pipelines still struggle with segmentation, timestamps, and translation [10, 14]. Extraction approaches rely on OCR or vision–language models to recover on-screen text [1, 15, 20], but OCR often loses temporal context and vision-based models face limits in handling bounding boxes and long videos. Several open-source and commercial tools exist such as VideoOCR [3] and PyVideoTrans [4] and other commercial cloud services like OpenVision [5] and GhostCut [6], though with restricted scope.

Beyond subtitle generation and extraction, specific subtitle quality issues remain. These include inaccurate timestamps [11], poor synchronization under noise or multi-speaker settings [1, 12], suboptimal positioning that overlaps visual content [13, 17], and length or readability constraints in prompting-based compression methods [4]. Resources for evaluation are sparse [6, 8], and recent LLM-based customization approaches [2] show mixed results.

In sum, existing methods either (i) focus on subtitle generation or extraction without integrated quality assurance or (ii) address one or two isolated subtitle quality issues. There is no comprehensive system that jointly detects the broad spectrum of subtitle issues (timing, content, format, positioning, reading speed, and contextual adaptation) and proposes corrections while remaining human-in-the-loop. This gap motivates our V-SAT tool, which aims to unify detection and correction across these dimensions using multimodal models and interactive validation.

[3]https://github.com/devmaxxing/videocr-PaddleOCR
[4]https://github.com/jianchang512/pyvideotrans
[5]https://vision.aliyun.com/
[6]https://cn.jollytoday.com/home/

3 Proposed Solution Outline

The proposed solution outline of the V-SAT tool is as follows:

- **Input:** The system takes a raw video file (without subtitles) and a original subtitle file in .srt or .vtt format as input.
- **Issues detection:** V-sat detects and corrects subtitle issues in two modes: language and image. Language mode issues include spelling and grammar errors, harmful words, timing mismatches, non-words, and segmentation problems such as overly long subtitles, etc. Image mode issues involve poor subtitle positioning and inappropriate font color in video frames.
- **Display issues and suggestion:** Detected issues are displayed with suggested corrections, timestamps, and corresponding audio content.
- **Human Validation:** All subtitles, even those without detected issues, are open for human validation. Annotators can accept or override suggested corrections.
- **Output:** Outputs are the final subtitle file similar to the uploaded format and the video embedded with the subtitle.

4 System Overview

In this section, we present an overview of V-SAT tool.

4.1 Pre-processing

In this sub-section, we describe the pre-processing of input files. The system requires two inputs: (a) a raw video file without subtitles and (b) a subtitle file in .srt/.vtt format. The subtitle file is first converted into CSV, and its timestamp information is used as a reference. For each timestamp, the corresponding audio and video clips are extracted separately.

4.2 Feature Components

In this sub-section, we describe the issues addressed by the V-SAT tool in video subtitles and outline the approaches used for their detection and correction.

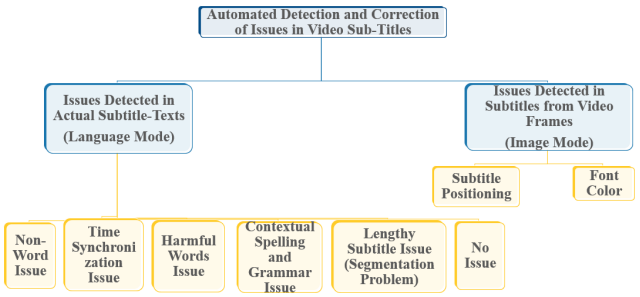

Figure 1: Summary of issues addressed in the V-SAT tool

4.2.1 Language mode issues: In this section, we briefly discuss the issues in actual subtitle text.

- **Issue 1: Contextual spelling and grammar-** This issue arises when a word is spelled correctly but used incorrectly in context. For example, in a cooking video, dessert is contextually correct, whereas desert is not. To detect such cases, we

prompt a Large Language Model (LLM) in zero-shot mode, using the current subtitle text along with its preceding context (subtitle texts). Unlike grammar checking, the goal is to identify contextually inappropriate words. Once identified, corrections are generated by prompting the LLM with a predefined set of rules, and each correction is validated against these rules. Overall, we find that this approach improves both detection and correction accuracy.
  - *Rule 1:* Correct words should NOT add a suffix (like -y, -ness, -ful, -less, -ed etc.) to change the grammatical category of the misspelled words.
  - *Rule 2:* Correct words must be meaningful and contextual.
  - *Rule 3:* Correct words differ only in spelling but fits the context better.
- **Issue 2: Harmful words-** This issue involves filtering offensive or HAP content (hate speech, abusive language, and profanity). V-SAT uses a Large Language Model (LLM) to detect such words, which are then masked with asterisks.
- **Issue 3: Time synchronization-** A time synchronization issue in subtitles occurs when the text on screen does not align with the corresponding audio, appearing too early, too late, or gradually drifting out of sync. In V-SAT, this issue is addressed by first extracting audio clips for each timestamp during pre-processing. Transcripts are then generated using the faster-whisper-medium model. The extracted transcripts are compared with the user-provided subtitle file using cosine similarity. If the similarity score falls below a defined threshold (i.e. 0.7), a time synchronization issue is flagged.
- **Issue 4: Non word-** Non-word issues arise when subtitles include cues such as [music playing] or [silence] to represent background sounds instead of spoken words. In V-SAT, this is addressed by extracting audio clips for each timestamp and applying Google's YamNet model for audio classification. YamNet predicts 521 event classes with associated probabilities, and if a class exceeds a set threshold (0.3), the corresponding event label is added to the subtitle text.
- **Issue 5: Segmentation-** A segmentation issue occurs when subtitles are not properly divided into readable chunks, often due to lengthy text, poor line breaks, or merging multiple sentences. To detect such cases, we calculate the Characters Per Line (CPL); if it exceeds 50, a segmentation issue is flagged. The subtitle text is then split at the 50-character threshold. Since the original timestamps become inaccurate after splitting, the faster-whisper-medium model is used to generate word-level timestamps, and the last word of each split segment of subtitle texts is used to realign the timing.
- **No issue-** Even when no issue is detected, V-SAT provides users with the option to manually annotate or edit subtitles.

4.2.2 Image mode issues: In this section, we briefly discuss the issues in actual subtitle text in video frames.

- **Issue 6: Subtitle positioning-** This issue occurs when subtitles are placed in a way that blocks important visual content or distracts the viewer. By default, subtitles appear at the bottom center, but problems arise when that area already contains text (e.g., speaker names, news tickers, or graphics) or when subtitles cover key visuals, are misaligned, or inconsistently positioned. To detect such issues, video clips are extracted per timestamp, and the first frame is analyzed. Using OpenCV, a saliency map is generated for each frame, and an overlap score is calculated between salient objects and the default subtitle position. If the score exceeds a threshold (0.006), a positioning problem is flagged. To correct it, the x and y coordinates of subtitles are shifted from the bottom center to alternative positions (e.g., middle center), and the final position is chosen where the overlap score is minimal.
- **Issue 7: Subtitle font color-** This issue arises when subtitles are difficult to read due to poor visual design, such as inappropriate font color. Font color issues involve low contrast, distracting choices, or inconsistent usage. To address this, the subtitle background region is extracted from each frame, and its average color is converted to a brightness value. If brightness exceeds 128, the subtitle font is rendered in black; otherwise, it is rendered in white to ensure readability.

4.2.3 Human in the loop: User can validate the output of the tool.

4.3 Demonstration Interface

The frontend UI, built with Streamlit, is organized into tabs. In the first tab, users can upload a raw video and a subtitle file (.srt or .vtt format). Separate tabs handle language and image-related issues in subtitle texts, with dropdowns listing specific issues. For example, see figure 2 for segmentation issue and figure 3 for subtitle positioning issue. The tool detects subtitle issues for each timestamp, provides suggested fixes, and lets users preview audio alongside the original and corrected text. For issues 1 to 4, users can manually edit subtitles; for others, they can accept or reject the suggestions. Users can then download the final subtitle file similar to the uploaded format (.srt/.vtt) and the video embedded with the subtitle. The demonstration video is available at https://youtu.be/zBg6DnKWcuA.

4.4 Tools and Techniques

The V-SAT tool leverages multiple state-of-the-art models and libraries to detect and correct subtitle issues. For language-related issues, it employs the GPT-4o mini model as the Large Language Model (LLM). For video analysis, it uses the OpenCV library to extract and process frames. Automatic Speech Recognition (ASR) is performed using OpenAI's Whisper model, which provides accurate transcripts along with word-level timestamps. For audio event detection, V-SAT integrates Google's YamNet model. Together, these components enable robust multimodal analysis of subtitles across language, audio, and visual dimensions. The system requirements to run this tool includes python 3.10, 16gb ram, and i5 intel processor.

4.5 Experimental Results

Experiments are done on a corpus of ten videos of average 3 minute duration. Both videos and ten transcript files are downloaded from YouTube. The ground truths for these are majority voted annotations of 3 annotators.

To measure the quality of corrected subtitles after resolving each issue, two different metrics are used: the SUBER score [19] for language mode issues and the F1-score for image mode issues. Scores are measured with respect to the ground truths. A lower

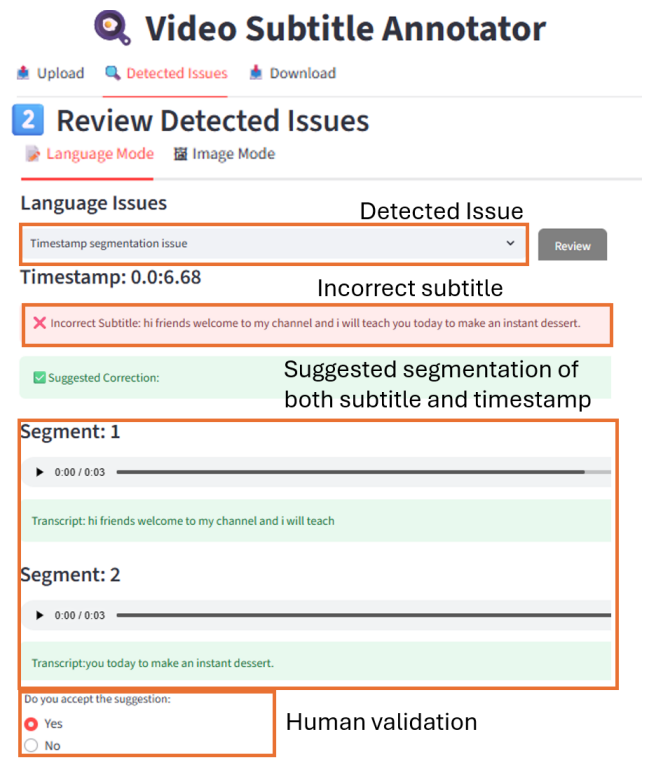

Figure 2: System interface for segmentation issues

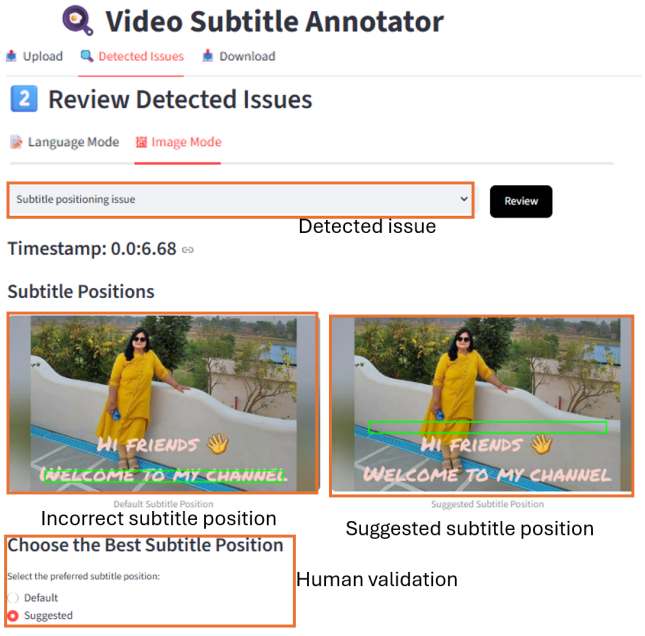

Figure 3: System interface for subtitle positioning issue

SUBER score indicates better alignment with the ground truth, while a higher F1-score reflects better detection accuracy.

System output	SUBER score
Input subtitle file	9.60
Issue1	9.10
Issue_1+2	9.09
Issue_1+2+3	6.57
Issue_1+2+3+4	4.55
Issue_1+2+3+4+5	3.53
All_lang_issue	3.54

(a)

Model	F1-score
Issue 6	0.82
Issue 7	0.80

(b)

Table 1: Performance after resolving issues in (a) language and (b) image mode which are described in section 4.2.1.

As shown in the table 1a, input subtitle file uploaded by user with all issues had an initial SUBER score of 9.6. After gradually resolving all language-mode issues, this score was significantly reduced to 3.54. Furthermore, table 1b illustrates for issues 6 and 7, we obtained F1-scores of 0.82 and 0.80, respectively.

5 Potential Applications

This work has several applications. It includes real-time subtitle personalization [2, 5], where subtitles can be dynamically adapted to user preferences such as reading speed, text size, or display position, thereby enhancing individual viewing experiences [2, 5]. In scenarios involving multiple speakers, the system can support selective subtitling, allowing viewers to focus on specific speakers or narrative threads to optimize video understanding. Further, this may help to explore different aspects of a story, optimizing their video understanding experience. It also enables movie and video subtitle translation, where subtitles can be corrected and adapted across languages while maintaining synchronization and contextual fidelity [20]. Furthermore, integration into object-based media workflows allows subtitles to function as modular components, enabling structured, context-aware adaptation across diverse devices and platforms. Beyond entertainment, the framework can be applied in education, accessibility for hearing-impaired users, live events, and multilingual conferencing, providing scalable and adaptive solutions for high-quality subtitling.

6 Conclusion

In this paper, we introduced V-SAT (Video Subtitle Annotation Tool), a novel and comprehensive framework that integrates LLMs, VLMs, IP, and ASR to automatically detect and correct a wide range of subtitle quality issues within a single platform. Unlike prior approaches that address problems in isolation, V-SAT effectively combines temporal and contextual information from both audio and video to ensure more accurate and adaptive subtitle correction. Each automated correction is followed by an interactive human validation step, enabling a robust human-in-the-loop process that enhances reliability. Furthermore, beyond correcting the predefined issues identified in this work, V-SAT empowers users to manually annotate other problems, ensuring greater flexibility and coverage. Future work includes real-time correction, multilingual support, and integration with streaming platforms.

References

[1] Oliver Alonzo, Hijung Valentina Shin, and Dingzeyu Li. 2022. Beyond subtitles: captioning and visualizing non-speech sounds to improve accessibility of user-generated videos. In *Proceedings of the 24th International ACM SIGACCESS Conference on Computers and Accessibility*. 1–12.

[2] Mariana Arroyo Chavez, Bernard Thompson, Molly Feanny, Kafayat Alabi, Minchan Kim, Lu Ming, Abraham Glasser, Raja Kushalnagar, and Christian Vogler. 2024. Customization of closed captions via large language models. In *International Conference on Computers Helping People with Special Needs*. Springer, 50–58.

[3] Helard Becerra Martinez, Alessandro Ragano, Diptasree Debnath, Asad Ullah, Crisron Rudolf Lucas, Martin Walsh, and Andrew Hines. 2024. Dialogue Understandability: Why are we streaming movies with subtitles? *arXiv e-prints* (2024), arXiv–2403.

[4] Marco Gaido, Sara Papi, Mauro Cettolo, Roldano Cattoni, Andrea Piergentili, Matteo Negri, and Luisa Bentivogli. 2024. Automatic Subtitling and Subtitle Compression: FBK at the IWSLT 2024 Subtitling track. In *Proceedings of the 21st International Conference on Spoken Language Translation (IWSLT 2024)*. 86–96.

[5] Benjamin M Gorman, Michael Crabb, and Michael Armstrong. 2021. Adaptive subtitles: preferences and trade-offs in real-time media adaption. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–11.

[6] Boyu Guan, Yining Zhang, Yang Zhao, and Chengqing Zong. 2025. TriFine: A Large-Scale Dataset of Vision-Audio-Subtitle for Tri-Modal Machine Translation and Benchmark with Fine-Grained Annotated Tags. In *Proceedings of the 31st International Conference on Computational Linguistics*. 8215–8231.

[7] Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, et al. 2020. Conformer: Convolution-augmented transformer for speech recognition. *arXiv preprint arXiv:2005.08100* (2020).

[8] Alina Karakanta, Matteo Negri, and Marco Turchi. 2020. MuST-cinema: a speech-to-subtitles corpus. *arXiv preprint arXiv:2002.10829* (2020).

[9] Alina Karakanta, Sara Papi, Matteo Negri, and Marco Turchi. 2021. Simultaneous speech translation for live subtitling: from delay to display. *arXiv preprint arXiv:2107.08807* (2021).

[10] Maarit Koponen, Umut Sulubacak, Kaisa Vitikainen, and Jörg Tiedemann. 2020. MT for subtitling: User evaluation of post-editing productivity. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*. 115–124.

[11] Jose Manuel Masiello-Ruiz, Belen Ruiz-Mezcua, Paloma Martinez, and Israel Gonzalez-Carrasco. 2023. Synchro-Sub, an adaptive multi-algorithm framework for real-time subtitling synchronisation of multi-type TV programmes. *Computing* 105, 7 (2023), 1467–1495.

[12] Lloyd May, Michael Clemens, Khang Dang, Keita Ohshiro, Sripathi Sridhar, Pauline Wee, Magdalena Fuentes, Sooyeon Lee, and Mark Cartwright. 2025. 'Choices? That's The Dream": Challenges and Opportunities in Non-Speech Information Closed-Captioning. *Frontiers in Computer Science* 7 (2025), 1575176.

[13] Bogdan Mocanu and Ruxandra Tapu. 2021. Automatic subtitle synchronization and positioning system dedicated to deaf and hearing impaired people. *IEEE Access* 9 (2021), 139544–139555.

[14] Sara Papi, Marco Gaido, Alina Karakanta, Mauro Cettolo, Matteo Negri, and Marco Turchi. 2023. Direct speech translation for automatic subtitling. *Transactions of the Association for Computational Linguistics* 11 (2023), 1355–1376.

[15] Aditya Ramani, Asmita Rao, V Vidya, and VR Badri Prasad. 2020. Automatic subtitle generation for videos. In *2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS)*. IEEE, 132–135.

[16] Matthias Sperber and Matthias Paulik. 2020. Speech translation and the end-to-end promise: Taking stock of where we are. *arXiv preprint arXiv:2004.06358* (2020).

[17] Ruxandra Tapu, Bogdan Mocanu, and Titus Zaharia. 2019. DEEP-HEAR: A multimodal subtitle positioning system dedicated to deaf and hearing-impaired people. *IEEE Access* 7 (2019), 88150–88162.

[18] Alex Waibel, Ajay N Jain, Arthur E McNair, Hiroaki Saito, Alexander G Hauptmann, and Joe Tebelskis. 1991. JANUS: a speech-to-speech translation system using connectionist and symbolic processing strategies. In *[Proceedings] ICASSP 91: 1991 International Conference on Acoustics, Speech, and Signal Processing*. IEEE, 793–796.

[19] Patrick Wilken, Panayota Georgakopoulou, and Evgeny Matusov. 2022. SubER: A metric for automatic evaluation of subtitle quality. *arXiv preprint arXiv:2205.05805* (2022).

[20] Haiyang Yu, Jinghui Lu, Yanjie Wang, Yang Li, Han Wang, Can Huang, and Bin Li. 2025. Eve: Towards end-to-end video subtitle extraction with vision-language models. *arXiv preprint arXiv:2503.04058* (2025).