METHODOLOGY | Article in Press

Improving weakly supervised ASR from TV subtitles via refinement and adversarial training

Xinnian Zhao[1] · **Hugo Van Hamme**[1]

Received: 28 November 2024 / Accepted: 25 August 2026
© The Author(s) 2026

Abstract

Automatic speech recognition (ASR) systems typically require large volumes of high-quality transcribed speech. In low-resource settings, where accurate verbatim transcripts are scarce, weakly supervised (WS) sources provide a practical alternative. In this work, we use the audio, subtitle texts, and subtitle timestamps from 16 television programs broadcast by the *Vlaamse Radio- en Televisieomroeporganisatie* (VRT, Flemish Radio and Television Broadcasting Organisation) as WS resources for ASR training. We examine the value of subtitles by comparing them with pseudo transcripts generated by existing ASR models, which are commonly used in WS ASR. Since TV subtitles are often imperfectly aligned with speech, we first propose a subtitle refinement pipeline that uses Whisper-based pseudo transcriptions, forced alignment, and subtitle-specific processing to produce better-aligned training pairs. We then introduce a more flexible end-to-end alternative based on adversarial training, which models text predictions at the distribution level and reduces the reliance on precise text–speech alignment. Experiments with different ASR backbones show that training on refined subtitles substantially improves recognition performance over raw subtitles and direct pseudo-transcript supervision. The adversarial approach achieves competitive overall performance while offering a simpler training framework for more weakly aligned settings.

Keywords Weakly supervised ASR · Subtitles · Refinement · Adversarial training

Abbreviations

ASR	Automatic speech recognition
WS	Weakly supervised
VRT	Vlaamse Radio- en Televisieomroeporganisatie (Flemish Radio and Television Broadcasting Organisation)
CGN	Corpus Gesproken Nederlands (Spoken Dutch Corpus)
CTC	Connectionist temporal classification
AED	Attention-based encoder-decoder
WER	Word error rate
IV-NE	In-vocabulary named entity
OOV-NE	Out-of-vocabulary named entity

This Article in Press is shared early to give you faster access to new research. It is citable and carries a permanent DOI. The final edited version will replace it automatically.

Published online: 09 September 2026

BPE	Byte-pair encoding
MFCC	Mel-frequency cepstral coefficient
CNN	Convolutional neural network

1 Introduction

Since the introduction of deep learning into speech recognition systems, remarkable improvements have been made in recognition performance [1–4]. However, these neural network-based models heavily depend on the quality and quantity of available data. With advances in computational power, the expansion and reinforcement of datasets even show a greater impact on performance than algorithmic refinements [5]. As a result, exploring diverse data sources has become increasingly important, rather than relying solely on manually curated datasets, which are costly and labor-intensive. A notable example of this trend is Whisper [6], a foundational speech processing system that leverages large-scale noisy data from web alongside limited labeled data, scaling the total audio dataset to 680k hours. Despite its competitive performance in zero-shot scenarios [7–9], Whisper still suffers from data bias [10]. For instance, only 0.3% of its dataset consists of Dutch, and within that, an even smaller portion is Flemish (variant of Dutch spoken in Belgium). The challenge of identifying and leveraging weakly supervised data for low-resource languages and domains remains a critical area of research.

Television subtitles offer a promising solution. TV programs closely resemble natural speech in real-world scenarios, encompassing a wide range of domains and acoustic conditions, including background noise, music, spontaneous conversations, language switching, heavy accents, and speech disfluencies. At the same time, TV subtitles provide highly relevant textual information, though they tend to be more concise and polished than verbatim transcripts, omitting repetitions, hesitations, and modifications for better on-screen readability. Several studies have employed transfer learning to train ASR models using subtitles due to their similar objectives [11, 12]. Pretraining ASR models on large-scale subtitle datasets serves as an effective proxy for the ASR task, allowing the models to capture transferable and generalizable knowledge across different domains. After pretraining, the models can be finetuned on a smaller set of verbatim transcripts, optimizing them for accurate transcription. However, the model's generalization ability is constrained by fine-tuning, and ASR performance becomes heavily reliant on the size of the verbatim dataset. Alternatively, a multitask architecture can be utilized to jointly train the ASR and subtitling tasks, enabling knowledge transfer from subtitles to ASR through a shared acoustic encoder [13]. Nevertheless, limited linguistic information can be captured from subtitles when separate decoders are employed for each task.

To address the alignment challenges and effectively utilize subtitled data for ASR training, we propose a preprocessing method that refines subtitle timestamps, aligning them more closely with the spoken audio, thereby making the training data more verbatim-like. Rather than introducing a new transcription or forced-alignment model, our contribution lies in a subtitle-oriented refinement pipeline for weakly supervised ASR training. This approach is supported by Whisper [6] used for generating hypothesis transcripts and Wav2vec2.0 [14] used for forced alignment [15]. Specifically, Whisper provides speech-grounded pseudo-transcriptions, while Wav2vec2.0-based forced alignment provides word-level timing information. The novelty of our method lies in how these off-the-shelf components are integrated with subtitle-specific processing, including timestamp shifting, subtitle-informed prompting, segment filtering, and text matching between pseudo-transcriptions and subtitles, to construct refined subtitle-audio pairs suitable for training. We design a detailed preprocessing pipeline for subtitle refinement, and our results demonstrate significant improvements in recognition accuracy when using the refined data. This success is largely due to the nature of subtitles, which are closely paired with speech and often closely resemble verbatim transcripts. Additionally, we introduce a more flexible method that eliminates the need for organized text and complex refinement processes by utilizing adversarial training. Instead of mapping speech to a paired subtitle sequence, a discriminator makes decisions at the distribution level. This approach achieves

results comparable to the refinement method but offers the advantage of a fully single-stage training process that can handle poorly aligned or unpaired data.

Our contributions can be summarized as follows:

1. We analyze why subtitles are competitive weakly supervised resources for ASR, and discuss their complementary properties with respect to automatically generated pseudo-transcripts.
2. We design a subtitle-oriented preprocessing pipeline that integrates Whisper-based pseudo-transcription and Wav2vec2.0-based forced alignment with subtitle-specific processing to refine subtitle-audio pairs for ASR training.
3. We propose an end-to-end architecture that incorporates adversarial training, enabling ASR systems to be trained directly from subtitled data.

2 Related works

2.1 Data refinement

Data refinement is a practical and direct approach to handling noisy data with weak supervision. Since the first Multi-Genre Broadcast (MGB) Challenge in 2015 [16–19], various works have focused on improving the quality of poorly aligned or noisy transcriptions through data refinement techniques.

Some research efforts have utilized semi-supervised learning to mitigate the effects of noisy data. For example, confidence-based filtering is employed in [20] to refine WS data, allowing ASR models to benefit from both clean and noisy transcripts. Similarly, [21] applied filtering strategies to subtitles by removing irrelevant or erroneous segments. However, these methods rely on the assumption that the model's confidence scores or patterns learned from clean data can be effectively transferred to noisy data, which may not always be reliable.

Another significant line of research focuses on improving the temporal alignment between speech and text in WS data. In [21], for instance, techniques such as voice activity detection and speaker diarization are applied to improve alignment, benefiting from metadata containing speaker information. Likewise, [22] combined forced alignment with pretraining on large-scale out-of-domain data to further refine WS datasets. While this approach proved effective, the availability of advanced pretrained forced alignment toolkits has simplified the process.

In contrast, our approach requires neither clean data nor additional training. Similar to WhisperX [23], we leverage two off-the-shelf models. However, we apply Whisper to generate hypothesis text directly on the short-form audio segments, according to subtitle timestamps, as the long-form broadcast audio includes intervals of music and advertisements. Additionally, we take advantage of whisper's prompt-based capabilities during text generation. We then apply Wav2vec2.0 for word-level forced alignment to further refine subtitle timestamps, offering a straightforward yet effective method for data refinement.

2.2 Sequence-level mapping

Sequence-level mapping is primarily explored in unsupervised paradigms, where no explicit supervised objective is available. In the context of unsupervised speech recognition, generative adversarial networks (GANs) [24] have been investigated in several studies [25–28]. In these frameworks, the ASR recognizer acts as the generator, converting speech representations into hypothesis sequences. Meanwhile, the discriminator, typically a convolutional network, is trained to provide a sequence-level representation of the input generated sequences and real text samples and distinguishing between them. However, purely adversarial training is highly challenging. As in Wav2vec-U [27], it requires a complex preprocessing pipeline that includes the careful selection of appropriate acoustic speech representations and the clustering of speech frames into segments, in order to achieve

unsupervised phoneme recognition. Additionally, the real sequence is randomly sampled from text and not paired with the input speech in the unsupervised setup, which challenges the robustness of GAN training [29]. Actually, despite showing promising results on phoneme recognition, no existing approach has achieved well-performing results at the grapheme level.

In [30], word distributions (bag of words) of the text sequence have been proposed as target labels to address the issue of unordered targets in WS ASR tasks. This method works in specific scenarios where target labels are expected to contain the exact speech content but in a disordered sequence. However, subtitles tend to be shorter and clearer than the paired speech content. In some cases, current subtitles can originate from previous segments of speech, which introduces misalignment among utterances. These factors make word distributions of subtitles a suboptimal target for ASR training.

Drawing inspiration from previous approaches, we propose to introduce an additional adversarial loss to a standard ASR model. This architecture is more flexible than purely unsupervised methods, as the original ASR loss helps stabilize adversarial training and remove the need for extensive feature engineering. Additionally, our approach allows the discriminator to learn the overall representation of target sequence, offering a more adaptable alternative to pre-computed word distributions.

3 Datasets

3.1 Dataset description

The subtitled dataset used in this work comprises 3,399 h of multi-genre recordings from 16 TV programs broadcast by the Vlaamse Radio- en Televisieomroeporganisatie (VRT, Flemish Radio and Television Broadcasting Organisation) between September 2020 and November 2022. The recordings span four main genres: news, talk shows, entertainment, and drama, covering a wide range of topics, including politics, economics, education, culture, and sports.

Apart from audio segments like advertisements that do not have subtitles, 2860 h of audio are subtitled as displayed on the screen. However, the screen timings are far from perfect, which significantly reduces the quality of the weakly supervised data. Thus, we perform an initial normalization of the subtitles by filtering out music and non-speech sounds (e.g., applause, ringtones) when they are consistently annotated (e.g., music is often marked with an asterisk, and non-speech audio events are indicated in all uppercase). After normalization, the dataset is scaled down to 1720 h of speech, as shown in Table 1. Additionally, we split a subset of 720 h of speech (915k utterances) across all genres and topics to investigate the influence of data volume on speech recognition performance. We name these two training sets as *subs-720h* and *subs-1720h*, respectively. For reliable evaluation, we manually annotated 6 h of speech (5k utterances) from each genre, excluding entertainment, with verbatim transcripts. This annotated dataset, referred to as *subs-annot*, is excluded from the training sets.

The verbatim dataset is incorporated alongside the subtitled data for some experiments in this work. This dataset is sourced from the Corpus Gesproken Nederlands (CGN, Spoken Dutch Corpus) [31], a standard Dutch speech recognition database containing 900 h of speech from both Dutch and Flemish speakers. The Flemish portion

Table 1 Audio duration and number of utterances in the subtitled dataset

Genre	Dur	Num of utterances			
		Total	$< 5sec$	$5 \sim 15sec$	$> 15sec$
news	527.2 h	701,438	669,204	31,983	251
talk show	253.7 h	309,817	282,935	26,737	145
entertainment	25.6 h	32,824	32,051	773	0
drama	916.4 h	1,327,040	1,316,144	10,871	25
total	1722.9 h	2,371,119	2,300,334	70,364	421

comprises 270 h of speech, categorized into several components based on different types of speech and environments: (i) read speech prepared by professional newsreaders; (ii) formal speech as lectures, presentations, meetings and broadcast news; (iii) dialogue speech as spontaneous interviews, discussions, debates and commentaries; and (iv) narrowband telephone conversations. From this, we have derived a training set consisting of 240 h of speech (350k utterances), referred to as *cgn-train*. Additionally, a representative test set of 8 h of speech (7k utterances) has been created by sampling several recordings from each component, excluding telephone recordings, with complete separation of speakers between the training and test sets. This test set is referred to as *cgn-dev*.

3.2 ASR benchmarks and analysis

State-of-the-art (SoTA) ASR systems, particularly multilingual models, have made significant progress in recognizing languages for which limited data resources are available [32, 33]. Additionally, there are public ASR models specifically trained for Dutch. In this work, we focus on two typical end-to-end ASR architectures: connectionist temporal classification (CTC) [34] and attention-based encoder-decoder (AED) models [4, 35, 36]. We selected models available on HuggingFace [37] for benchmarking. The results of the benchmark tests on *subs-annot* and *cgn-dev are* shown in Table 2. One of the models, XLSR53-dutch, is a fine-tuned version of the cross-lingual Wav2vec2.0 model XLSR-53 [38], trained on the Dutch portion of Common Voice [39] and CSS10 [40]. Another model, MMS-all-ndl [41], is also based on the Wav2vec2.0 architecture and uses an adapter model fine-tuned for Dutch. Lastly, Whisper [6] is an AED model trained on 680k hours of audio from the web, including approximately 2k hours of Dutch, but it has not been specifically fine-tuned for Dutch.

Fine-tuning is an effective method for transferring knowledge to a specific domain, but the model's generalization is often limited by the dataset used during training. For example, while MMS reports a significant reduction in word error rate (WER) on the FLEURS benchmark [42] (which MMS was fine-tuned on), Whisper demonstrates better zero-shot performance on both *subs-annot* and *cgn-dev*, as shown in Table 2. This highlights an important insight from Whisper: the volume and diversity of the training dataset are crucial for developing general-purpose ASR systems. Moreover, the higher WER observed on *subs-annot* compared to *cgn-dev* suggests that broadcast audio in subtitled data contains more complex and challenging speech than the standard ASR dataset. This characteristic makes subtitled data particularly well-suited for training general ASR models.

Subtitle texts offer distinct advantages over other weakly supervised resources, such as the web-sourced captions used in Whisper [6]. These resources are often pseudo transcripts automatically generated by existing ASR models. Although SoTA models are capable of producing pseudo transcripts with relatively high accuracy, as demonstrated in the benchmarks, they tend to struggle when transcribing words that are either unseen or rarely encountered by the model during training. These words are often specific named entities, which can be crucial to the overall meaning of the text. In contrast, subtitles naturally include such terms. To illustrate this, we used the best pseudo transcripts generated by Whisper-large-v3 (see Table 2) and computed the recall for different ratios of low-frequency words from *subs-annot* in the pseudo transcripts, comparing these results with those in the subtitles. As shown in Fig. 1, while pseudo transcripts achieve a higher recall across words with higher frequency in *subs-annot*, subtitles outperform pseudo transcripts in recalling rare words—those in the bottom 0.1% and 1% frequency ranges.

Table 2 WER (%) benchmarks for ASR models on *subs-annot* and *cgn-dev*

Model	Size	*subs-annot*				*cgn-dev*			
		News	Talk show	Drama	Avg	Read	Formal	Dialogue	Avg
XLSR53-dutch	315M	28.60	36.88	59.57	38.62	17.58	31.71	43.95	34.35
Whisper medium	764M	15.41	21.52	31.11	21.55	8.41	16.45	26.38	19.43
Whisper large-v3	1.54B	10.04	13.27	22.48	14.01	4.44	10.71	19.61	13.62
MMS-all-ndl	965M	17.77	24.29	46.12	26.36	15.11	25.36	33.86	27.08

https://doi.org/10.1186/s13636-026-00478-x

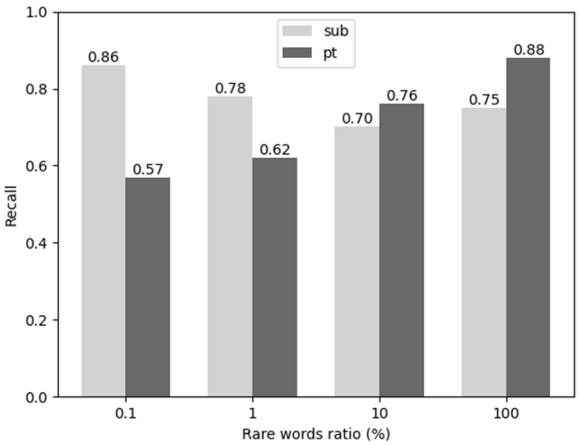

Fig. 1 Recall of rare words from verbatim transcripts in subtitles (sub) and pseudo transcripts (pt). The abscissa represents cumulative word frequency, sorted from the most rare words (left) to all words (right)

In summary, the combination of broadcast audio, with its varied and informal speech patterns, and reliable subtitles, which preserve low-frequency but contextually significant terms, makes subtitled data highly desirable for weakly supervised ASR training.

4 Subtitle refinement

4.1 Methods

The proposed refinement pipeline consists of four steps, as shown in Fig. 2. Since the full subtitled dataset contains 1720 h of audio (2.4 million utterances), we load portions of the long-form wave files using timestamps, rather than splitting audio into small segments. This approach alleviates the strain on the I/O system and CPU memory during batch processing. Consequently, all processing is performed based on segment timestamps, with no pre-processing applied to the audio itself.

In the first step, we shift each speech segment by -3 s at the beginning and -1 s at the end of the timestamp. This adjustment is necessary because subtitles typically lag behind the speech and often include content from the previous speech segment. The exact shift values were derived from *subs-annot*, where both verbatim and subtitle timestamps are available. We apply the maximum left shift and the minimum right shift to capture the complete content of the subtitles.

In the pseudo labeling step, we generate pseudo transcripts using the Whisper-large-v3 model. Whisper's text decoder is a generative model, which supports generative prompting. However, Whisper was not explicitly trained with specific formats of text prompts, making prompt design a challenge [43]. Table 3 provides an example of using the subtitle to prompt Whisper. The model struggles to generate rare proper names accurately, introducing spelling errors when no prompt is provided. While subtitles can provide additional context during transcript generation, directly using subtitle as a prompt (second row) often results in the decoder aligning the subtitles with the input speech, treating the subtitle as a prefix that is not re-generated. The third row in the prompt column is the left context subtitle and the output shows the decoder is highly sensitive to the prompt. In addition to the truncated transcript shown here, certain prompts cause disorder in the decoder by generating repeated terms. Alternatively, we prompt Whisper using rare words (those in the bottom 10% of frequency). A slight reduction in WER is observed, bringing the WER from 14.01% to 13.88% on *subs-annot*. Given that subtitles are sometimes oversimplified, we filter out very short subtitles based on the threshold given by the ratio between the number of words in the subtitles and the pseudo transcripts. Subtitles with fewer than half the number of words compared to the pseudo transcripts (threshold = 0.5) are excluded.

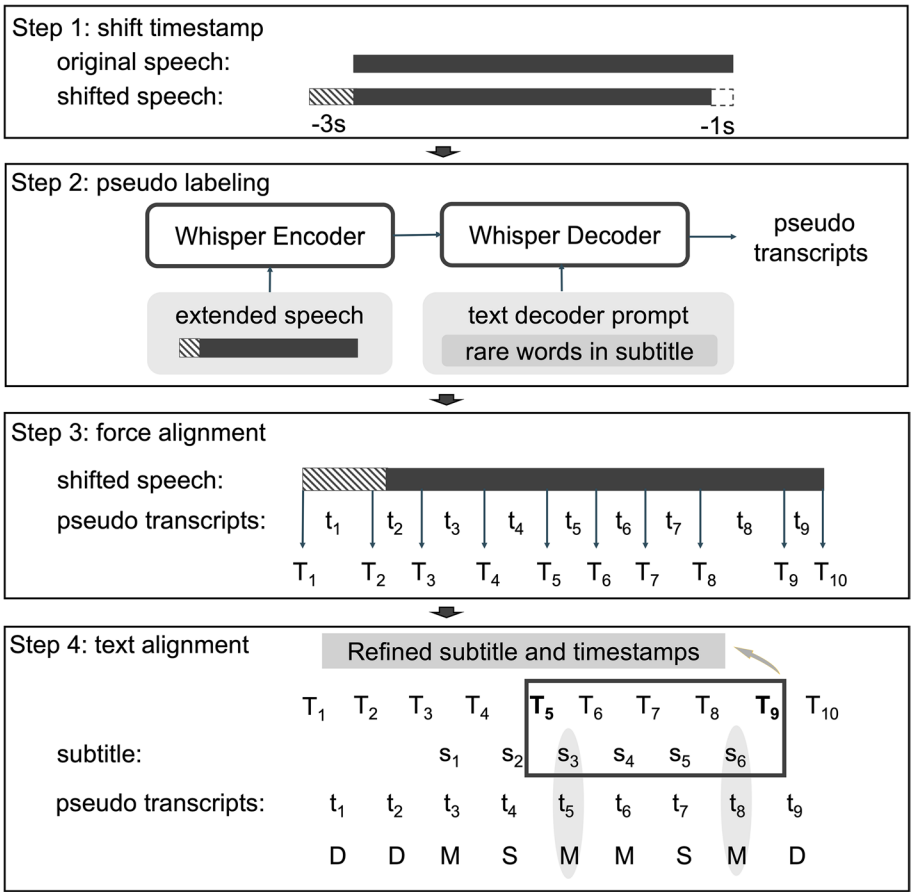

Fig. 2 Pipeline of subtitle refinement

Table 3 An example of prompting Whisper with subtitles. The first row in the prompt column shows no prompt, the second row uses the exact subtitle, the third row provides the preceding subtitle context, and the fourth row includes rare words

Subtitle	**Verbatim**
Bedankt, Peter Wittoeck.	Dank u wel, Peter Wittoeck.
Prompt	**Pseudo transcript**
-	Dank u wel, Peter Witthoek.
Bedankt, Peter Wittoeck.	...
En leven we vanaf morgen in een pasjesmaatschappij.	nd
Wittoeck	Dank u wel, Peter Wittoeck.

In the forced alignment step, we align the pseudo transcripts to the speech at the word level using a pretrained Wav2vec2.0 model. We perform forced alignment on the pseudo transcripts rather than the subtitles because Whisper's output is expected to have better alignment with the speech than the subtitles. The word-level timestamps obtained from this step are used to define the speech boundaries.

In the text alignment step (step 4 in Fig. 2), the pseudo transcript and subtitle are aligned. Since the shifted speech is 2 s longer than the original and subtitles are typically shorter than the full speech content, the pseudo transcript tends to have more words than the subtitle at the beginning, potentially covering words from the previous utterance. Additionally, since the start of an utterance frequently recalls words from the end of the prior one, this can result in alignment errors at the beginning of the subtitle. Therefore, we identify the start as the first match

of two consecutive words (s_3 and s_4) and the end as the last matched word (s_6). Together with the timestamps from the forced alignment step, we obtain the refined timestamps paired with the subtitles.

The refined {audio, subtitle} pairs obtained from step 4 are then used as input to train both CTC models and encoder-decoder models. Although the size of the refined dataset is reduced, the improved alignment enhances the accuracy of ASR training, as demonstrated in the Section 5.3.

4.2 Training details

We finetune both the CTC model XLSR53-dutch and the AED model Whisper-medium on the refined subtitled data, balancing the model size and computational resources.

For XLSR53-dutch, the input is raw audio sampled at 16 kHz. The feature encoder processes the raw waveform through 7 temporal convolutional blocks, each with 512 channels, producing latent speech representations. This results in an output frequency of 49 Hz, with each sample corresponding to a 25 ms window. The model further extracts contextual speech representations using 24 transformer blocks, each with a model dimension of 1024, an inner dimension of 4096, and 16 attention heads. During fine-tuning, a classification layer is added on top of the transformer blocks, along with the CTC loss, to map the contextual speech representations to the target vocabulary. The convolutional feature encoder remains frozen during fine-tuning. The target texts are tokenized into byte-pair encoding (BPE) (sub)word units, with a vocabulary size of 5000 BPE units, trained on both the verbatim transcriptions from *cgn-train* and the subtitles from *subs-train-1720h*. No language model is applied during decoding.

For Whisper, the input consists of 80-channel log-magnitude Mel spectrogram features, computed using 25 ms windows with a 10 ms stride. Both the encoder and decoder in the medium model consist of 24 transformer blocks, each with a model dimension of 1024, an inner dimension of 4096, and 16 attention heads, all of which are fine-tuned during training. The target texts are tokenized using a BPE tokenizer with a vocabulary size of 50k identical to the GPT-2 tokenizer [44].

The models were implemented using the HuggingFace Transformers library [37]. All models were trained on a single H100 GPU using the Adam optimizer [45]. For XLSR-53, we used a learning rate of 1×10^{-4}, a batch size of 128, and 2k warmup steps, with a maximum of 10k training steps and early stopping. For Whisper, we used a learning rate of 5×10^{-5}, a batch size of 64, and 5k warmup steps, with a maximum of 18k training steps.

4.3 Results and discussion

The complete set of 1720 h of subtitled data was refined using the proposed method, and the final dataset size is shown in Table 4. Utterances with extremely short subtitles were removed in step 2. Audio segments might also be truncated in step4 by adjusting the start and end timestamps based on text alignment. We finally get 580 h (733k utterances) and 1300 h (1.96M utterances) of audio for *subs-train720h* and *subs-train-1720h*, respectively. Although the total amount of data is reduced through this refinement process, the quality of the dataset is improved due to better alignment between the audio and the subtitles.

We evaluate ASR performance on the subtitle-based training data using both the CTC model XLSR53-dutch and the AED model Whisper-medium, and compare raw, pseudo, and refined training targets, as shown in Table 5. The first row reports the zero-shot benchmark results of the pretrained models without subtitle-based finetuning.

Table 4 Audio duration and number of utterances in the refined training set of subtitled data

Dataset	Duration		Num of utterances	
	Raw	Refined	Raw	Refined
subs-train-720h	716.1 h	578.3 h	915,776	732,721
subs-train-1720h	1722.9 h	1307.7 h	2,371,119	1,960,882

Table 5 WER (%) results on refined data compared with that on raw data or pseudo transcript from whisper using both XLSR53-dutch and Whisper-medium

Dataset		XLSR-53		Whisper-medium	
		subs-annot	*cgn-dev*	*subs-annot*	*cgn-dev*
None		38.62	34.35	21.55	19.43
cgn-train		25.18	15.87	10.83	**8.67**
subs-train-720h	Raw	47.13	47.82	39.02	41.44
	Pseudo	35.25	32.43	14.13	15.72
	Refined	22.76	17.21	9.97	12.70
subs-train-1720h	Raw	45.68	48.05	38.16	44.36
	Pseudo	33.82	32.17	13.92	15.49
	Refined	21.90	16.73	9.71	12.25
subs-train-1720h + *cgn-train*		**21.83**	**15.76**	**9.63**	8.89

Since no language model is used in XLSR decoding, the absolute WERs of XLSR and Whisper are not directly comparable; instead, we focus on the relative trends within each model.

Models finetuned on the standard ASR corpus *cgn-train* achieve strong performance on *cgn-dev*, but generalize less well to *subs-annot*. In contrast, when training only on subtitle-based data, using the raw subtitles as targets leads to very high WERs for both models. This is expected, since the raw subtitles are only loosely aligned with the speech and differ substantially from verbatim transcripts. For reference, the raw subtitles of *subs-annot* have a WER of 33.81% with respect to the verbatim transcripts.

Using Whisper-generated pseudo transcripts as training targets provides a much stronger baseline than raw subtitles, showing that the pseudo transcripts are acoustically better matched to the speech. However, the proposed refinement method consistently outperforms pseudo-transcript training for both subtitle subsets and both ASR models. This indicates that refinement is not simply reproducing Whisper outputs, but benefits from combining speech-grounded pseudo transcripts with the lexical and contextual information preserved in subtitle text. Compared with training on raw subtitles, refinement reduces WER by more than half on *subs-annot* for both XLSR53-dutch and Whisper-medium.

Finally, combining *subs-train-1720h* with *cgn-train* does not always improve performance on the cleaner *cgn-dev* set, where the best result is still obtained by finetuning Whisper-medium directly on *cgn-train*. However, the mixed setting achieves the best result for XLSR53-dutch on *subs-annot* and remains highly competitive for Whisper-medium, suggesting that combining large-scale subtitle data with standard ASR data can improve robustness and generalization, making them more suitable for general-purpose use.

We further analyze the recovery of named entities (NEs) on *subs-annot*, since overall WER does not fully reflect performance on rare but semantically important words. Table 6 reports recall and precision separately for in-vocabulary (IV) and out-of-vocabulary (OOV) NEs with respect to the verbatim transcripts. In the *subs-annot* set, there are 1558 IV NEs and 902 OOV NEs in total.

For IV NEs, the raw subtitles achieve the highest recall, indicating that subtitle text often preserves important contextual entities better than Whisper pseudo transcripts. At the same time, subtitle-trained models remain competitive in precision, suggesting that subtitle-based supervision does not substantially harm the generation

Table 6 Recall and precision of in-vocabulary (IV) and out-of-vocabulary(OOV) named entity (NE) with respect to the verbatim transcripts in the *subs-annot* set

Text	IV-NE		OOV-NE	
	Recall	Precision	Recall	Precision
Raw subtitle	60.19%	82.00%	18.07%	93.68%
Whisper pseudo transcript	41.79%	68.89%	21.95%	89.59%
Raw-subtitle-trained transcript	50.75%	89.47%	22.62%	94.44%
Refined-subtitle-trained transcript	46.27 %	90.32%	16.74%	94.97%
Refined-subtitle + CGN-trained transcript	59.70%	85.47%	53.22%	80.76%

https://doi.org/10.1186/s13636-026-00478-x

Table 7 Example of filler recovery in transcripts produced by different training settings

Verbatim transcript	uhm dus er waren heel wat emotionele problemen
Raw subtitle	er waren heel wat emotionele problemen
Whisper pseudo transcript	dus er waren heel wat emotionele problemen
Raw-subtitle-trained transcript	er waren heel wat emotionele problemen
Refined-subtitle-trained transcript	dus er waren heel wat emotionele problemen
Refined-subtitle + CGN-trained transcript	uhm dus d'r waren heel wat emotionele problemen

Table 8 Recall of fillers with respect to the verbatim transcripts in the *subs-annot* set

Text	Recall of fillers
Raw subtitle	2.58%
Whisper pseudo transcript	4.51%
Raw-subtitle-trained transcript	1.29%
Refined-subtitle-trained transcript	2.97%
Refined-subtitle + CGN-trained transcript	**79.31%**

of seen entities. However, refinement does not improve IV-NE recall over raw-subtitle training, which is consistent with the fact that the refinement process mainly improves speech–text alignment rather than introducing new lexical information.

A different pattern is observed for OOV-NEs. All subtitle-only settings show much lower recall, while adding CGN substantially improves OOV-NE recovery. This suggests that large-scale subtitle data alone is insufficient for robust recognition of unseen entities, and that additional standard ASR supervision helps the model better handle lexically challenging words. Overall, these results indicate that subtitle-based training retains useful entity information, especially for IV-NEs, but still has clear limitations for OOV-NEs.

To further examine the remaining mismatch between subtitle-based supervision and verbatim transcription, we analyze the recovery of fillers in spontaneous speech. Although the proposed refinement method improves overall ASR performance, the final training target remains subtitle text, which is written for readability and often omits fillers and hesitations. We therefore include Tables 7 and 8 to analyze this limitation.

Table 7 shows a representative case in which the filler "uhm" is present in the verbatim transcript but omitted in the subtitle, and this omission is reflected in the outputs of both the raw-subtitle-trained and refined-subtitle-trained models. In contrast, the model trained with additional CGN data recovers the filler.

Table 8 confirms this trend quantitatively. The verbatim references in *subs-annot* contain 1,706 fillers in total, including forms such as "uhm," "ah," and "hè." Both subtitle text and subtitle-trained models show very low filler recall, indicating that the weak recovery of such phenomena mainly stems from subtitle-style supervision itself rather than from the use of Whisper pseudo transcripts in the refinement process. Adding CGN, which provides verbatim ASR transcriptions, substantially alleviates this issue.

These results suggest that the main limitation is inherent to subtitle-based weak supervision: while subtitle refinement makes the training pairs more acoustically aligned and improves ASR performance, it does not fully remove the mismatch between readable subtitle text and spontaneous spoken content. This also explains why adding *CGN* is beneficial in our combined training setting: the subtitle data contributes scale and diversity, while the standard ASR data helps the model retain verbatim characteristics that are usually absent from subtitles.

5 Adversarial training

5.1 Methods

The raw {audio, subtitle} pairs, denoted as $\{x, y\}$ in the raw subtitled data, where each audio file is paired with a roughly aligned subtitle, are not optimal for standard ASR tasks. To address this, we propose adversarial training to map audio representations to text distributions. Figure 3 illustrates the framework of our approach.

The audio encoder extracts the speech representations, denoted as $\mathcal{S}$, from the input x. Given $\mathcal{S}$, the generator in the GAN architecture, acting as an ASR recognizer, computes a probability distribution over the grapheme vocabulary and outputs the grapheme sequence $G(\mathcal{S})$. The discriminator then takes both the one-hot encoded real text y and the generated text distribution $G(\mathcal{S})$ as inputs. This shallow CNN-based discriminator further extracts basic patterns and simple word structure from the input texts, condensing them into single-element representations, $D(y)$ and $D(G(\mathcal{S}))$, through mean pooling. These sequence-level representations disregard exact words within the sequence, hence avoiding alignment of audio and text. The representation for verbatim and subtitle data will also be similar. $D(y)$ and $D(G(\mathcal{S}))$ are then used for binary classification, where the goal is to distinguish the generated text (negative sample) from the real text (positive sample). As adversarial task, the generator is encouraged to produce text with sequence-level representations closely resemble those of the real text.

Adversarial training, however, is highly sensitive. Even slight changes to the architecture or substitute of dataset can lead to failure of training. To mitigate this sensitivity and avoid overly complex pre-processing of audio, we incorporate an ASR loss to stabilize the training process. As shown in Fig. 3, one split flow of the generated sequence $G(\mathcal{S})$ is used for computing ASR loss with y as the target, providing an additional learning signal.

For simplicity, the speech representations $\mathcal{S}$ are pre-trained and remain fixed during adversarial training. Specifically, we pre-train the audio encoder and the recognizer (generator) using the CTC loss on the CGN dataset, and we extract speech representations of subtitled data from the final layer of the audio encoder. While freezing the representations may result in a slight degradation of ASR performance, our goal is to demonstrate the effectiveness of adversarial training in weakly supervised ASR.

The total loss for the generator is composed of two parts: one for adversarial training ($\mathcal{L}_G^{gan}$) and one for ASR training ($\mathcal{L}_G^{asr}$). The generator's objectives are defined as follows:

$$\mathcal{L}_G^{gan} = \min_G \mathop{\mathbb{E}}_{\mathcal{S}} [log(1 - D(G(\mathcal{S})))] \tag{1}$$

$$\mathcal{L}_G^{asr} = \max_G \sum_{\mathcal{S}} log(p(y|\mathcal{S}; G)) \tag{2}$$

To balance the adversarial and ASR losses, we compute an adaptive scale factor α for $\mathcal{L}_G^{gan}$ and $\mathcal{L}_G^{asr}$:

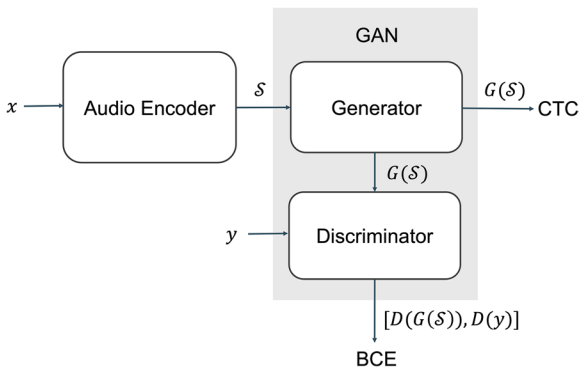

Fig. 3 The framework of the adversarial training

$$\alpha = \frac{||\nabla \mathcal{L}_G^{asr}||}{||\nabla \mathcal{L}_G^{gan}|| + \delta} \tag{3}$$

where $\nabla\mathcal{L}$ denotes the gradient of the loss function with respect to the generator parameters, and $\delta = 10^{-6}$ ensures numerical stability. Additionally, we introduce a fixed weight λ to control the relative importance of the two loss functions. The final objective for the generator is:

$$\mathcal{L}_G = \lambda \mathcal{L}_G^{asr} + \alpha(1-\lambda)\mathcal{L}_G^{gan} \tag{4}$$

The discriminator's objective remains unchanged from the original GAN formulation, where it assigns high probabilities to the real sequence y and low probabilities to the generated sequence $G(s)$:

$$\mathcal{L}_D = \max_D \left(\mathop{\mathbb{E}}_{y}[logD(y)] + \mathop{\mathbb{E}}_{\mathcal{S}}[log(1 - D(G(\mathcal{S})))] \right) \tag{5}$$

We also conducted unsupervised training to further verify the robustness of our approach. Following the methodology of wav2vec-U 2.0 [28], we keep four auxiliary objectives: gradient penalty $\mathcal{L}gp$, smoothness penalty $\mathcal{L}sp$, phoneme diversity loss $\mathcal{L}pd$, and auxiliary self-supervised loss $\mathcal{L}ss$ in the unsupervised training. However, given that the ASR loss in our approach stabilizes the adversarial training process, we chose to omit these auxiliary objectives for simplicity.

5.2 Training details

The proposed method features a stable architecture that eliminates the need for intricate design in a GAN model. We extract speech representations from the final transformer layer of the XLSR-53 model finetuned on CGN data (as described in Sec. 4) as input to the generator. The generator consists of a single linear layer, identical in size to the classifier used for finetuning XLSR-53, with an input dimension of 1,024 and an output BPE vocabulary of 5k units. Batch normalization is applied to the input speech features, and a CTC loss is added to the generator to enhance stability. The discriminator is composed of two causal CNN layers with a hidden size of 384. With a kernel size of 8 and a stride of 1, the discriminator achieves a receptive field size of 15.

To compare with pure unsupervised training, we also implement a model with the same configuration as wav2vec-U 2.0 [28], where the generator is a two-layer CNN with a stride of 3. Additionally, an auxiliary self-supervised objective is incorporated into the adversarial training, where self-supervised labels are generated through k-means clustering on Mel-frequency cepstral coefficient (MFCC) features.

The models were implemented using the Fairseq library [46]. A batch size of 80 was used for training all models. Models with a linear generator were trained for 5k updates, while models with CNN generators were trained for 100k updates. Fixed learning rates were set at 3×10^{-4} for the discriminator, 1×10^{-5} for the linear generator, and 5×10^{-5} for the CNN generator. For fully unsupervised training, the model was trained for 320k updates. Following the wav2vec-U 2.0 approach, 64 k-means clusters were applied to the MFCC features as self-supervised labels, and the weights for $\mathcal{L}gp$, $\mathcal{L}sp$, $\mathcal{L}pd$, and $\mathcal{L}ss$ were set to 1.0, 1.5, 3.0, and 0.5, respectively.

5.3 Results

We present the experimental results of the proposed adversarial training for grapheme recognition, evaluated based on word error rate (WER) on the test set *subs-annot*. Additionally, experiments using the verbatim CGN data are evaluated on the *cgn-dev* test set as well.

We first investigate the impact of incorporating $\mathcal{L}^{asr}$ on the stability of adversarial training. The relative importance of $\mathcal{L}^{asr}$ and $\mathcal{L}^{gan}$ can be controlled by the weights λ and $(1-\lambda)$ in the loss function Equation (4). Table 9

Table 9 WER (%) results for the proposed model trained on *subs-train-720h* with varing values of λ

Weight λ	WER (%) *subs-annot*
1.0	29.43
0.9	29.12
0.8	25.96
0.7	**23.81**
0.6	26.37
0.5	> 100

presents the results for various combinations of these weights, with the first row representing the baseline model ($\lambda = 1$), which is trained solely with weakly supervised learning from the subtitled data.

As is well known, the convergence of adversarial training relies on a delicate balance between the generator and the discriminator. Since the generator's inherently random nature makes it harder to train than the discriminator, the hyperparameter λ plays a critical role in facilitating the generator's convergence. In Table 9, we observe that the model begins to diverge at $\lambda = 0.5$, and the divergence becomes more pronounced as λ decreases further. Nevertheless, λ improves results across a wide range, with its optimal value depending on the specific application. Moreover, when working with more weakly supervised data sources, such as titles and post texts from general video data, a smaller λ is expected to be more effective.

Next, we examine the role of $\mathcal{L}^{gan}$ in the proposed method. The baseline model achieves a higher WER (29.43%) compared to the WER (25.18%) obtained by finetuning XLSR53 on CGN and testing on *subs-annot*, as shown in Table 5. This result is expected, as our setup trains the linear classifier on top of fixed speech representations, and the classification layer has been modified. However, when incorporating adversarial training with $\lambda = 0.7$, we observe a 5.62% improvement in WER on *subs-annot*. Although the final WER (23.81%) remains slightly higher than the WER (22.76%) achieved by fine-tuning XLSR-53 on the refined *subs-train-720h*, we argue that introducing adversarial loss offers a flexible and adaptable approach for any end-to-end ASR model. Incorporating this method into a CTC ASR model, where the speech features are learned in conjunction with the classifier, is likely to yield even better performance.

To highlight the importance of $\mathcal{L}^{gan}$ on ASR outputs, Table 10 provides a sample utterance from a natural conversation from the *subs-annot* set. The verbatim transcript *ref*, translates as "yeah, no, I'm not feeling well, you know. Could I come over?". The differences in word choice (e.g., "mij" vs. "me" and "kunnen" vs. "mogen") and the absence of interjections like "seg" in the subtitle *sub* illustrates that subtitles are often simplified or adjusted for clarity and readability. For comparison, the third row in Table 10 presents the output *hyp* from the fixed audio encoder of the proposed model, which uses the XLSR-53 model trained on CGN. Although some acoustic clues are retained in the hypothesis, there are notable errors as a result of the limited training data and the domain mismatch between the training set CGN and the test set *subs-annot*. At $\lambda = 1.0$, where the generator is trained on the subtitled data solely using the CTC loss, the speech features are preserved from the audio encoder, but the generator tends to adopt subtitle-like patterns, omitting elements such as "ja" from the speech, while quite some

Table 10 An example of ASR outputs (hyp) from the proposed model at $\lambda = 1.0$ and $\lambda = 0.7$. The third row shows the outputs from the fixed audio encoder of the proposed model, trained on CGN

ref	ja nee ik voel mij niet goed seg zou ik kunnen langskomen
sub	nee ik voel me niet goed zou ik mogen langskomen
hyp for XLSR53-cgn	janerek voor mench hroor zich zou jet je nog langskomen
hyp ($\lambda = 1.0$)	ne ik vond mej niet goed zeg zou it kunnen langskomen
hyp ($\lambda = 0.7$)	ja nee ik voel me niet goed zeg zou ik kunnen langskomen

Table 11 WER (%) results comparing wav2vec-U 2.0 trained on subtitled data with tuned auxiliary losses (first row) to models incorporating CTC loss with either removed auxiliary losses (second row) or fixed auxiliary losses (third row)

Model	subs-annot
wav2vec-U 2.0	> 100
+ CTC	24.09
+ CTC + aux	**24.02**

Table 12 WER (%) results for the proposed model trained on the extra data

Data	*subs-annot*	*cgn-dev*
subs-train-720h	23.81	25.45
subs-train-1720h	23.67	25.41
+ CGN	**22.58**	**17.47**

recognition errors remain ("nee" → "ne"; "voel" → "vond"; "mij" *rightarrow* "mej"). However, with the introduction of $\mathcal{L}^{gan}$ at $\lambda = 0.7$, the generator shows an enhanced ability to restore missing elements from the subtitles and produce a more accurate verbatim transcript.

Furthermore, we demonstrate the robustness of our proposed approach by applying it to the Wav2vec-U 2.0 architecture [28], where the generator is a single-layer CNN. We trained the Wav2vec-U 2.0 model on subtitle data, with the results shown in the first row of Table 11. The original Wav2vec-U 2.0 achieves excellent performance in phoneme recognition through unsupervised training on English. However, when applied to grapheme recognition, the model fails to converge, even after tuning the regularization term weights and testing different initialization values for the batch normalization scaling factor, which were reported as critical strategies for convergence in Wav2vec-U 2.0. Grapheme recognition presents a significant challenge to the robustness of Wav2vec-U 2.0. Moreover, Wav2vec-U 2.0 was implemented on read speech, which is much more well-structured than the spontaneous speech in the subtitled data. In contrast, integrating $\mathcal{L}^{asr}$ significantly stabilizes the adversarial training, even when using a different generator than initially proposed. With $\lambda = 0.7$ for $\mathcal{L}^{asr}$, which yields the best results in Table 9, we achieve a comparable WER of 24.09% using a CNN generator. While adding auxiliary objectives as $\mathcal{L}gp$, $\mathcal{L}sp$, $\mathcal{L}pd$, and $\mathcal{L}ss$ from Wav2vec-U 2.0 to the proposed model does not lead to significant improvements in WER, it introduces additional hyperparameters that require careful tuning.

Finally, we scale up the subtitled data to 1720 h. Additionally, we also address some of the challenges posed by subtitle data, such as the difficulty in predicting hesitations (e.g., 'uh') or handling apostrophes in abbreviated words, by incorporating the verbatim CGN dataset. This allows the model to learn from a broader range of speech patterns and linguistic variations. We train our proposed model with $\lambda = 0.7$, using both the scaled subtitle data and CGN, ensuring that the model benefits from both spontaneous speech in subtitles and more structured speech from CGN. The speech representations for the CGN data, fed to the generator, are also extracted from the finetuned XLSR-53 model. As shown in Table 12, the linear classifier does not show significant improvements from the larger subtitled dataset, likely because the audio encoder is not updated during training, and the additional data may be redundant in this context. However, the inclusion of CGN verbatim data still contributes to a further reduction in WER, even though CGN is only seen by the classifier. This suggests that while the pretrained speech representations may have reached a saturation point, the classifier can still benefit from learning the more structured speech patterns provided by the verbatim data.

We also report NE recovery after adversarial training in Table 13. Compared with the refinement-based setting, adversarial training yields consistently lower recall for both IV and OOV NEs, while precision remains relatively stable. This suggests that the generator is less effective at recovering entity tokens under adversarial supervision, even though it does not introduce severe hallucination. One possible explanation is that the discriminator provides a relatively coarse-grained supervision signal at the sequence level, rather than explicit token-level guidance for

Table 13 Recall and precision of in-vocabulary (IV) and out-of-vocabulary(OOV) named entity (NE) after adversarial training

Data	IV-NE		OOV-NE	
	Recall	Precision	Recall	Precision
subs-train-720h	38.52%	72.09%	17.62%	83.73%
subs-train-1720h	40.09%	72.50%	18.24%	84.43%
+ CGN	43.50 %	71.91%	24.33%	79.57%

lexical recovery. Combined with the imperfect alignment and compression in subtitle supervision, this may cause the generator to produce outputs that are globally plausible but locally mismatched to the verbatim transcript, which particularly affects the recall of NEs. At the same time, the reasonably high precision indicates that the generated text is still constrained by the ASR objective and tends to follow patterns observed in the training data, instead of generating arbitrary entity strings. As in the refinement-based setting, adding CGN improves both IV-NE and OOV-NE recall, with a more visible gain for OOV entities. This again suggests that standard ASR-style verbatim supervision complements subtitle-based training by providing more accurate lexical guidance for entity recovery.

6 Conclusions

In conclusion, this work shows that TV subtitles are a valuable resource for weakly supervised ASR. On the one hand, subtitle text preserves useful lexical information, especially low-frequency words and in-vocabulary named entities, making it a competitive supervision source compared with purely automatically generated transcripts. On the other hand, subtitles are often loosely aligned with speech and are written for readability rather than full verbatim transcription, which introduces both temporal mismatch and omissions of spontaneous speech phenomena such as fillers.

To address the alignment problem, we proposed a subtitle refinement pipeline that combines Whisper-based pseudo transcriptions, forced alignment, and subtitle-specific processing. The refined data substantially improved ASR performance for both the CTC model XLSR53-dutch and the AED model Whisper-medium, and consistently outperformed training on both raw subtitles and direct Whisper pseudo transcripts. These results show that refinement is effective because it combines the acoustic grounding of pseudo transcripts with the lexical and contextual information preserved in subtitle text. At the same time, our additional analyses reveal an important limitation of subtitle-based supervision: while it is beneficial for recovering many content words and named entities, it remains weak for spontaneous speech phenomena and out-of-vocabulary named entities. Adding standard ASR data such as CGN helps alleviate this limitation by providing verbatim supervision.

To provide a more flexible alternative that does not rely on a multi-stage refinement procedure or tightly structured subtitle pairs, we further introduced adversarial training for subtitle-based ASR. This approach can be integrated into end-to-end ASR models and enables training under weaker alignment assumptions. Although the adversarial approach achieved competitive overall ASR performance, our analyses show that its token-level recovery of named entities is weaker than that of the refinement-based approach, likely because the discriminator provides a coarser distribution-level supervision signal than explicit token-aligned targets.

Overall, our findings suggest that subtitle refinement is the stronger solution when subtitle structure can be exploited, while adversarial training offers a more flexible alternative for weakly aligned settings. In future work, we plan to extend the adversarial training framework to AED ASR systems and to evaluate both approaches on less structured and more distant supervision sources, such as loosely related text collections, where the mismatch between text and speech is even more severe.

Acknowledgements The authors gratefully acknowledge the dataset provided by the Flemish Radio and Television Broadcasting Organisation.

Authors' contributions XZ implemented the proposed method, conducted the experiments, and wrote the manuscript. HVH supervised the work and revised the manuscript. All authors have read and approved the final manuscript.

Funding This research was supported by Research Foundation Flanders (FWO) under grant S004923N of the SBO programme.

Data availability The subtitled dataset used in this paper are available from Vlaamse Radio- en Televisieomroeporganisatie (VRT, Flemish Radio and Television Broadcasting Organisation) and so is not publicly available. The data can be accessed from the authors upon reasonable request, subject to VRT's permission. The CGN dataset used in this paper is publicly available from https://taalmaterialen.ivdnt.org/download/tstc-corpus-gesproken-nederlands.

Declarations

Competing interests The authors declare that they have no competing interests.

Open Access This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by-nc-nd/4.0/.

References

1. A.L. Maas, P. Qi, Z. Xie et al., Building dnn acoustic models for large vocabulary speech recognition. Comput. Speech Lang. **41**, 195–213 (2017)
2. O. Abdel-Hamid, A. Mohamed, H. Jiang et al., Convolutional neural networks for speech recognition. IEEE/ACM Trans. Audio. Speech Lang. Process. **22**(10), 1533–1545 (2014)
3. O. Vinyals, S.V. Ravuri, D. Povey, in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*. Revisiting recurrent neural networks for robust ASR (2012), pp. 4085–4088
4. L. Dong, S. Xu, B. Xu, in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*. Speech-transformer: A no-recurrence sequence-to-sequence model for speech recognition (2018), pp. 5884–5888
5. C. Sun, A. Shrivastava, S. Singh et al., in *Proc. IEEE Int. Conf. Comput. Vis.* Revisiting unreasonable effectiveness of data in deep learning era (2017), pp. 843–852
6. A. Radford, J.W. Kim, T. Xu et al., in *Proc. Int. Conf. Mach. Learn.Robust speech recognition via large-scale weak supervision* (PMLR, 2023), pp. 28492–28518
7. P. Peng, B. Yan, S. Watanabe, D. Harwath, in *Proc. Interspeech*. Prompting the hidden talent of web-scale speech models for zero-shot task generalization (Dublin, 2023)
8. L. Zhuo, R. Yuan, J. Pan et al., in *Proc. ISMIR*. LyricWhiz: Robust multilingual zero-shot lyrics transcription by whispering to ChatGPT (2023)
9. W. Liu, Y. Qin, Z. Peng et al., in *Proc. ICASSP 2024 IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*. Sparsely shared LoRA on Whisper for child speech recognition (IEEE, 2024), pp. 11751–11755
10. M. Fuckner, S. Horsman, P. Wiggers et al., in *Proc. 2023 Int. Conf. Speech Technol. Hum.-Comput. Dialogue (SpeD)*. Uncovering bias in ASR systems: Evaluating Wav2vec2 and Whisper for Dutch speakers (IEEE, 2023), pp. 146–151
11. M.L. Cheng, C.Y. Wang, J. Huang, X.B. Wang, in *Proc. Interspeech*. Weakly supervised construction of ASR systems from massive video data (2021), pp. 4533–4537
12. K. Singh, V. Manohar, A. Xiao et al., in *Proc. Interspeech*. Large scale weakly and semi-supervised learning for low-resource video ASR (2020), pp. 3770–3774
13. J. Poncelet, H. Van Hamme, in *Proc. Spoken Language Technology Workshop (SLT)*. Learning to jointly transcribe and subtitle for end-to-end spontaneous speech recognition (IEEE, 2023), pp. 182–189
14. A. Baevski, Y. Zhou, A. Mohamed, wav2vec 2.0: A framework for self-supervised learning of speech representations. Adv. Neural. Inf. Process. Syst. **33**, 12449–12460 (2020)
15. J. Zhu, C. Zhang, D. Jurgens, in *Proc. ICASSP IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*. Phone-to-audio alignment without text: A semi-supervised approach (IEEE, 2022), pp. 8167–8171

16. P. Bell, M.J.F. Gales, T. Hain et al., in *Proc. Workshop on Automatic Speech Recognition and Understanding (ASRU)*. The MGB challenge: Evaluating multi-genre broadcast media recognition (IEEE, 2015), pp. 687–693
17. A. Ali, P. Bell, J. Glass et al., in *Proc. IEEE Spoken Language Technology Workshop (SLT)*. The MGB-2 challenge: Arabic multi-dialect broadcast media recognition (IEEE, 2016), pp. 279–284
18. A. Ali, S. Vogel, S. Renals, in *Proc. IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. Speech recognition challenge in the wild: Arabic MGB-3 (IEEE, 2017), pp. 316–322
19. A. Ali, S. Shon, Y. Samih et al., in *Proc. IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. The MGB-5 challenge: Recognition and dialect identification of dialectal Arabic speech (IEEE, 2019), pp. 1026–1033
20. V. Manohar, H. Hadian, D. Povey et al., in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*. Semi-supervised training of acoustic models using lattice-free MMI (2018), pp. 4844–4848
21. P. Lanchantin, M.J.F. Gales, P. Karanasou et al., in *Proc. Interspeech*. Selection of multi-genre broadcast data for the training of automatic speech recognition systems (2016), pp. 3057–3061
22. V. Manohar, D. Povey, S. Khudanpur, in *Proc. IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. JHU Kaldi system for Arabic MGB-3 ASR challenge using diarization, audio-transcript alignment and transfer learning (IEEE, 2017), pp. 346–352
23. M. Bain, J. Huh, T. Han et al., Whisperx: Time-accurate speech transcription of long-form audio (2023). ArXiv preprint arXiv:2303.00747
24. I. Goodfellow, J. Pouget-Abadie, M. Mirza et al., Generative adversarial networks. Commun. ACM **63**(11), 139–144 (2020)
25. D.R. Liu, K. Chen, H.Y. Lee, L.S. Lee, in *Proc. Interspeech*. Completely unsupervised phoneme recognition by adversarially learning mapping relationships from audio embeddings (2018)
26. K.Y. Chen, C.P. Tsai, D.R. Liu et al., in *Proc. Interspeech*. Completely unsupervised speech recognition by a generative adversarial network harmonized with iteratively refined hidden Markov models (2019)
27. A. Baevski, W.N. Hsu, A. Conneau et al., Unsupervised speech recognition. Adv. Neural. Inf. Process. Syst. **34**, 27826–27839 (2021)
28. A.H. Liu, W.N. Hsu, M. Auli et al., in *Proc. IEEE Spoken Language Technology Workshop (SLT)*. Towards end-to-end unsupervised speech recognition (IEEE, 2023), pp. 221–228
29. A.H. Liu, C.I.J. Lai, W.N. Hsu et al., in *Proc. Interspeech*. Simple and effective unsupervised speech synthesis (2022), pp. 843–847
30. V. Pratap, Q.T. Xu, T. Likhomanenko et al., in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*. Word order does not matter for speech recognition (IEEE, 2022)
31. N. Oostdijk, in *Proc. Int. Conf. on Language Resources and Evaluation (LREC)*. The Spoken Dutch Corpus: Overview and first evaluation, vol. 2 (2000)
32. C. Yi, J. Wang, N. Cheng et al., Applying wav2vec 2.0 to speech recognition in various low-resource languages (2020). ArXiv preprint arXiv:2012.12121
33. B. Li, R. Pang, T.N. Sainath et al., in *Proc. IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. Scaling end-to-end models for large-scale multilingual ASR (IEEE, 2021), pp. 1011–1018
34. A. Graves, S. Fernández, F. Gomez et al., in *Proc. Int. Conf. Mach. Learn.* Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks (2006), pp. 369–376
35. W. Chan, N. Jaitly, Q. Le et al., in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*. Listen, attend and spell: A neural network for large vocabulary conversational speech recognition (IEEE, 2016), pp. 4960–4964
36. A. Zeyer, P. Bahar, K. Irie et al., in *Proc. IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. A comparison of transformer and LSTM encoder-decoder models for ASR (IEEE, 2019), pp. 8–15
37. T. Wolf, L. Debut, V. Sanh et al., in *Proc. Conf. Empirical Methods Nat. Lang. Process.: Syst. Demonstrations*. Transformers: State-of-the-art natural language processing (2020), pp. 38–45
38. A. Conneau, A. Baevski, R. Collobert et al., in *Proc. Interspeech*. Unsupervised cross-lingual representation learning for speech recognition (2021)
39. R. Ardila, M. Branson, K. Davis et al., in *Proc. Twelfth Lang. Resour. Eval. Conf.* Common Voice: A massively-multilingual speech corpus (2020), pp. 4218–4222
40. K. Park, T. Mulc., Css10: A collection of single speaker speech datasets for 10 languages (2019). ArXiv preprint arXiv:1903.11269
41. V. Pratap, A. Tjandra, B. Shi et al., Scaling speech technology to 1,000+ languages. J. Mach. Learn. Res. **25**(97), 1–52 (2024)
42. A. Conneau, M. Ma, S. Khanuja et al., in *Proc. IEEE Spoken Language Technology Workshop (SLT)*. Fleurs: Few-shot learning evaluation of universal representations of speech (IEEE, 2023), pp. 798–805
43. X. Zhao, H. Van Hamme, in *2025 33rd European Signal Processing Conference (EUSIPCO)*. Refining Transcripts With TV Subtitles by Prompt-Based Weakly Supervised Training of ASR (IEEE, 2025), pp. 346–350
44. A. Radford, J. Wu, R. Child, Language models are unsupervised multitask learners (2019)

45. D.P. Kingma, J. Ba, in *Proc. Int. Conf. on Learn. Representations (ICLR)*. Adam: A method for stochastic optimization (2015)
46. C. Wang, Y. Tang, X. Ma et al., in *AACL-IJCNLP*. FAIRSEQ S2T: Fast speech-to-text modeling with FAIRSEQ (2020), p. 33

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Authors and Affiliations

Xinnian Zhao[1] · **Hugo Van Hamme**[1]

✉ Xinnian Zhao
xzhao1@esat.kuleuven.be

Hugo Van Hamme
hugo.vanhamme@esat.kuleuven.be

[1] Department of Electrical Engineering ESAT-PSI, KU Leuven, Kasteelpark Arenberg 10, 3001 Leuven, Belgium