Poncelet and Van hamme
*Journal on Audio, Speech, and Music Processing* (2026) 2026:20
https://doi.org/10.1186/s13636-026-00450-9

Journal on Audio,
Speech, and Music Processing

METHODOLOGY

Open Access

# Leveraging broadcast media subtitle transcripts for automatic speech recognition and subtitling

Jakob Poncelet[1*] and Hugo Van hamme[1]

**Abstract**

The recent advancement of speech recognition technology has been driven by large-scale datasets and attention-based architectures, but many challenges still remain, especially for low-resource languages and dialects. This paper explores the integration of weakly supervised transcripts from TV subtitles into automatic speech recognition (ASR) systems, aiming to improve both verbatim transcriptions and automatically generated subtitles. To this end, verbatim data and subtitles are regarded as different domains or languages, due to their distinct characteristics. We propose and compare several end-to-end architectures that are designed to jointly model both modalities with separate or shared encoders and decoders. The proposed methods are able to jointly generate a verbatim transcription and a subtitle. Evaluation on Flemish (Belgian Dutch) demonstrates that a model with cascaded encoders and separate decoders allows to represent the differences between the two data types most efficiently while improving on both domains. Despite differences in domain and linguistic variations, combining verbatim transcripts with subtitle data leads to notable ASR improvements. The proposed methods circumvent the need for extensive preprocessing, filtering and cleaning of the subtitled data, which is commonly used to leverage such data for ASR training. Additionally, experiments with a large-scale subtitle dataset show the scalability of the proposed approach. The methods not only improve ASR accuracy but also generate subtitles that closely match standard written text, offering several potential applications.

**Keywords** Automatic speech recognition, Weak supervision, End-to-end modelling, Subtitles, Broadcast media data

## 1 Introduction

Speech recognition has seen remarkable improvements in recent years due to the introduction of large-scale datasets and attention-based architectures to the pre-training of speech models [1–4]. Many efforts have focused on self-supervised pre-training to learn representations from large amounts of diverse, unlabeled data, which achieves strong results on downstream tasks like automatic speech recognition (ASR) with minimal labeled data for fine-tuning and improved generalisation to different domains [5–8]. As a result, massive multilingual pre-trained models are bridging the performance gap between high and low-resource languages [9], although there are still many languages and local dialects for which the automatic transcriptions are far from human-quality [10].

Similarly, scaling up supervised training has produced speech models with powerful decoders capable of performing multiple tasks jointly, such as speech recognition and speech translation [11, 12].

However, when ASR models are trained on weakly supervised web-scraped data as in Whisper [11], the advantages of manually labelled ASR datasets are often omitted, and the ability to transcribe exactly *ad verbatim* is usually lost. Therefore, this work investigates the

*Correspondence:
Jakob Poncelet
jakob.poncelet@kuleuven.be
[1] Department Electrical Engineering ESAT-PSI, KU Leuven, Kasteelpark Arenberg 10, B-3001 Leuven, Belgium

© The Author(s) 2026. **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/.

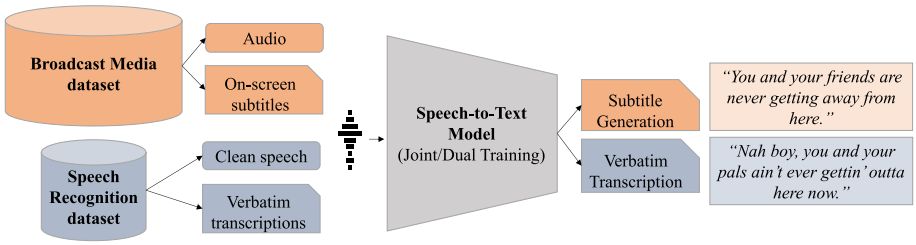



**Fig. 1** Overview of the proposed approach. Verbatim transcriptions from ASR datasets and subtitle transcripts from a large source of broadcast media data are gathered. A dual speech-to-text model is trained to output a verbatim transcription for the input speech and to generate a well-suited subtitle at the same time, by jointly learning from both data streams

possibilities to include both manual transcripts from ASR datasets and weakly supervised transcripts in the form of TV subtitles into an ASR system, by jointly performing verbatim speech recognition and automatic subtitling in a multitask model, as illustrated in Fig. 1.

TV subtitles are interesting for ASR training for several reasons. First, they are abundantly available in many languages and have been manually produced by human annotators.[1] Subtitlers apply specific rephrasings (e.g. shortening, paraphrasing), corrections (e.g. word choice, grammar, hesitations, repetitions) and normalisations (e.g. dialect) adhering to strong guidelines to improve the readability and comprehension for the viewer [13, 14]. As such, subtitles offer a very useful mapping between spoken language, with all its disfluencies and mistakes, and more standardised written language. Second, subtitles cover many domains, ranging from read and prepared speech in broadcast news reports to conversational and spontaneous speech in talk shows, interviews, and even soaps and sitcoms. These real-life dialogues are far from the speech in most labelled speech datasets [15]. Third, as a broader domain is covered, dialectal speech[2], accented speech and non-native speech are better represented than in common speech datasets, increasing the potential to improve the robustness and generalisation of ASR models. Finally, as the amount of spoken multimedia content is growing at a fast pace, automatic subtitling solutions are becoming increasingly worthwhile and cost-effective. An automatic subtitling model already in the style of real subtitles would drastically reduce the effort for human subtitlers compared to a verbatim model. Moreover, transcripts in the style of standard written text can be more effectively utilised by large language models (LLM), trained on large text datasets, compared to verbatim transcripts.

However, applying subtitles to ASR is non-trivial. As there is a big shift between exact, verbatim transcriptions and standardised, clean TV subtitles, they are ineffective for direct end-to-end (E2E) ASR training. On top of that, the timestamps of subtitles can be inaccurate and misaligned to the audio [16]. The standard approach to deal with these issues is to apply (iterative) data cleaning and data curation [11, 17, 18]. In this approach, the audio is pseudo-labeled with a pre-trained ASR model, and the predicted transcripts are compared to the corresponding subtitles or video captions (using some error metric). Then, only the "clean" segments, i.e. segments with a high agreement between the prediction and the corresponding subtitle, are retained for further training. While effective at large scale, this iterative cleaning approach has two major drawbacks: (1) it also filters out heavily accented, dialectal, noisy and difficult speech segments for which the pre-trained model performs sub-par, which is actually very valuable speech data to improve the model, and (2) in case the filtered subtitles are used as training targets, it produces models that tend to lose full verbatim fidelity, generating outputs that lie somewhere between verbatim transcripts and subtitle-style text [11].

In this work, we compare and propose several *supervised strategies* to combine verbatim and subtitle annotations to build strong multitask encoder-decoder ASR models, which improve verbatim speech recognition compared to standalone end-to-end models and introduce enhanced subtitling capabilities. Our approach explicitly separates the subtitle and verbatim annotations to train a joint model and does not require extensive preprocessing (e.g. alignment, filtering, iterative refinement) of the subtitle data, compared to the standard approach. There is no parallel data available for which both annotation types are present.

In previous work [19], we have proposed to combine both data streams with a parallel model, where the encoder is shared for both tasks, and two separate

[1] In Flanders (Belgium), broadcasted media is subtitled specifically to support hard-of-hearing people, a practice legally mandated to some extent by the Flemish government, following European Union accessibility guidelines.

[2] In recent years, there has been a rise in the popularity of strongly dialectal speech in several TV shows in Flanders.

decoders are trained, conditioned on the outputs of the shared encoder. Although this strategy has the benefits of multitask learning and results in an improved encoder, it leaves out possibilities to improve the decoder and address the domain shift between the two datasets. First, we propose to view a subtitle as a translated version of the verbatim transcription. To this end, we add an additional subtitle encoder block and cascade it to the ASR encoder, to transform the verbatim output features of the ASR encoder into subtitle features. The two separate decoders are then conditioned on these different encoder outputs. As the encoded features of both encoders are useful for the decoders, we introduce a double cross-attention in the decoders to both encoder outputs. We investigate several architectures that incorporate these ideas. Second, we explore a task-specific shared decoder that is conditioned on a task token representing which transcription type to generate, as has been proposed for multilingual [12] and multitask [20] training. The idea of a decoder that can perform multiple tasks, conditioning on a prompt or task description, has seen increased popularity in recent state-of-the-art work in speech recognition and beyond [11], making it a very suitable candidate for comparison.

All experiments are carried out on Belgian Dutch (Flemish), which is a medium-resourced language with many local dialects and a significant shift between written language and colloquial spoken language. We start from a strong Conformer-based, encoder-decoder E2E ASR baseline to which we make the proposed adaptations. All models are evaluated for verbatim transcription and subtitling capabilities on test sets with read speech and prepared talks and with spontaneous speech. We also carry out scaling experiments, showing the benefits of large-scale data, and introduce serialised output training [21]. While the direct E2E ASR model degrades by using the subtitles as if they were verbatim, the proposed models show strong improvements on all evaluations. Finally, we compare our method to Whisper and evaluate a pipeline approach that generates subtitles with an LLM from ASR transcripts.

The main *contributions* of this research paper can be summarised as:

- The investigation of several approaches to combine subtitles with verbatim data for end-to-end ASR-training, without requiring extensive cleaning and filtering of data sources,
- Proposing new architectures and supervised training techniques specifically tailored for weakly-supervised transcripts, more specifically TV subtitles, which strongly improve ASR performance,
- The expansion of a joint modelling framework to construct models that are able to generate both averbatim transcription and a subtitle, and
- An extensive evaluation of different training paradigms on small and large-scale datasets, and acomparison to state-of-the-art models and methods.

All produced code and resulting models are made open-source.[3,4]

## 2 Related work

This section gives an overview of recent advancements within ASR and related fields that make use of subtitles. The proposed subtitle method is also situated within the context of weakly supervised ASR training with imperfect labels. In some works, generating a transcription in the same language as is spoken (*intralingual*) is termed captioning, while generating a transcription in a different language (*interlingual*) is termed subtitling [22]. This work focuses only on intralingual subtitling. We use the term subtitling, as all data has arisen from subtitles on TV, and captions can have a broader meaning (e.g. image captioning).

### 2.1 Subtitles in ASR

Traditionally, there have been two main approaches to leverage subtitle data for ASR. First, when generating pseudo-labels of broadcast media data with a pre-trained acoustic model, the subtitles can be exploited to filter and/or refine bad hypotheses based on some alignment metric, and then the generated transcripts are used to iteratively refine the acoustic model [17, 23, 24]. Similarly, the subtitles themselves can also be used as training targets to gradually build an acoustic model on a larger corpus by iteratively refining the alignment [16, 18], which can be done with external models [25–27] or sophisticated preprocessing algorithms [17]. These methods fall under the first and most common approach of iterative data cleaning. Second, subtitles can be used to refine ASR outputs, either by training a biased (often program-based or genre dependent) language model on the subtitles [17, 23, 28], or with postprocessing techniques, e.g. to restore punctuation [29–31] or compress the transcript for optimised screen readability [32]. These methods, however, do not actually improve the ASR model.

### 2.2 Subtitles in speech translation

Most production-house movies and series are subtitled concurrently in many languages, leading to a big manual

[3] Code: https://github.com/nelfproject/NeLF_Transcription_ASR

[4] Models: https://huggingface.co/nelfproject

corpus suited for multilingual machine and speech translation [14]. Recent work [22] has investigated a dual decoder model to simultaneously produce an intralingual and a translated subtitle from the output of a pre-trained verbatim ASR model. To improve over general cascaded models [33], end-to-end speech translation models have also been proposed that directly produce a translated subtitle from speech, e.g. by regularising the encoder with a CTC loss for the intralingual subtitle and generating a translated subtitle with the decoder [34].

### 2.3 Written text generation

A similar strand of research considers the differences between written text and spoken text transcriptions [20, 35, 36]. In that case, spoken text is the verbatim transcription, and written text is an adapted version where disfluencies and fillers are removed and punctuation marks are added, which can then be used for natural language processing (NLP) applications [37, 38]. While subtitle generation is close to the task of written text transcription, there are many effects like rephrasings, summarisations and harsh subphrase deletions that are very specific to subtitles and not part of these written text annotations. Therefore, subtitles are a weaker form of supervision than written text, which is basically filtered and punctuated verbatim text. While many systems rely on post-hoc processing of the output of an ASR model for written text generation, using e.g. inverse text normalisation [39], disfluency and filler removal [40], and spelling correction [41], an end-to-end model that jointly learns to transcribe spoken and written text has been proposed which outperforms separate models and the cascaded approach [20], by leveraging a shared decoder that is conditioned on the task to perform.

### 2.4 Weakly supervised ASR

There has been a long-standing interest in improving acoustic modelling for ASR using weak supervision [23, 42, 43]. Typically, the weak labels are used to filter or improve the outputs of a pre-trained acoustic model, such that they can be used for ASR training [23], as with subtitles. The popularity of weakly supervised training has mainly been driven by the rise of very large datasets [44], which are often created by web-scraping audio sources as in Whisper [11] and forced-aligning transcriptions extracted from the web or ASR outputs [16, 45, 46].

### 2.5 Proposed approach

This work differs from previous efforts by jointly utilising the verbatim transcriptions from a standard ASR dataset and the subtitle transcriptions from a large dataset of broadcast media data, and carefully designing a model that can learn from and also generate both modalities. The model is trained completely end-to-end and the method does not require any preprocessing (e.g. filtering, pseudo-labeling or forced alignment) of the weakly labelled data, nor any postprocessing (e.g. an external program-based LM, inverse text normalisation), nor an iterative refinement of the acoustic models and/or data, although some of these techniques could still be applied. Finally, in this work, we do not focus on predicting line breaks and screen breaks [47], which can be done with segmentation models [48].

## 3 Methods

We propose several architectures to improve end-to-end verbatim ASR using subtitled data. We start from a strong conformer-based encoder-decoder ASR baseline. The following section details all proposed models, their differences and the integration of both verbatim and subtitled data.

### 3.1 End-to-end ASR

In end-to-end ASR, audio inputs are directly mapped to a transcription. It is customary to apply an encoder-decoder model to this sequence-to-sequence problem. The encoder extracts some informative features from the acoustic input, after which the decoder builds a sentence autoregressively, conditioned on the acoustic evidence in the encoder states using cross-attention.

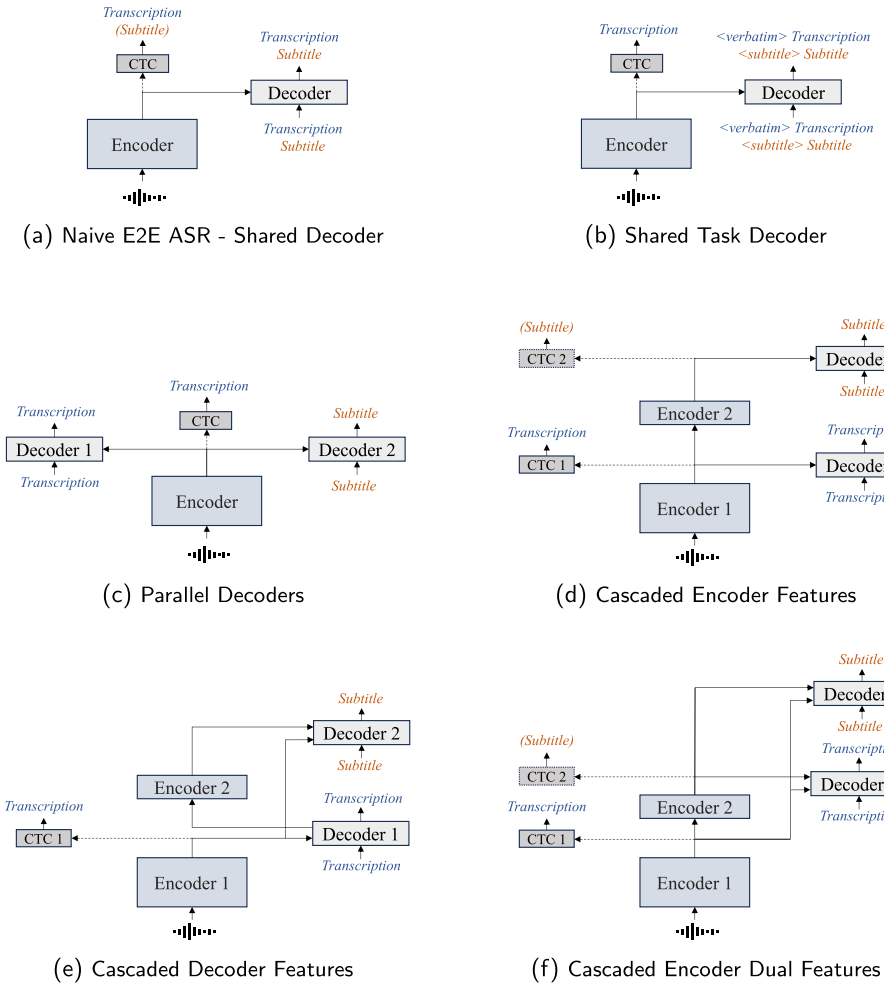
To regularise the encoder and guide it towards acoustically meaningful outputs, a connectionist temporal classification (CTC) [49] objective is imposed on its final outputs and on intermediate layer outputs [50].

The encoder-decoder hybrid CTC/Attention model [51] is optimised end-to-end, with a weighted sum of both objectives. The loss function in Eq. 1 is computed as a sum of the CTC regularisation loss (on the encoder's output $\mathcal{L}_{ctc}$ and an intermediate layer $\mathcal{L}_{interctc}$) and the cross-entropy classification loss $\mathcal{L}_{att,asr}$ of the decoder, weighted by the regularisation parameters $\alpha$ (CTC weight) and $\beta$ (inter-CTC weight).

$$\mathcal{L}_{asr} = (1-\alpha)\mathcal{L}_{att,asr} + \alpha[(1-\beta)\mathcal{L}_{ctc} + \beta\mathcal{L}_{interctc}] \tag{1}$$

During decoding, the model combines the CTC and attention-based output probabilities in a joint beam search.

A naive way to incorporate subtitles into the baseline E2E ASR model, is to treat the subtitles as regular verbatim annotations and directly use them standalone or mixed with verbatim ASR data. We call this approach *naive end-to-end ASR*. It is shown in Fig. 2a.




(a) Naive E2E ASR - Shared Decoder

(b) Shared Task Decoder

(c) Parallel Decoders

(d) Cascaded Encoder Features

(e) Cascaded Decoder Features

(f) Cascaded Encoder Dual Features

**Fig. 2** Comparison of all proposed models

### 3.2 Shared task decoder

There is an obvious mismatch in domains between verbatim ASR and the subtitling task. Inspired by recent advances in multi-task ASR decoder training [12], we extend the decoder to be able to perform both tasks while sharing the weights, by prepending a task token to the transcriptions, as proposed in [20] for dual spoken and written text transcription. For verbatim ASR data, the token *<verbatim>* is added before the start-of-sentence token in the transcription. For subtitle data, the token *<subtitle>* is used. The advantage of this model is that the decoder is optimised with all of the data, although it has to do more heavy lifting. Figure 2b presents the proposed method.

### 3.3 Parallel model

Subtitles can degrade the verbatim output predictions of the decoder when naively combining the verbatim transcripts and the subtitles. Moreover, the shared task decoder is prone to regime overfitting, where output styles align with input speech characteristics (verbatim vs. subtitle) rather than the intended task. A model with separate decoders for both tasks solves these issues, as we proposed in [19]. Both decoders attend to the encoder's output, but one is trained to generate a verbatim ASR transcription, while the other is trained to generate a subtitle transcription. The decoders work independently in parallel. By keeping the encoder shared, the model still enjoys the advantage of processing both data streams, and thus is adapted to both domains in a multi-task fashion [52]. Since the CTC objective assumes a monotonic alignment, and subtitles align very differently (due to e.g. rephrasing, compression and summarisation), the CTC loss is only incurred on the verbatim data.

During training, both data types are mixed and the network optimises a weighted combination of the verbatim ASR loss and the subtitling loss. The ASR loss $\mathcal{L}_{asr}$ from Eq. 1 is backpropagated over the verbatim data only,

masking the subtitle utterances in the verbatim decoder. The subtitle loss contains the subtitle decoder's attention loss $\mathcal{L}_{att,subs}$ and is backpropagated over the subtitle data (masking the verbatim utterances). The parameters $\lambda_{asr}$ and $\lambda_{subs}$ can be tuned to weigh the importance of both tasks during optimisation. The final loss is given in Eq. 2. The parallel model is depicted in Fig. 2c.

$$\mathcal{L}_{tot} = \lambda_{asr}\mathcal{L}_{asr} + \lambda_{subs}\mathcal{L}_{att,subs} \tag{2}$$

### 3.4 Cascaded model

Using a completely shared encoder could limit its capabilities, as the verbatim and subtitle objectives might counteract each other when attended to by the decoders. To remedy this, we introduce an additional (smaller) encoder, which we call the subtitle encoder, that should act as a translation block from verbatim to subtitle domain. The subtitle encoder, consisting of a few Transformer encoder layers, is cascaded to the outputs of the ASR encoder. The cascaded model is inspired by advances in end-to-end speech translation, where the model is decomposed in multiple encoder-decoder structures, each optimised for a given task and chained together [53, 54]. We will refer to the shared encoder (like the one in the parallel and naive methods) as the ASR encoder, since its output is used for the verbatim CTC objective and the verbatim ASR decoder. The subtitle encoder computes the features for the subtitle decoder. We investigate three possibilities for the cascaded model, which all incorporate the concept of a multi-transformer decoder.

#### 3.4.1 Multi-transformer

We define a multi-transformer decoder as a transformer block with multi-sequence attention, as proposed in previous work for speech translation [54] and multimodal translation [55]. In a regular transformer encoder-decoder model, there is one (multi-head) cross-attention layer in every decoder layer that attends to the encoder states. The multi-transformer decoder has two cross-attention layers per decoder layer, attending to different encoders, to leverage multiple information streams.

#### 3.4.2 Cascaded encoder features

The first approach for a cascaded model is depicted in Fig. 2d. In the first scheme, the outputs of the ASR encoder are directly fed as inputs to the subtitle encoder. Notice that the ASR encoder is regularised with CTC using verbatim labels (for verbatim data only). Then, the subtitle encoder transforms the acoustic verbatim features towards subtitle-like features. Finally, to not lose any temporal information in the subtitle encoder block, the subtitle decoder is augmented to a multi-transformer decoder. It can attend to the ASR encoder output and to the subtitle encoder output, effectively combining both streams to produce its best subtitle hypothesis.

Because of the translating subtitle encoder block, it is also possible to apply an additional CTC loss to the output of the subtitle encoder for subtitle data, although we found only limited benefits in doing this. Previous work [56, 57] has argued that a Transformer model with a CTC objective is able translate an input sequence to a non-monotonic alignment, despite the monotonicity assumptions in the CTC framework (e.g. in machine translation). Applying a subtitle CTC loss was not possible in the previous methods, as a shared CTC module for verbatim and subtitled data, or separate CTC objectives applied to the same encoder output, would lead to contradictory objectives for the encoder layers due to the different alignments between verbatim transcriptions and subtitles. Still, even without an additional CTC objective, the subtitle encoder block allows the model to translate verbatim features to subtitle features, without hampering the verbatim CTC and the verbatim ASR decoder objective.

#### 3.4.3 Cascaded decoder features

In the second scheme, shown in Fig. 2e, the final hidden states of the ASR decoder are used as input for the subtitle encoder. For subtitle data, there is no verbatim reference transcription such that backpropagation from the ASR decoder outputs is not possible during training. One possible solution would be to run the ASR decoder in inference mode (i.e. perform beam search) for subtitle data during training. However, this drastically slows down training since the parallelism is lost, and does not align well with the joint training approach proposed here. Another solution is to generate verbatim pseudo-labels for the subtitled data with a pre-trained ASR model (or with the current model after a certain number of epochs) and forward them through the ASR decoder. However, this is also not very scalable and is very slow in case of large subtitle datasets.

Based on these considerations, we use an alternative approach. If there is no reference verbatim transcription for the utterance (i.e. for subtitle utterances), we forward an $<unk>$ token through the decoder as verbatim reference (which is masked in the ASR decoder's loss) and let the verbatim decoder generate a sentence embedding for this token. We observed that this improves the optimisation of the ASR decoder. As during training all temporal acoustic detail is lost when using the decoder states as input, the subtitle decoder is again converted to a multi-transformer attending to both encoders' outputs, so that it can still access the temporal information in the ASR encoder. As such, the subtitle decoder can leverage the verbatim sentence embedding produced by the ASR decoder as well as the acoustic features produced by the

ASR encoder to generate a subtitle prediction.[5] Due to the sequence length reduction when going to verbatim labels, the subtitle CTC loss is not applied here.

#### 3.4.4 Cascaded encoder dual features

The third scheme, depicted in Fig. 2f, is similar to the first scheme, directly cascading the encoder blocks. Intuitively, the verbatim decoder might benefit from a proposed subtitle to generate a transcription. In this method, both decoders are multi-transformer decoders and attend to the two encoders' outputs. The amount of subtitle data is larger than the amount of verbatim data, but it is not directly backpropagated through the ASR decoder. Hence, the ASR decoder can now attend to the subtitle encoder, which is trained on all subtitle data, via the additional cross-attention layer.

#### 3.4.5 Loss function

The computation of the loss in the cascaded models is similar to the parallel model. The only difference is the possible inclusion of the subtitle-specific CTC loss $\mathcal{L}_{ctc,subs}$ with weight $\gamma$. The final objective function is detailed in Eq. 3.

$$\mathcal{L}_{tot} = \lambda_{asr}\mathcal{L}_{asr} + \lambda_{subs}\left[(1-\gamma)\mathcal{L}_{att,subs} + \gamma\mathcal{L}_{ctc,subs}\right] \tag{3}$$

## 4 Experimental setup

This section describes the details of the setup used in the experiments, including the model configurations, training aspects and the datasets.

### 4.1 Datasets

This paper proposes several methods to combine a verbatim annotated ASR dataset with a large collection of subtitled audio in the same language, i.e. Belgian Dutch (Flemish).

#### 4.1.1 Verbatim data

As a source of verbatim data, we use a standard speech recognition database called Corpus Gesproken Nederlands (CGN) or Spoken Dutch Corpus [58]. The Flemish part of the dataset contains 270 h of manually annotated speech recordings, divided over a multitude of components which each represent a specific type of speech and environment. Among the components, we can find (1) prepared, read speech by professional (news)readers, (2) recordings of interviews, lectures and meetings, (3) live sports commentary, (4) narrowband telephone conversations and (5) face-to-face discussions and conversational speech. We have derived a training set of 240 h of speech (350,000 utterances), which we call *cgn-train*. For the scaling experiments, where the ratio of subtitle data to verbatim data is increased, we use a 3-fold speed perturbed version of *cgn-train* with speed perturbation factors 0.9, 1.0 and 1.1 [59]. Furthermore, we use a representative test set of 8 h of speech (7000 utterances, 83,000 words) by sampling several recordings from every component, excluding telephone recordings, with a complete separation of speakers between train and test set. This test set, called *cgn-dev*, corresponds with previous work [19, 60]. Finally, we also created a verbatim test set *subs-annot* of 6 h from broadcast TV data, which is described in the next subsection.

#### 4.1.2 Subtitled data

We have compiled a large in-house dataset of subtitled data as broadcasted on Flemish TV. The subtitles and corresponding audio streams have been provided by the Flemish public broadcaster VRT. Due to copyright restrictions, this data cannot be released open-source. For the experiments, we use three data splits. The first split, used for analysis of the proposed models, contains 720 h of speech (915,000 utterances), identical to [19]. This split is a collection of audio streams from several topics, including TV talk shows with many different guests, broadcast TV news, live interviews, soap series, political talk shows, etc. For every recording, we have the corresponding subtitle as it appeared on screen. We have built a subtitle dataset by segmenting the audio recordings based on these screen timings. However, the screen timings are far from perfect, which makes the supervision a lot weaker as well. The subtitles have been normalised to some extent by filtering out music and non-speech (e.g. applause, ringtones) if they were annotated consistently (e.g. music is often tagged with an asterisk, non-speech audio events are reported in all uppercase). The formatting of such non-semantic elements is a common practice in subtitles, and hence easy to filter out, although not necessarily required (the model can also learn to predict these audio events). In a second stage, this dataset has been scaled up to 2000 h with recordings from the same or similar sources. This second split will be used to perform initial scaling experiments. In a final stage, we have built a dataset of 14,000 h of audio recordings with all kinds of content, either broadcasted on TV or online. Because of the scale of this third and final data split, we only do a few experiments on this dataset. The three training splits will be referred to as *subs-720h*, *subs-2kh* and *subs-14kh* respectively. For subtitle evaluation,

[5] We note that there is a mismatch between training and testing in terms of decoder states, but the subtitle decoder learns to focus on the first ASR decoder's hidden state (i.e. the sentence embedding). This approach has more potential in case iterative re-training with generated verbatim pseudo-labels is applied.

we retain two held-out data splits. The first one originates from the 2000 h of data, which we refer to as *subs-valid*, and consists of 11 h (110,000 words) of speech. The second one is a sample from the entire 14,000 h dataset, which we call *subs-valid-14kh*, and consists of 22 h (190,000 words) of subtitled data.

Lastly, we have created a verbatim test set called *subs-annot*, where we selected a representative subset of the TV shows in the *subs-720h* set, gathered new data from these shows and manually transcribed it *ad verbatim*. The *subs-annot* test set contains 6 h of transcribed speech (5000 utterances, 71,000 words) and corresponds with previous work [19].

### 4.2 Model details

#### 4.2.1 Input format

The audio recordings are converted to 16 kHz wav-files. We extract 80-dimensional mel-filterbanks and 3-dimensional pitch features using a window of 25 ms and a frame shift of 10 ms. The filterbank and pitch features are concatenated and mean-variance normalised at the utterance level. During training, SpecAugment [61] augmentations are applied to the input features.

#### 4.2.2 Model layout

All proposed models are modifications of the baseline encoder-decoder ASR model, implemented in ESPnet [62]. The ASR encoder is a Conformer [2], preceded by a Conv2d input layer which subsamples the input features four-fold and transforms them to the hidden dimension of 256. Relative positional encodings are added to the inputs and used in the self-attention Conformer layers. The Conformer encoder has 12 layers in macaron-style with Swish activations. Every layer has 4 attention heads, the feed-forward layer dimension is 2048 and the CNN has a kernel size of 31. The decoders are 6-layer Transformer decoders, with 4 attention heads, 2048 feed-forward units and the same hidden dimension as the encoder. Dropout of 0.1 is applied at all layers. These hyperparameters arise from standard recipes for ASR models [62] and have been deducted based on optimal performance of the baseline ASR model. In case the model implements an additional subtitle encoder, it consists of 2 regular Transformer layers with the same dimensions as the ASR Conformer encoder layers. For the experiments on the *subs-2kh* and *subs-14kh* data, the subtitle encoder consists of 6 Transformer layers. For the scaling experiments in Sect. 5.3.2, we train a larger variant, denoted *XL-model*, with a hidden dimension of 512 and 8 attention heads per layer. The base variants contain 70 M parameters and the XL variant has 180 M parameters. The baseline ASR model has 50 M parameters.

#### 4.2.3 Training

The models are trained with a joint hybrid CTC/Attention loss [51], with CTC weight $\alpha = 0.3$ applied to the encoder outputs for the verbatim targets only. Intermediate CTC [50] is applied to layer 6 of the ASR encoder, with a weight of $\beta = 0.3$. When indicated, subtitle CTC loss is applied with $\gamma = 0.3$. During multi-task training, a batch consists of an equal amount of utterances from the verbatim and subtitle datasets. If the subtitle dataset is larger than the verbatim dataset, the verbatim utterances are oversampled in an epoch (except in the naive method). The transcripts are tokenised into byte-pair encoding (BPE) (sub)word units. The BPE model has a vocabulary of 5000 unigram BPE's and is trained on both the verbatim transcriptions and an equally sized sample of the subtitles, such that the verbatim and subtitle transcriptions can share the same BPE token space. All transcripts are normalised and lower-cased without punctuation. The decoder predicts the target BPE's which are smoothed with a label smoothing weight of 0.1. By default, the prediction losses of the ASR encoder and the subtitle decoder are weighted with an equal weight $\lambda_{asr} = \lambda_{subs} = 0.5$, which was found optimal in previous work [19] when mixing batches equally with utterances from both data streams.

The models are trained on a single GPU (24 GB), with an effective batch size of 1024 and the Adam optimiser [63] with an exponential decaying learning rate with a peak of 0.004 and 25,000 linear warmup steps. The models are trained until convergence with a maximum of 150 epochs for *subs-720h* and 30 epochs for *subs-2kh*. The 10 best intermediate training checkpoints with the highest validation accuracy are averaged for evaluation. For the large-scale experiments on *subs-14kh*, the models are trained for 5 epochs with a learning rate of 0.001 and 100,000 warmup steps on 16 GPU's (40GB) by accumulating gradients over GPU's.

#### 4.2.4 Decoding

During inference, the models generate a verbatim transcription and a subtitle in parallel. The decoder integrates the CTC prefix-scores with a weight of 0.3. The best hypotheses for both decoders are computed with a beam search keeping the 20 best running hypotheses. No language model is applied during decoding.

### 4.3 Evaluation

The verbatim transcription outputs of the models are evaluated based on the word error rate (WER) with respect to the reference verbatim transcripts. The subtitle outputs are evaluated based on a BLEU score [64, 65] with respect to the real subtitles on screen, such that

differences in sentence ordering are punished less. We use a smoothed BLEU-4 score with uniform weights for all n-grams ($n = 1..4$).

The statistical significance of the results is analysed by computing the $p$-values between hypotheses from different models. For ASR results (WER), the MAPSSWE [66] test from the NIST Scoring Toolkit (SCTK)[6] is used. For subtitling results (BLEU), we conduct paired tests with bootstrap resampling ($n = 1000$ fold) [67] with the SacreBLEU[7] toolkit. All results from different models (on the same test set) that are *statistically insignificant* from each other with $p > 0.05$ are marked with the same tag († or ‡) in tables. If more than two results in a table have the same tag, all pairwise comparisons are also statistically insignificant.

## 5 Experiments

In this section, we perform several experiments and compare the proposed methods to established baseline models. Both the verbatim transcription and subtitle generation capabilities of the models are assessed on different benchmarks. Furthermore, we analyse the effect of the double cross-attention in the multi-transformer decoder layers and the different output modalities of the verbatim and subtitle decoder, and compare our methods to the standard approach of data filtering based on the reference subtitles. Moreover, we perform a scaling experiment on a large subtitle dataset using long-form punctuated transcriptions. Finally, we compare our proposed model to the related state-of-the-art Whisper [11] speech model. We also evaluate whether LLMs can be used to generate subtitles from ASR outputs and compare this approach to our end-to-end model.

### 5.1 Performance comparison

#### 5.1.1 Verbatim speech recognition

First, the proposed models are evaluated on a verbatim speech recognition task. All models are trained on two subtitle datasets, consisting of 720 h (*subs-720h*) and 2000 h (*subs-2kh*) of speech respectively, combined with the verbatim CGN dataset (*cgn-train*). All models are evaluated on *cgn-dev* and *subs-annot*, by comparing the outputs of the verbatim ASR decoders to the reference verbatim transcriptions. Results are shown in Table 1.

Naively using subtitles as verbatim transcriptions leads to a degradation on the ASR dataset *cgn-dev*, and even diverges when the subtitle-to-verbatim ratio is too high. As the proposed models are able to distinguish between the two domains, they don't suffer from these drawbacks.

**Table 1** WERs (%) of verbatim speech recognition experiments (↓). The models are trained using the verbatim data from CGN and either 720 h or 2000 h subtitle data. They are evaluated on *cgn-dev* and *subs-annot*

| | *Train* | *720 h subtitles* | | *2000 h subtitles* | |
|---|---|---|---|---|---|
| **Model** | ***Test*** | ***cgn-dev*** | ***subs-annot*** | ***cgn-dev*** | ***subs-annot*** |
| E2E ASR - CGN only | | 10.71 | 14.06 | 10.71 | 14.06 |
| Naive E2E ASR | | 13.74 | 30.34 | 97.67 | 94.17 |
| Shared task decoder | | 9.38 | 20.99 | 8.51 | 16.75 |
| Parallel decoders | | 10.01 | 10.76 | 8.84 | 10.07 |
| Cascaded encoder | | **8.78** | 9.93† | 8.27† | 9.65 |
| Cascaded decoder | | 8.94† | 10.09† | **8.26**† | 9.33† |
| Cascaded Enc. Dual | | 8.99† | **9.89**† | 8.29† | **9.26**† |

The parallel decoder model benefits from domain adaptation in the shared encoder yielding substantial gains on the *subs-annot* set, which is adopted from TV series, but lacks strong improvements on *cgn-dev* with less data. The shared task-decoder model shows nice improvements on *cgn-dev*, benefiting from the additional data to train the decoder. However, it performs worse on *subs-annot*. It is likely that the decoder has learned the difference between the two training datasets, and therefore outputs more subtitle-like transcripts for the *subs-annot* set, despite being conditioned on the task token. This is a phenomenon also reported by other works involving multi-task decoders [12], which seems to be a general drawback of that approach. Finally, the proposed cascaded models report the lowest WERs, in both data regimes. The cascaded encoder model is essentially the same as the parallel model, except for the additional subtitle encoder block, which gives more freedom to the model to encode the verbatim and subtitle targets differently without hampering the CTC objective applied to the ASR encoder. The cascaded model with dual encoder features performs best on the *subs-annot* broadcast TV test set, as the verbatim decoder is able to attend to both the ASR encoder and the subtitle encoder outputs. The cascaded models result in a 20 to 30% relative improvement on the verbatim speech recognition task, without any additional verbatim data or filtering methods. In addition, they produce a dual subtitle output, evaluated in the next section.

#### 5.1.2 Automatic subtitling

Second, the subtitle outputs of the models are evaluated with a BLEU score with respect to the reference on-screen subtitle. To this end, for cascaded and parallel models, the output of the subtitle decoder is used. For the task-specific model, the decoder is conditioned on the subtitle task token. These are the same models as in

[6] https://github.com/usnistgov/SCTK

[7] https://github.com/mjpost/sacrebleu

**Table 2** BLEU scores (%) of subtitle recognition experiments (↑). The models are trained using the verbatim data from CGN and either 720 h or 2000 h subtitle data. They are evaluated on *subs-valid*. For the baseline ASR model and the naive E2E ASR model, the predictions of the ASR decoder are scored

| Model | *Train* / *Test* | *720 h subtitles* *subs-valid* | *2000 h subtitles* *subs-valid* |
|---|---|---|---|
| E2E ASR - CGN only | | 29.88 | 29.88 |
| Naive E2E ASR | | 51.05 | 33.32 |
| Shared task decoder | | 50.63$^{\dagger}$ | 52.56 |
| Parallel decoders | | 46.16 | 46.35 |
| Cascaded encoder | | **51.17** | **54.14**$^{\dagger}$ |
| Cascaded decoder | | 50.67$^{\dagger}$ | 53.95$^{\dagger}$ |
| Cascaded Enc. Dual | | 50.58$^{\dagger}$ | 54.06$^{\dagger}$ |

Table 1, trained using either 720 h or 2000 h of subtitled data. All multi-task models are evaluated on *subs-valid*, and the BLEU scores are reported in Table 2.

The parallel model is not able to generate strong subtitles, compared to the shared task decoder and the cascaded models. In most cases, the cascaded models produce the highest quality subtitles with very promising BLEU scores. The cascaded encoder model slightly has the edge over the other architectures.

### 5.2 Analysis and ablation

#### *5.2.1 Multi-transformer decoder*

We investigate the quantitative and qualitative effect of the proposed multi-transformer decoder. The cascaded encoder model (Fig. 2d) and cascaded decoder model (Fig. 2e), which have a Multi-Transformer subtitle decoder, are compared to their respective variants with regular Transformer subtitle decoders without the additional cross-attention (i.e. in that case, the arrow from Encoder 1 to Decoder 2 is dropped in Fig. 2). Figure 3 depicts the WER and BLEU scores for this ablation study. All differences are statistically significant except for the encoder results in Fig. 3a.

The additional cross-attention to the ASR encoder's output in the multi-transformer decoder leads to an improvement over the single cross-attention Transformer decoder. For the cascaded decoder model, the ASR encoder outputs are necessary for fine-grained temporal detail which is lost in the decoder features that are fed to the subtitle encoder, leading to expected improvements. The cascaded encoder model enjoys notable improvements on *subs-annot*, as the additional cross-attention leads to a stronger backpropagation through the ASR encoder for the subtitled data.

#### *5.2.2 Dual outputs—translation effects*

This section gives some additional information about the differences between subtitles and verbatim transcripts in our dataset. In the last decades, the predominant spoken language in Flanders (the Northern, Dutch-speaking part of Belgium) has been changing towards a variant called "*tussentaal*" or "intermediate language", which is structurally positioned in between the local Flemish dialects and the Dutch standard language [68, 69]. In contrast to Standard Belgian Dutch (SBD), the official written language, this variant is called Colloquial Belgian Dutch (CBD): an all-encompassing term for the multiple regiolects in Flanders, which are strongly subjected to regional and social variation [70]. While Standard Dutch remains the norm in formal domains and non-fictional informational television (e.g. documentaries), informal speech is shifting from local dialects to regiolects [71] and has become a public medium, omnipresent in less-informational TV programs [68]. CBD has several key features that differentiate it from SBD, which can mainly be categorised into lexical (e.g. "*appelsien*" instead of "*sinaasappel*" both meaning the fruit "orange"), morphological (e.g. diminutive "*-ke*" instead of "*-je*", personal pronoun "*ge*/*gij*" instead of "*je*/*jij*" meaning "you") and syntactic features (e.g. using a double negation like "don't know nothing") [70, 72]. These could be compared to colloquialisms and slang in English (e.g. "bloody knackered" instead of "very exhausted" in British English, "ain't" instead of "isn't" in American English), as well as contractions used in informal speech (e.g. "gonna/going to", "wanna/want to", "kinda/kind of" and "innit/isn't it"). Generally, regiolectal speech is not transcribed literally on Flemish television, but according to guidelines converted into SBD written form [69]. Especially CBD features that are closer to dialect are more often converted to SBD in subtitles [70].

This linguistic discrepancy between spoken and written language is reflected in the differences between the outputs of the subtitle decoder and the verbatim decoder. The subtitle decoder maps the spoken dialectal language towards its Standard Dutch version, sometimes with complete rephrasing, while the verbatim decoder does not make these conversions. Furthermore, the subtitle decoder corrects hesitations, repetitions and other disfluencies, as well as typical abbreviations in spontaneous speech, which are less common in written text. In Appendix, we have included several examples that demonstrate these differences.

#### *5.2.3 Data filtering*

The common practice applied in most efforts [17, 23, 24] leverages subtitles to filter the hypotheses of a pre-trained ASR model, such that the remaining subtitles (or hypotheses) can be used as reference verbatim transcriptions to

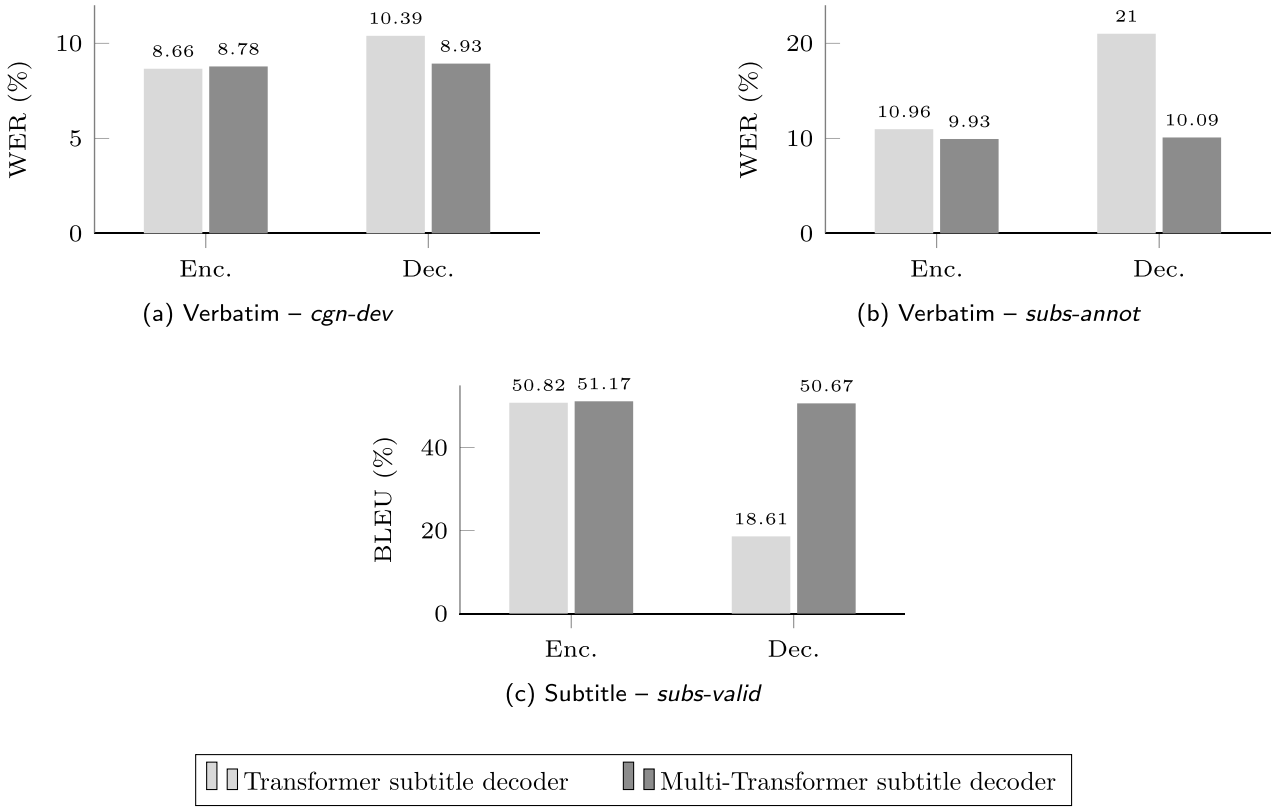



**Fig. 3** Comparison between subtitle decoder blocks with only one cross-attention to the output of the subtitle encoder ("Transformer"), or with two cross-attentions, i.e. once to the output of the ASR encoder and once to the output of the subtitle encoder ("Multi-Transformer"). The subtitle encoder is either conditioned on the ASR encoder ("Enc.") or decoder ("Dec.") features

train an improved ASR model. In our approach, such filtering methods are not required, as the models can distinguish between verbatim and subtitle labels, but they can still be applied complementary to the proposed training pipeline. To remove subtitles which are too different from the uttered sentence (e.g. misaligned with bad timings, harsh deletions, completely different), we perform some experiments with automated data filtering. To this end, a segment is removed if the verbatim prediction of a pre-trained ASR model is inconsistent with the on-screen subtitle. The pre-trained ASR model is the baseline from Table 1, trained on *cgn-train*. The BLEU scores between the hypotheses and the subtitles are computed, and the segments with a BLEU score below the filtering threshold are dropped. A new ASR model (standard or cascaded) is then trained on the combination of the filtered dataset and *cgn-train*. The effects of filtering on dataset size and WER are shown in Table 3. We compare our cascaded encoder model, which leverages two Transformer decoders as detailed in Sect. 3.4.2, to the naive approach which uses a single E2E model. For a fairer comparison to the naive approach in terms of data filtering, the cascaded encoder model is trained with the inclusion of the subtitle CTC loss induced on the subtitle encoder output. This additional loss is the only difference with the cascaded encoder model results in Table 1.

**Table 3** Data filtering experiments for verbatim speech recognition, evaluated in terms of WER (↓) on *cgn-dev* and *subs-annot*

| Model | Subtitle dataset | Filtering BLEU | Remaining subtitles | WER (%) *cgn-dev* | WER (%) *subs-annot* |
|---|---|---|---|---|---|
| Naive E2E ASR | *none* | 0% | 0 h | 10.71 | 14.06 |
| | *subs-720h* | 0% | 720 h | 13.74 | 30.34 |
| | *subs-2kh* | 60% | 270 h | 12.10 | 11.11 |
| Cascaded Encoder | *subs-720h* | 0% | 720 h | 8.80 | **10.39** |
| | | 10% | 480 h | 9.37 | 10.69 |
| | *subs-2kh* | 0% | 2000 h | 8.90† | 11.56 |
| | | 10% | 1200 h | **8.74**† | 11.16 |
| | | 60% | 270 h | 9.43 | 11.86 |

For the naive approach, where there is no explicit difference between subtitles and verbatim transcriptions, strongly filtering out divergent subtitles is beneficial for

**Table 4** WERs (%) of verbatim speech recognition experiments (↓) with long-form serialised output training. The models are evaluated on the short-form *cgn-dev* and *subs-annot*, and the long-form *cgn-dev-long* and *subs-annot-long*

| Model | *cgn-dev* | *cgn-dev-long* | *subs-annot* | *subs-annot-long* |
|---|---|---|---|---|
| E2E ASR - CGN only (S) | 10.75$^{\dagger}$ | 15.12 | 14.06 | 16.85$^{\ddagger}$ |
| E2E ASR - CGN only | 13.03$^{\ddagger}$ | 10.61 | 15.05 | 14.17 |
| Shared task decoder | 13.34$^{\ddagger}$ | 12.32 | 18.42 | 17.86$^{\ddagger}$ |
| Parallel decoders | 13.22$^{\ddagger}$ | 10.75 | 13.00 | 13.13 |
| Cascaded encoder | 11.09$^{\dagger}$ | 9.09$^{\dagger}$ | 11.59$^{\dagger}$ | 11.64$^{\dagger}$ |
| Cascaded decoder | 11.14$^{\dagger}$ | 9.09$^{\dagger}$ | 11.30$^{\dagger}$ | 11.40$^{\dagger}$ |
| Cascaded Enc. Dual | **11.00**$^{\dagger}$ | **8.86** | **10.80** | **10.84** |

the model. However, based on the results in Table 3, this standard filtering approach does not reach the same performance as the cascaded models, as lots of invaluable speech is thrown out. For the cascaded model, some filtering can still be useful to throw out the misaligned speech segments with incorrect timestamps. Compared to Table 1, we do not see improvements on the *subs-annot* test set with more data in Table 3 and slightly worse results, which is a result of the additional subtitle CTC loss during training.

Data filtering with the cascaded architecture is not mandatory as the subtitles and verbatim transcriptions are nicely separated, and only useful feedback from the subtitles is backpropagated to improve the ASR modelling. Hence, our approach allows to conveniently scale up the weak supervision. As mentioned in the introduction, the fact that our method does not require such filtering is a strong advantage over the common practice of intensive data cleaning, especially regarding the incorporation of heavily accented and dialect speech. As will be discussed in Sects. 5.3.2 and 6, scaling up the subtitle data and incorporating more dialectal and accented speech results in models that are strong against local dialects.

### 5.3 Scaling experiments

#### *5.3.1 Long-form punctuated training*

To build a powerful model of high performance that is readily deployable for automatic transcription services, we extend the capabilities of the decoders by enriching the target transcriptions. To this end, we add complete punctuation and capitalisation instead of normalising the transcriptions (which was done in the previous experiments), so that the model can generate a fully formatted transcription. For the verbatim transcriptions, we also add transcription tags (e.g. $<*a>$, $<*v>$) to some words indicating when a word is cut off mid-word, when it is a foreign or dialect word, etc. Those tags are part of the rich manual annotations of the verbatim dataset [58].

Furthermore, we noticed that an utterance in the dataset is on average only 3 s long. Training a baseline model on these short utterances as in previous experiments (Table 1), leads to a WER of 10.75 % on the *cgn-dev* test set of equally short utterances (4 s on average), but degrades to 15.12 % when evaluating on the same test set but with concatenated utterances of 10 seconds on average. We denote this test set *cgn-dev-long*. To remedy this effect, we combine short consecutive utterances in the training dataset, so that the utterance durations are more uniformly distributed with an average duration of 9 s and a maximum of 15 s. Both the verbatim and subtitle dataset are transformed into a long-form equivalent by concatenating utterances. Additionally, when two consecutive utterances come from different speakers, we add a speaker change token $<spk>$ between the utterances, following serialised output training techniques [21, 73, 74]. For the subtitle dataset, this is based on the colour of the subtitle (which should switch when there are different speakers). This pushes the model to learn acoustically when different people talk within one segment, without requiring corpus-level speaker information. Analogous to *cgn-dev-long*, we create a long-form equivalent of *subs-annot* (4 s on average) and denote it *subs-annot-long* (9 s on average).

Table 4 shows the results of a first experiment with long-form data and serialised output training. The models are trained on a subset of 20% of the 2000 h subtitle dataset from previous experiments (in order to reduce the computational load in these initial experiments) and evaluated on the short-form test sets *cgn-dev* and *subs-annot*, and their longer form equivalents *cgn-dev-long* and *subs-annot-long*. The first row shows the results of an ASR model trained on short-form and normalised data, the other results are from long-form models. Additionally, Table 5 shows the BLEU scores of the evaluated models on the short form *subs-valid* set, and on the longer form *subs-valid-14kh* set, which is also more difficult as it covers a wider range of TV shows and types of

**Table 5** BLEU scores (%) of subtitle prediction experiments (↑) with long-form serialised output training. The models are evaluated on *subs-valid* and *subs-valid-14kh*. For the E2E ASR model, the output of the (verbatim) ASR decoder is used. For the other models, the predictions of the subtitle decoder are used

| Model | *subs-valid* | *subs-valid-14kh* |
|---|---|---|
| E2E ASR - CGN only (S) | 29.88 | 33.71 |
| E2E ASR - CGN only | 32.01 | 34.06 |
| Shared task decoder | 39.27$^{\dagger}$ | **45.12** |
| Parallel decoders | 35.51 | 37.02 |
| Cascade encoder | **39.83** | 42.95 |
| Cascade decoder | 39.18$^{\dagger}$ | 43.36$^{\dagger}$ |
| Cascade Enc. Dual | 39.41$^{\dagger}$ | 43.51$^{\dagger}$ |

speech. The additional tags (verbatim tags, punctuation, speaker change) are neglected during WER and BLEU calculation.

The trends observed for long-form training are similar to previous training setups. Note that compared to Tables 1 and 2, less subtitle data is used, and the decoder does a lot more (tag prediction, punctuation, etc.). For verbatim ASR, the cascaded models outperform the other approaches, as in the short-form setting. The cascaded model with dual encoder features significantly outperforms the other methods for most test sets. For subtitling, in general the same trends across models are visible. The BLEU scores on the short-form *subs-valid* in Table 5 are lower than in previous experiments, due to less subtitled training data used and the short-form effects observed in Table 4. For practical use of ASR models, this long-form training method is more robust to long utterances and multi-speaker fragments, while generating a fully formatted output including punctuation, capitalisation and anomaly tagging.

#### *5.3.2 Large scale modelling*

The cascaded model with dual encoder features from Fig. 2f, which performed best in Table 4, is used to perform a scaling experiment on the large subtitle dataset of 14,000 h *subs-14kh*, with enriched combined utterance transcriptions. We generate 4 different subsets of the dataset with an increasing amount of subtitle data. The verbatim data is in every experiment a 3-fold speed perturbed version of the verbatim CGN dataset. Figure 4 shows the verbatim results in terms of WER on the short-form *cgn-dev* and *subs-annot* and the long-form *cgn-dev-long* and *subs-annot-long*, and the subtitling results in terms of BLEU score on *subs-valid* and *subs-valid-14kh*. The *XL-model* in Fig. 4 is a larger version of the model as described in Sect. 4.2.2.

All figures indicate that the inclusion of more subtitled data leads to an improved performance, showing that the proposed method is scalable. For the verbatim speech recognition experiments, we note unseen scores with up to 50% relative WER reduction compared to the baseline ASR model on the test sets, without adding any additional verbatim data to the verbatim ASR decoder, which can be seen on Fig. 4a, b. For large-scale data, the larger *XL-model* (i.e. the right-most column in Fig. 4) is better able to learn all variations in the data. As expected, the achieved WER reductions do not scale linearly with the amount of weakly supervised data (note that this is still a relatively small model). However, models with larger capacity and training time would probably reach even higher returns. Finally, Fig. 4c shows the impressive subtitle generation capabilities of the proposed model. We remark that the *subs-valid-14kh* evaluation set is a very difficult set with random held-out samples from the entire 14,000 h corpus, containing a very broad spectrum of speech types and dialects. The cascaded model trained on *subs-14kh* exhibits very high BLEU scores, which can be interpreted as strong translations.

#### *5.3.3 Comparison to whisper*

Finally, we compare our proposed model to the state-of-the-art multilingual speech recognition model Whisper [11]. Whisper is an encoder-decoder multi-task speech model that is able to transcribe speech as well as translate speech to English text. It is trained on 680,000 h (or more, for *Whisper-large-v3*) of weakly labelled speech data. Since Whisper is trained on long-form audio (30-s chunks), for a fair comparison we evaluate on the long-form test sets. Table 6 shows the WERs on *cgn-dev-long* and *subs-annot-long*, comparing our proposed method to Whisper. The first row corresponds to the baseline encoder-decoder ASR model trained on long-form supervised data from the CGN dataset only (second row of Table 4). For our proposed method, we use the cascaded model with dual encoder features that was trained on the combination of CGN and 14,000 h of subtitle data from Sect. 5.3.2. We include both the base and XL model sizes from Fig. 4. We compare these against two baselines: the off-the-shelf *Whisper-large-v3*[8] model, and a version of *Whisper-large-v2* finetuned on the CGN dataset[9].

The results in Table 6 show the benefits of the proposed approach and the synergistic effect of using subtitles in the same language on the resulting verbatim speech recognition performance. While Whisper is trained on a huge amount of speech data, it lacks the

[8] https://huggingface.co/openai/whisper-large-v3

[9] https://huggingface.co/kul-speech-lab/whisper_large_CGN

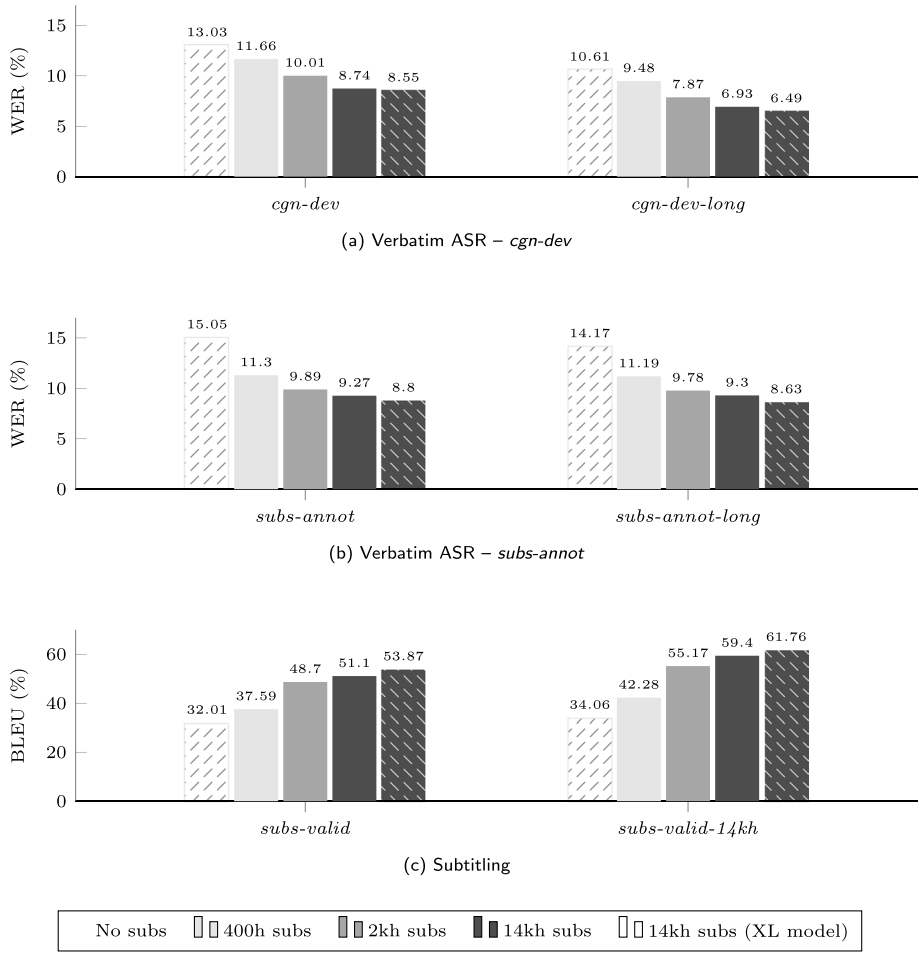



**Fig. 4** Scaling experiments for the cascaded model with dual encoder features. On the horizontal axis, increasing amounts of subtitled training data are used. **a** shows WERs (↓) of the verbatim ASR decoder outputs with respect to the reference verbatim transcription. **b** shows BLEU scores (↑) of the subtitle decoder outputs with respect to the reference subtitles. All pairwise comparisons between the results are statistically significant

**Table 6** WERs (%) of verbatim speech recognition experiments (↓) with care evaluated on *cgn-dev-long* and *subs-annot-long*. The second column denotes the number of parameters in every model

| Model | # Params. | *cgn-dev-long* | *subs-annot-long* |
|---|---|---|---|
| E2E ASR | 50M | 10.61 | 14.17 |
| Proposed Model | 70M | 6.93 | 9.30 |
| Proposed Model (XL) | 180M | **6.49** | **8.63** |
| Whisper large | 1550M | 11.54 | 13.76 |
| Whisper large finetuned | 1550M | 7.83 | 10.64 |

distinction between verbatim transcripts and subtitle transcripts, and is outperformed by our method in this verbatim speech recognition task. Even after finetuning the high-capacity Whisper model on verbatim Dutch data, our proposed model reaches better WERs, as a result of the subtitle data. If the Whisper model were also to be finetuned on the entire subtitle dataset (for which we do not have the required resources), its output would probably differ strongly from verbatim transcripts, leading to high WERs on this task.

Moreover, our proposed model is very efficient in terms of parameters and compute resources compared to the large Whisper model, using only about 10% of its parameters. In experiments not reported here, we found that smaller Whisper models perform much worse.

In addition to producing verbatim transcripts, our methods are able to generate subtitles at the same time. We have analysed whether cascading LLMs to ASR outputs leads to subtitles of similar quality as our proposed methods, which are able to condition on the audio as well. We found that the produced subtitles with the joint model outperforms a general ASR+LLM pipeline. Detailed results can be found in 1.

## 6 Discussion

The experiments in Sect. 5 have shown that the proposed methods are able to leverage subtitle data to improve ASR models, through disambiguation between the two domains. In most cases, the cascaded models outperform the other approaches. As Table 4 illustrates, the cascaded model with dual encoder features generally yields the best ASR performance. There seems to be no harm in combining a large body of subtitle data, including potentially noisy or dialectal samples which are often filtered by other methods, with a small set of verbatim data. Moreover, the subtitles bring about a positive learning effect which even enhances the verbatim transcription, as the encoder is improved. The proposed approach offers up to 20% relative improvement on in-domain test data using only 720 h of subtitled data, which can be increased to even 50% when using more data. On top of that, the domain mismatch between the speech in the ASR dataset and the (often more spontaneous) speech on TV is relieved.

Furthermore, the proposed models are able to generate subtitles that contain many translational effects with regards to written text (Standard Dutch). Based on extensive use of our model, we notice that even if the verbatim transcription on an out-of-domain sample is far from perfect, often due to a strong local dialect, the subtitle decoder is in most cases able to grasp the content fairly well, as it can be trained more easily on lots of data. Moreover, the subtitle decoder tends to be very strong at recognising rare words, such as specific terms and proper names. This can be attributed to the large vocabulary that is modelled in the subtitle decoder by leveraging the large body of subtitle data. Despite the general improvements across the board, it is very difficult to also perfect the recognition accuracy of the verbatim decoder for these rare words and very strong dialects in our proposed architectures without parallel data or contextualisation methods.

Additionally, the experiments with large-scale data have shown very good results on a broad domain, with a performance that is suitable for automatic subtitling. Reaching optimal performance on all local variations probably requires more balancing of the dataset with respect to local dialects and overrepresented speakers when the data, and therefore also the required computing resources, become very large.

Finally, as the proposed models generate both a verbatim transcription and a subtitle, they can be used for multiple applications. Depending on the use case, either the verbatim transcript (e.g. for speaker analysis, note taking) or the subtitle (e.g. for multimedia) can be useful. For NLP applications, the subtitle outputs should also work better as they are closer to standard written text than verbatim ASR outputs. While validation on other languages than Flemish would further strengthen these findings, the proposed architectures separating verbatim and subtitle data streams are rather language-agnostic (i.e. no assumptions are made about language-specific aspects), suggesting potential for generalisation to other scenarios where similar weakly supervised subtitle data is available, especially in low-resource settings. Moreover, general discrepancies between colloquial spoken language and standardised written text or subtitles can be found in many languages [35, 75, 76].

## 7 Conclusion and future work

We proposed several architectures to improve automatic speech recognition with weakly supervised data in the form of TV subtitles. The proposed models are able to generate both a verbatim transcription and a subtitle for a spoken utterance. A cascaded encoder approach with separate decoders that attend to both encoder outputs shows a strong learning effect from the subtitle data to the verbatim branch. Evaluation has shown improvements on both verbatim ASR and on automatic subtitling of broadcast TV shows. We carried out scaling experiments on a large subtitle database to prove the scalability of the methods. This work has resulted in a new state-of-the-art ASR model for Belgian Dutch, a language with only a limited amount of manually annotated verbatim training data. Moreover, the proposed approach provides a general architecture that can cope with approximate, weakly supervised transcripts. The analysis conducted on Flemish serves as a compelling proof-of-concept, highlighting the potential advantages of applying our methods to other languages, which is a promising avenue to explore and which might result in strong benefits especially for low-resource languages, given that there are subtitled data sources available.

For future work, we are interested in investigating verbatim pseudo-labeling approaches to generate parallel data to learn even stronger connections between the different modalities. The strengths of the subtitle decoder in recognising rare words and strong dialectal speech could be transferred to the pseudo-labels through contextualisation methods. Furthermore, as there are many datasets containing subtitle texts (e.g. from open-source movie subtitles), a language model can be built with these sources to improve the final transcriptions. It might also be interesting to incorporate timestamp prediction in the current framework to predict subtitle timings on the fly. Moreover, a multilingual dual model can be trained by leveraging subtitles and ASR datasets from many languages. Finally, future work could evaluate the impact of weakly supervised transcripts for ASR models that consist of a combination of a speech encoder and an LLM decoder.

**Table 7** Comparison of the reference (REF) verbatim and subtitle transcriptions to the hypotheses (HYP) generated by the outputs of the verbatim ASR decoder and the subtitle decoder. Translation in English is added for the reader in italics

| | |
|---|---|
| **REF** (verbatim) | Wacht wacht wacht momentje. Zijt de<∗*d*> zeker dat ze d'r niet is? Misschien moet ge eerst efkes checken. |
| | *Wait wait wait a moment. Are you sure that she's not there? Maybe you just have to check first.* |
| **REF** (subtitle) | Wacht, wacht. Ben je zeker dat ze er niet is? Misschien moet je eerst checken. |
| | *Wait, wait. Are you sure that she is not there? Maybe you have to check first.* |
| **HYP** (verbatim) | Wacht wacht wacht. Zijt de<∗*d*> zeker dat ze d'r niet is? Misschien moet eerst effekes checken? |
| | *Wait wait wait. Are you sure that she's not there? Maybe you just have to check first?* |
| **HYP** (subtitle) | Wacht. Ben je zeker dat ze er niet is? Check eerst even. |
| | *Wait. Are you sure that she is not there? Just check first.* |
| (1) Example 1: Spontaneous conversation from a soap series | |
| **REF** (verbatim) | We hadden gehoopt eigenlijk na die die tweede dosis we zagen dat er goeie bescherming was dat dat zou blijven duren. |
| | *We had hoped actually after that that second dose we saw that there was good protection that that would last.* |
| **REF** (subtitle) | Wij hadden gehoopt dat het na de tweede doses zou blijven duren. |
| | *We had hoped that it would last after the second doses.* |
| **HYP** (verbatim) | En we hadden gehoopt eigenlijk na die die tweede dosis. We zagen dat de goeie bescherming was dat dat zou blijven duren. |
| | *And we had hoped actually after that that second dose. We saw that the good protection was that that would last.* |
| **HYP** (subtitle) | We hadden gehoopt na die tweede dosis dat dat zou blijven duren. |
| | *We had hoped after that second dose that it would last.* |
| (2) Example 2: Live conversation with a virologist in a talk show | |

## Appendix 1 Dual outputs: verbatim and subtitle annotations

### Examples of differences between verbatim and subtitle transcripts

Section 5.2.2 has explained the general differences between Dutch spoken and written language as well as the differences between verbatim and subtitle annotations. To visualise the dissimilarity and their consequences in practice, Table 7 shows several examples for speech samples from the *subs-valid-14kh* test set. The examples depict the reference subtitle and a manual verbatim transcription, as well as the predictions from the cascaded model with dual encoder features from Table 4 (using either the subtitle decoder output or the verbatim ASR decoder output respectively). Notice that our model produces both outputs at the same time.

The examples in Table 7 show several of the effects described in Sect. 5.2.2. In Example 1, the colloquial pronoun "*ge*", common in spoken language, is translated to "*je*" in the subtitle. The expression "*zijt de*" ("are you") is a dialectal form of "*zijt ge*", which is generally written as "*ben je*". Notice that it is tagged with <∗*d*>, and these tags are also predicted by the verbatim decoder. The colloquial word "*efkes*" ("shortly") is translated to its standard version "*even*". Furthermore, the subtitle output tends to rephrase the spoken sentence towards standard written text, but retains the primary content of the message. Repetitions and speaking errors, such as the repeated "*die*" ("this") and "*dat*" ("that") in Example 2, are corrected in the subtitle. It is clear from the examples that the subtitle annotations are generally shorter than the verbatim annotations, and in many cases uninformative words and/or clauses are dropped.

### Generating subtitles from ASR outputs with LLMs

The previous section has highlighted the differences between verbatim transcripts that correspond to spoken language and subtitles that correspond to written language. The proposed methods in this paper are able to produce both annotation types directly from audio. In this section, we evaluate whether LLMs are capable to perform the same conversion task, but in the textual domain. More specifically, we task an LLM to convert a verbatim prediction of an ASR model to a subtitle using a specific prompt[10].

First, we evaluate the zero-shot capabilities of the LLM using this prompt and without giving any reference examples. Additionally, we evaluate the LLM when giving a set of reference examples in the prompt. To this end, $N$ pairs of ASR hypotheses and reference subtitles are added to the prompt, with $N$ the number of examples

[10] The following prompt was found to work best with the LLM: "You have to create a subtitle based on an automatically generated verbatim transcription from an ASR model. To this end, you have to convert the spoken language (in the verbatim transcription) to standard written language, without changing the content. This involves correcting errors and spelling mistakes, and removing disfluencies and hesitations. The transcription and subtitle will be in Dutch. Only respond with the subtitle." More extensive prompts (e.g. as in [76]) with extensive requirements did not achieve better performance.

**Table 8** BLEU scores (%) on *subs-valid-14kh* of subtitle prediction experiments (↑) with integration of LLMs and comparison to Whisper. The LLMs are prompted zero-shot without any examples or with *N* (5, 20 or 50) examples. For the proposed model with cascaded encoders using dual features, either the verbatim predictions or the subtitle predictions are used. The overall best model result is in bold. The best result for every model separately is underlined

| ***LLM*** | | **None** | **Llama-3.1** | | | | **Geitje** | | |
|---|---|---|---|---|---|---|---|---|---|
| **Model** | ***# Examples*** | **/** | $N = 0$ | $N = 5$ | $N = 20$ | $N = 50$ | $N = 0$ | $N = 5$ | $N = 20$ |
| E2E ASR | | <u>34.06</u> | 28.77 | 33.35 | 33.74 | 33.94 | 27.03 | 29.63 | 25.87 |
| Proposed model - verbatim | | 43.05 | 36.05 | 40.86 | <u>43.14</u> | 42.55 | 34.81 | 36.60 | 34.39 |
| Proposed model (XL) - verbatim | | 44.35 | 36.72 | 42.19 | <u>44.68</u> | 43.60 | 35.00 | 37.78 | 35.31 |
| Proposed model - subtitle | | <u>59.40</u> | 38.24 | 47.89 | 50.52 | 48.49 | 48.65 | 51.52 | 41.47 |
| Proposed model (XL) - subtitle | | **<u>61.76</u>** | 39.67 | 49.71 | 52.79 | 50.76 | 51.61 | 53.72 | 45.23 |
| Whisper | | <u>50.58</u> | 36.19 | 43.89 | 46.89 | 43.78 | 43.27 | 43.48 | 36.37 |
| Whisper finetuned | | 39.57 | 33.07 | 40.36 | <u>41.88</u> | 41.47 | 37.58 | 35.07 | 33.80 |

(5, 20 or 50). The examples are derived from a list of randomly selected utterances from the *subs-valid-14kh* test set. The ASR hypotheses in the examples correspond to the respective ASR model for which the outputs are used.

We use the recent multilingual instruction-tuned Llama-3.1 model[11] (8B parameters) [77], as well as a Dutch-finetuned language model based on Mistral-7B-v0.1 (7B parameters) [78] referred to as Geitje[12]. For Geitje, the prompt is translated to Dutch.

Table 8 depicts the BLEU scores on the long-form *subs-valid-14kh* test set. The ASR models correspond to the previous section (Table 6). The BLEU scores are calculated for the ASR outputs as well as for the LLM generated subtitles with these ASR outputs. For our proposed method, we evaluate both the verbatim prediction and the subtitle prediction of the model as inputs for the LLM and analyse which one is most suitable.

As the results in Table 8 reveal, converting the verbatim transcript to a subtitle (in Dutch) is not a trivial task for an LLM, despite their impressive performance on many language-related tasks. The Dutch-finetuned language model Geitje seems to work better than the Llama model for zero-shot (without examples) transcript to subtitle conversion. However, the resulting BLEU scores of the LLM pipeline are still below the BLEU scores attained with the subtitle output of our proposed model. In fact, using an LLM in a zero-shot fashion even leads to a lower BLEU score than the verbatim outputs without LLM. When the LLM is given some examples of how to do the subtitling task, i.e. leveraging in-context learning, its predictions tend to be better subtitles than the original verbatim outputs. The amount of in-context learning is limited by the context window and attentive capabilities of the specific LLM. As a result, our method, which learns to generate subtitles based on both verbatim and acoustic content, has significant benefits. We remark that the performance of the LLM pipeline could be further improved by conditioning on longer historical contexts and by integrating additional contextual information [76]. Finetuning the LLM with paired subtitle-verbatim examples or providing more accurate descriptions of the actual subtitling protocol (which we do not have) might yield further improvements, although adapting to verbatim transcripts is probably necessary.

Moreover, we note that the BLEU score for the standard Whisper model is already quite good. In combination with the results in Table 6, this further indicates that Whisper transcriptions are closer to standard written language than they are to verbatim transcriptions. In e.g. assessment tasks such as fluency measurement, this is a downside resulting from not distinguishing transcript types during weakly supervised training, which we aim to solve with the methods in this paper. If Whisper is finetuned on verbatim data, its subtitling capabilities are greatly reduced. In any case, our proposed models with dedicated subtitle decoders still attains the highest BLEU scores, although it should be noted that they are aware of the subtitling protocol as they were trained on subtitles from the same broadcaster.

[11] https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct

[12] https://huggingface.co/BramVanroy/GEITje-7B-ultra

**Abbreviations**

| | |
|---|---|
| ASR | Automatic speech recognition |
| BLEU | Bilingual evaluation understudy (score) |
| BPE | Byte-pair encoding |
| CBD | Colloquial Belgian Dutch |
| CGN | Corpus Gesproken Nederlands (Spoken Dutch Corpus) |
| CNN | Convolutional neural network |
| CTC | Connectionist temporal classification |
| E2E | End-to-end |
| GPU | Graphics processing unit |

| | |
|---|---|
| LLM | Large language model |
| LM | Language model |
| MAPSSWE | Matched-pair sentence segment word error |
| NLP | Natural language processing |
| SBD | Standard Belgian Dutch |
| WER | Word error rate |

**Acknowledgements**
Not applicable.

**Authors' contributions**
Both authors worked together to come up with the methods. JP developed the code, performed the experiments, analysed the results, and drafted the article. HV critically discussed the methods and experimental results with JP and proofread and revised the article. Both authors read and approved the final manuscript.

**Funding**
This research was supported by Research Foundation Flanders (FWO) under grant S004923N of the SBO programme and by the Flanders AI Impulse Programme - FAIR2.0. The computing resources and services used in this work were provided by the VSC (Flemish Supercomputer Center), funded by the Research Foundation Flanders (FWO) and the Flemish Government. We also thank VRT for the data resources.

**Data availability**
The Spoken Dutch Corpus (CGN) used in this work is available at https://taalmaterialen.ivdnt.org/download/tstc-corpus-gesproken-nederlands/ for research purposes and at https://taalmaterialen.ivdnt.org/download/tstc-corpus-gesproken-nederlands-c/ for commercial purposes. The subtitle data used in this work cannot be shared. The code for inference can be found at https://github.com/nelfproject/NeLF_Transcription_ASR/ and the models at https://huggingface.co/nelfproject/.

## Declarations

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

Received: 22 September 2025 Accepted: 16 February 2026
Published online: 04 March 2026

**References**

1. A. Vaswani, et al., in *Proc. Conf. on Neural Information Processing Systems (NeurIPS)*, Attention is all you need (2017), pp. 5998–6008
2. A. Gulati, et al., in *Proc. Interspeech*, Conformer: Convolution-augmented transformer for speech recognition (2020). pp. 5036–5040. https://doi.org/10.21437/Interspeech.2020-3015
3. Y. Zhang, BigSSL: exploring the frontier of large-scale semi-supervised learning for automatic speech recognition. IEEE J. Sel. Top. Signal Process. **16**(6), 1519–1532 (2022). https://doi.org/10.1109/JSTSP.2022.3182537
4. Y. Zhang, et al., in *Proc. Conf. on Neural Information Processing Systems (NeurIPS): SAS Workshop*, Pushing the limits of semi-supervised learning for automatic speech recognition (2022)
5. A. Baevski, H. Zhou, A. Mohamed, M. Auli, in *Proc. Conf. on Neural Information Processing Systems (NeurIPS)*, Wav2vec 2.0: A framework for self-supervised learning of speech representations (2020), pp. 12449–12460
6. W.-N. Hsu, B. Bolte, Y.-H. Tsai, K. Lakhotia, R. Salakhutdinov, A. Mohamed, HuBERT: self-supervised speech representation learning by masked prediction of hidden units. IEEE/ACM Trans. Audio Speech Lang. Process. **29**, 3451–3460 (2021). https://doi.org/10.1109/TASLP.2021.3122291
7. S. Chen, WavLM: large-scale self-supervised pre-training for full stack speech processing. IEEE J. Sel. Top. Signal Process. **16**(6), 1505–1518 (2022). https://doi.org/10.1109/JSTSP.2022.3188113
8. W.-N. Hsu, et al., in *Proc. Interspeech*, Robust wav2vec 2.0: Analyzing domain shift in self-supervised pre-training (2021), pp. 721–725. https://doi.org/10.21437/Interspeech.2021-236
9. A. Babu, et al., in *Proc. Interspeech*, XLS-R: Self-supervised cross-lingual speech representation learning at scale (2022), pp. 2278–2282. https://doi.org/10.21437/Interspeech.2022-143
10. S. Feng, B.M. Halpern, O. Kudina, O. Scharenborg, Towards inclusive automatic speech recognition. Computer Speech & Language **84**, 101567 (2024). https://doi.org/10.1016/j.csl.2023.101567
11. A. Radford, J.W. Kim, T. Xu, G. Brockman, C. Mcleavey, I. Sutskever, in *Proc. Int. Conf. on Machine Learning (ICML)*, Robust speech recognition via large-scale weak supervision (2023), pp. 28492–28518
12. Y. Peng, et al., in *Proc. IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, Reproducing Whisper-style training using an open-source toolkit and publicly available data (2023)
13. J.D. Cintas, A. Remael, *Audiovisual translation: subtitling* (Routledge, London, 2014)
14. A. Karakanta, M. Negri, M. Turchi, in *Proc. Int. Conf. on Language Resources and Evaluation (LREC)*, MuST-Cinema: a speech-to-subtitles corpus (2020), pp. 3727–3734
15. V. Panayotov, G. Chen, D. Povey, S. Khudanpur, in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, LibriSpeech: An ASR corpus based on public domain audio books (2015), pp. 5206–5210
16. Y. Yin, D. Mori, S. Fujimoto, in *Proc. 29th Annual Meeting of the Association for Natural Language Processing*, ReazonSpeech: A free and massive corpus for Japanese ASR (2023), pp. 1134–1139
17. J.-U. Bang, M.-Y. Choi, S.-H. Kim, O.-W. Kwon, Automatic construction of a large-scale speech recognition database using multi-genre broadcast data with inaccurate subtitle timestamps. IEICE Trans. Inf. Syst. **E103.D**(2), 406–415 (2020). https://doi.org/10.1587/transinf.2019EDP7234
18. S. Ando, H. Fujihara, in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Construction of a large-scale Japanese ASR corpus on TV recordings (2021), pp. 6948–6952
19. J. Poncelet, H. Van hamme, in *Proc. IEEE Spoken Language Technology Workshop (SLT)*, Learning to jointly transcribe and subtitle for endto- end spontaneous speech recognition (2023), pp. 182–189. doi: https://doi.org/10.1109/SLT54892.2023.10022420
20. M. Ihori, H. Sato, T. Tanaka, R. Masumura, S. Mizuno, N. Hojo, in *Proc. Interspeech*, Transcribing speech as spoken and written dual text using an autoregressive model (2023), pp. 461–465. https://doi.org/10.21437/Interspeech.2023-1655
21. N. Kanda, Y. Gaur, X. Wang, Z. Meng, T. Yoshioka, in *Proc. Interspeech*, Serialized output training for end-to-end overlapped speech recognition (2020), pp. 2797–2801. https://doi.org/10.21437/Interspeech.2020-999
22. J. Xu, F. Buet, J. Crego, E. Bertin-Lemée, F. Yvon, in *Proc. Int. Conf. on Spoken Language Translation (IWSLT)*, Joint generation of captions and subtitles with dual decoding (2022), pp. 74–82. https://doi.org/10.18653/v1/2022.iwslt-1.7
23. L. Lamel, J.-L. Gauvain, G. Adda, Lightly supervised and unsupervised acoustic model training. Comput. Speech Lang. **16**(1), 115–129 (2002). https://doi.org/10.1006/csla.2001.0186
24. P. Lanchantin, et al., in *Proc. Interspeech*, Selection of multi-genre broadcast data for the training of automatic speech recognition systems (2016), pp. 3057–3061. https://doi.org/10.21437/Interspeech.2016-462
25. P. Bell, S. Renals, in *Proc. IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, A system for automatic alignment of broadcast media captions using weighted finite-state transducers (2015), pp. 675–680. https://doi.org/10.1109/ASRU.2015.7404861
26. O. Saz, Lightly supervised alignment of subtitles on multi-genre broadcasts. Multimedia Tools Appl. **77**, 30533–30550 (2018). https://doi.org/10.1007/s11042-018-6050-1
27. 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 (2017), pp. 346–352. https://doi.org/10.1109/ASRU.2017.8268956
28. V. Gupta, P. Deléglise, G. Boulianne, Y. Estève, S. Meignier, A. Rousseau, in *Proc. IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, CRIM and LIUM approaches for multi-genre broadcast media transcription (2015), pp. 681–686. https://doi.org/10.1109/ASRU.2015.7404862
29. N.M. Guerreiro, R. Rei, F. Batista, Towards better subtitles: a multilingual approach for punctuation restoration of speech transcripts. Expert Syst. Appl. **186**, 115740 (2021). https://doi.org/10.1016/j.eswa.2021.115740

30. R. Geislinger, B. Milde, C. Biemann, in *Proc. Conf. on Natural Language Processing (KONVENS)*, Improved open source automatic subtitling for lecture videos (2022), pp. 98–103
31. B. Milde, R. Geislinger, I. Lindt, T. Baumann, in *Proc. Conf. on Electronical Speech Signal Processing (ESSV)*, Open source automatic lecture subtitling (2021), pp. 128–135
32. D. Liu, J. Niehues, G. Spanakis, in *Proc. Int. Conf. on Spoken Language Translation (IWSLT)*, Adapting end-to-end speech recognition for readable subtitles (2020), pp. 247–256. https://doi.org/10.18653/v1/2020.iwslt-1.30
33. X. Che, S. Luo, H. Yang, C. Meinel, in *Proc. IEEE Int. Conf. on Advanced Learning Technologies (ICALT)*, Automatic lecture subtitle generation and how it helps (2017), pp. 34–38. https://doi.org/10.1109/ICALT.2017.11
34. S. Papi, M. Gaido, A. Karakanta, M. Cettolo, M. Negri, M. Turchi, Direct speech translation for automatic subtitling. Trans. Assoc. Comput. Linguist. **11**, 1355–1376 (2023). https://doi.org/10.1162/tacl_a_00607
35. M. Ihori, A. Takashima, R. Masumura, in *Proc. Int. Conf. on Language Resources and Evaluation (LREC)*, Parallel corpus for Japanese spokento-written style conversion (2020), pp. 6346–6353
36. J. Liao et al., Improving readability for automatic speech recognition transcription. ACM Trans. Asian Low-Resour. Lang. Inf. Process. (2023). https://doi.org/10.1145/3557894
37. J. Nozaki, T. Kawahara, K. Ishizuka, T. in *Proc. Interspeech*, Hashimoto, End-to-end speech-topunctuated- text recognition (2022), pp. 1811–1815. https://doi.org/10.21437/Interspeech.2022-5
38. H. Futami, et al., in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Streaming joint speech recognition and disfluency detection (2023). https://doi.org/10.1109/ICASSP49357.2023.10094620
39. M. Sunkara, C. Shivade, S. Bodapati, K. Kirchhoff, in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Neural inverse text normalization (2021), pp. 7573–7577. https://doi.org/10.1109/ICASSP39728.2021.9414912
40. Z. Wang, Y. Wang, S. Wang, W. Che, in *Proc. Int. Conf. on Computational Linguistics (COLING)*, Adaptive unsupervised self-training for disfluency detection (2022), pp. 7209–7218
41. J. Guo, T.N. Sainath, R.J. Weiss, in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, A spelling correction model for end-to-end speech recognition (2019), pp. 5651–5655. https://doi.org/10.1109/ICASSP.2019.8683745
42. S. Li, X. Lu, S. Sakai, M. Mimura, T. Kawahara, in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Semi-supervised ensemble DNN acoustic model training (2017), pp. 5270–5274. https://doi.org/10.1109/ICASSP.2017.7953162
43. B. Li, T.N. Sainath, R. Pang, Z. Wu, in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Semi-supervised training for end-toend models via weak distillation (2019), pp. 2837–2841. https://doi.org/10.1109/ICASSP.2019.8682172
44. G. Chen, et al., in *Proc. Interspeech*, GigaSpeech: An evolving, multi-domain ASR corpus with 10,000 hours of transcribed audio (2021), pp. 3670–3674. https://doi.org/10.21437/Interspeech.2021-1965
45. B. Zhang, et al., in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, WenetSpeech: A 10000**+** hours multi-domain Mandarin corpus for speech recognition (2022), pp. 6182–6186. https://doi.org/10.1109/ICASSP43922.2022.9746682
46. D. Galvez, et al., in *Proc. Conf. on Neural Information Processing Systems (NeurIPS): Track on Datasets and Benchmarks*, The People's Speech: A large-scale diverse English speech recognition dataset for commercial usage (2021)
47. P. Wilken, P. Georgakopoulou, E. Matusov, in *Proc. Int. Conf. on Spoken Language Translation (IWSLT)*, SubER: A metric for automatic evaluation of subtitle quality (2022). https://doi.org/10.18653/v1/2022.iwslt-1.1
48. A. Karakanta, F. Buet, M. Cettolo, F. Yvon, in *Proc. Int. Conf. on Language Resources and Evaluation (LREC)*, Evaluating subtitle segmentation for end-to-end generation systems (2022), pp. 3069–3078
49. A. Graves, S. Fernández, F. Gomez, J. Schmidhuber, in *Proc. Int. Conf. on Machine Learning (ICML)*, Connectionist Temporal Classification: Labelling unsegmented sequence data with recurrent neural networks (2006), pp. 369–376. https://doi.org/10.1145/1143844.1143891
50. J. Lee, S. Watanabe, in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Intermediate loss regularization for CTC-based speech recognition (2021), pp. 6224–6228. https://doi.org/10.1109/ICASSP39728.2021.9414594
51. S. Watanabe, T. Hori, S. Kim, J.R. Hershey, T. Hayashi, Hybrid CTC/attention architecture for end-to-end speech recognition. IEEE Journal of Selected Topics in Signal Processing (JSTSP) **11**(8), 1240–1253 (2017). https://doi.org/10.1109/JSTSP.2017.2763455
52. A. Anastasopoulos, D. Chiang, in *Proc. Conf. of the North American Chapter of the Association for Computational Linguistics (NAACL): Human Language Technologies*, Tied multitask learning for neural speech translation, vol. 1 (2018). pp. 82–91. https://doi.org/10.18653/v1/N18-1008
53. H. Inaguma, S. Dalmia, B. Yan, S. Watanabe, in *Proc. IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, Fast-MD: Fast multi-decoder end-to-end speech translation with non-autoregressive hidden intermediates (2021), pp. 922–929. https://doi.org/10.1109/ASRU51503.2021.9687894
54. S. Dalmia, B. Yan, V. Raunak, F. Metze, S. Watanabe, in *Proc. Conf. of the North American Chapter of the Association for Computational Linguistics (NAACL): Human Language Technologies*, Searchable hidden intermediates for end-to-end models of decomposable sequence tasks (2021), pp. 1882–1896. https://doi.org/10.18653/v1/2021.naacl-main.151
55. J. Helcl, J. Libovický, D. Variš, in *Proc. Conf. on Machine Translation (WMT)*, CUNI system for the WMT18 multimodal translation task (2018), pp. 616–623. https://doi.org/10.18653/v1/W18-6441
56. S.-P. Chuang, Y.-S. Chuang, C.-C. Chang, H.-Y. Lee, in *Findings of the Association for Computational Linguistics (ACL-IJCNLP)*, Investigating the reordering capability in CTC-based non-autoregressive end-to-end speech translation (2021), pp. 1068–1077. https://doi.org/10.18653/v1/2021.findings-acl.92
57. B. Yan, et al., in *Proc. Conf. of the European Chapter of the Association for Computational Linguistics*, CTC alignments improve autoregressive translation (2023), pp. 1623–1639. https://doi.org/10.18653/v1/2023.eacl-main.119
58. N. Oostdijk, in *Proc. Int. Conf. on Language Resources and Evaluation (LREC)*, The Spoken Dutch Corpus: Overview and first evaluation, vol. 2 (2000)
59. T. Ko, V. Peddinti, D. Povey, S. Khudanpur, in *Proc. Interspeech*, Audio augmentation for speech recognition (2015), pp. 3586–3589. https://doi.org/10.21437/Interspeech.2015-711
60. J. Poncelet, H. Van hamme, in *Proc. IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, Comparison of self-supervised speech pre-training methods on Flemish Dutch (2021), pp. 169–176. https://doi.org/10.1109/ASRU51503.2021.9688061
61. D.S. Park, et al., in *Proc. Interspeech*, SpecAugment: A simple data augmentation method for automatic speech recognition (2019), pp. 2613–2617. https://doi.org/10.21437/Interspeech.2019-2680
62. S. Watanabe, et al., in *Proc. Interspeech*, ESPnet: End-to-end speech processing toolkit (2018), pp. 2207–2211. https://doi.org/10.21437/Interspeech.2018-1456
63. D.P. Kingma, J. Ba, in *Proc. Int. Conf. on Learning Representations (ICLR)*, Adam: A method for stochastic optimization (2015). arXiv:1412.6980
64. K. Papineni, S. Roukos, T. Ward, W.-J. Zhu, in *Proc. 40th Annual Meeting of the Association for Computational Linguistics*, BLEU: a method for automatic evaluation of machine translation (2002), pp. 311–318. https://doi.org/10.3115/1073083.1073135
65. M. Post, in *Proc. Conf. on Machine Translation (WMT)*, A call for clarity in reporting BLEU scores (2018), pp. 186–191
66. D.S. Pallet, W.M. Fisher, J.G. Fiscus, in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Tools for the analysis of benchmark speech recognition tests (1990), pp. 97–100. https://doi.org/10.1109/ICASSP.1990.115546
67. P. Koehn, in *Proc. Conf. on Empirical Methods in Natural Language Processing*, Statistical significance tests for machine translation evaluation (2004), pp. 388–395
68. R. Vandekerckhove, A. De Houwer, A. Remael, Between language policy and linguistic reality: intralingual subtitling on Flemish television. Pragmatics **19**(4), 609–628 (2009). https://doi.org/10.1075/prag.19.4.05van
69. A. Remael, A. De Houwer, R. Vandekerckhove, Intralingual open subtitling in Flanders: audiovisual translation, linguistic variation and audience needs. J. Specialised Transl. (JoSTrans) **10**, 76–105 (2008)
70. L. Prieels, G. De Sutter, A mixed-method approach to the use of colloquial Belgian Dutch in intralingual subtitling on Flemish television. Taal en Tongval **70**(2), 211–256 (2018). https://doi.org/10.5117/TET2018.2.PRIE
71. R. Vandekerckhove, Dialect loss and dialect vitality in Flanders. Int. J. Sociol. Lang. **2009**(196–197), 73–97 (2009). https://doi.org/10.1515/IJSL.2009.017

72. R. Vandekerckhove, "Tussentaal" as a source of change from below in Belgian Dutch. A case study of substandardization processes in the chat language of Flemish teenagers, in *Germanic Language Histories 'from Below' (1700–2000), 'Tussentaal' as a source of change from below in Belgian Dutch. A case study of substandardization processes in the chat language of Flemish teenagers*. (De Gruyter, Berlin, 2007), pp. 189–204. https://doi.org/10.1515/9783110925463.189
73. L.E. Shafey, H. Soltau, I. Shafran, in *Proc. Interspeech*, Joint speech recognition and speaker diarization via sequence transduction (2019), pp. 396–400. https://doi.org/10.21437/Interspeech.2019-1943
74. N. Kanda, et al., in Proc. Interspeech, Streaming multi-talker ASR with token-level serialized output training (2022), pp. 3774–3778. https://doi.org/10.21437/Interspeech.2022-7
75. H. Choi et al., Spoken-to-written text conversion for enhancement of Korean-English readability and machine translation. ETRI J. **46**(1), 127–136 (2024). https://doi.org/10.4218/etrij.2023-0354
76. J. Liu, C. Deng, Q. Zhang, Q. Chen, H. Yu, W. Wang, Recording for Eyes, Not Echoing to Ears: Contextualized Spoken-to-Written Conversion of ASR Transcripts (2024). arXiv:2408.09688
77. A. Dubey, et al., The Llama 3 Herd of Models (2024). arXiv:2407.21783
78. A.Q. Jiang, et al., Mistral 7B (2023). arXiv:2310.06825

## Publisher's Note

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

**Jakob Poncelet** Obtained the MSc degree in Electrical Engineering from KU Leuven in 2019 and the PhD degree from KU Leuven in 2025. He currently works as a post-doc researcher at the Department of Electrical Engineering (ESAT-PSI) at KU Leuven. His main research interests include automatic speech recognition, self-supervised learning, speech-LLMs, and noise and accent robust speech recognition.

**Hugo Van hamme** Hamme obtained the MSc degree from Imperial College, London in 1988 and the PhD degree from Vrije Universiteit Brussel in 1992. From 1993 till 2002, he held positions from senior researcher till senior director of research at Lernout & Hauspie and Scansoft. In 2002, he joined KU Leuven as a professor in speech processing. His research interests include automatic speech recognition, spoken language understanding, speech assessment for speech therapy and modelling of human speech recognition.