university of groningen / faculty of science and engineering / artificial intelligence

Language Model Integration and Confidence-Based Self-Training for Low-Resource Gronings Speech Recognition

Bachelor's Project Thesis

Yi Wang, s5517559, y.wang.259@student.rug.nl,
Supervisors: Dr. Maruf Dhali & Koen van Schuijlenburg

Abstract: Automatic speech recognition (ASR) performs poorly for low-resource language varieties such as Gronings, a Low-Saxon variety spoken in the north of the Netherlands. Building on the work of Bartelds et al. (2023), this study investigates two routes to improving a fine-tuned XLS-R acoustic model without collecting additional transcriptions: (i) integrating an external n-gram language model at decoding time, and (ii) confidence-based self-training, in which low-confidence pseudo-labels are filtered out and the unlabelled pool is scaled with new out-of-domain audio. A KenLM language model trained on 2.5 million words of written Gronings yields a consistent relative word error rate (WER) reduction of roughly 10 to 12 percent on both an in-domain and an out-of-domain test set, and the gain persists even for a much stronger acoustic model. For self-training, confidence filtering of the limited in-domain pool does not help and can hurt the performance, because filtering trades quantity for only a marginal quality gain. Adding a moderate amount (3 to 10 hours) of new unlabelled audio lowers the WER on both test sets, but adding the full 22 hours of noisy audio shows diminishing returns and even reversal. On this large, noisy pool, the outcome reverses: every confidence-filtered model beats the baseline out-of-domain, the best result being obtained by keeping the top 75 percent, while in-domain performance stays close to the baseline. This indicates that on a large, noisy pool, confidence filtering can meaningfully improve out-of-domain generalisation without hurting in-domain performance.

1 Introduction

Modern automatic speech recognition (ASR) systems reach high accuracy for languages for which large amounts of transcribed speech are available for training. For minority languages, regional languages and dialects, however, performance remains substantially lower, because the supervised data needed to adapt models to these varieties is scarce (Bartelds et al., 2023). Communities that speak such varieties therefore benefit little from recent advances in speech technology.

The core obstacle is not audio but transcribed audio. Recording speech is relatively easy, but turning it into paired audio and text samples that supervised training needs requires manual labeling. This manual-transcription bottleneck is what keeps performance low, and it is the reason data-augmentation methods, which create additional training material without additional manual transcription, are ways to mitigate this issue. The methods studied here follow that logic: rather than transcribe more speech, they make fuller use of resources that are cheap or already available, namely untranscribed audio and written text.

Gronings is one such low-resource language. It is a Low-Saxon variety spoken in the province of Groningen in the north of the Netherlands, with regional lexical, grammatical, and acoustic variation. There are only limited hours of transcribed Gronings speech available for training, which makes it a true low-resource setting. Self-supervised speech models such as wav2vec 2.0 (Baevski et al., 2020) and its multilingual successor XLS-R (Babu et al., 2021) mitigate this problem: they are pre-trained on large quantities of unlabelled multilingual audio and can then be fine-tuned for a target lan-

guage with comparatively little transcribed data. Even so, the amount of labelled Gronings speech remains the limiting factor, and obtaining more transcriptions is expensive. With only a few hours of Gronings, the model sees each word form a handful of times at most, so rare words and spellings stay poorly modelled. Bartelds et al. (2023) studied data-augmentation strategies for exactly this setting. They showed that self-training, which uses a model trained on a small amount of transcribed speech to automatically label additional audio, reduces the Gronings word error rate (WER) by up to 20.5 percent relative, and that augmentation with a text-to-speech system reduces it by up to 25.5 percent. Their approach, however, leaves three openings. First, all generated transcriptions were used, with no attempt to filter out unreliable ones. Second, the unlabelled audio pool was limited to the in-domain training recordings, which consist of Gronings book texts read aloud by a small number of speakers. Third, no external language model was used during decoding since the authors argued that there was little Gronings text available.

This study addresses all three openings. A substantial corpus of written Gronings (published books, short stories and columns) is collected and used to train an n-gram language model that is integrated into the decoder of an ASR model. In addition, a confidence-based self-training pipeline is developed in which the acoustic model's confidence is used to filter pseudo-labels, and in which the unlabelled pool is enlarged with new and out-of-domain Gronings audio recordings, which are different from the read-aloud training data in speaker and content. The overarching research question is: to what extent do language model integration and confidence-based self-training improve low-resource Gronings ASR, and how does the size and noise of the data pool affect the effectiveness of confidence filtering?

2 Background and related work

This section introduces the three components the thesis builds on: self-supervised acoustic models for low-resource ASR, self-training with pseudo-labels, and n-gram language models in CTC decoding.

2.1 Self-supervised models for low-resource ASR

wav2vec 2.0 (Baevski et al., 2020) learns speech representations from raw audio through a contrastive self-supervised objective, after which a linear layer is added and the model is fine-tuned for ASR using connectionist temporal classification (CTC; Graves et al., 2006). CTC allows the network to be trained on unsegmented speech by marginalizing over all alignments between audio frames and output characters, using a special blank symbol for frames that emit no character. XLS-R (Babu et al., 2021) extends this pre-training approach to a multilingual setting, pre-training on roughly 436,000 hours of speech in 128 languages. Because multilingual pre-training transfers well to unseen low-resource languages (Conneau et al., 2020), XLS-R is a strong starting point for Gronings, and it is the acoustic model used throughout this work, following Bartelds et al. (2023).

Such models help in the low-resource setting because pre-training has already captured general acoustic structure from large amounts of unlabelled multilingual speech. Fine-tuning then only has to learn how those representations map onto the characters of the target language, which needs far less transcribed speech than training a model from scratch would.

2.2 Self-training and pseudo-labeling

Self-training is a semi-supervised learning technique in which a teacher model, trained on the available labelled data, transcribes a pool of unlabelled data; the resulting pseudo-labels are then combined with the labelled data to train a student model (Kahn et al., 2020). The student can outperform the teacher because it is exposed to more acoustic variation. A difficulty is that pseudo-labels contain errors, and training on incorrect labels can reinforce the teacher's mistakes. Iterative and filtering-based variants have been proposed to limit this effect (Xu et al., 2020): by keeping only the labels the model is most confident in, filtering feeds the student a cleaner signal, though this helps only if the data it discards is worth less than the noise it removes. The present work follows the self-training procedure of Bartelds et al. (2023), but

adds an explicit confidence-based filtering step, motivated by the intuition that discarding the less reliable pseudo-labels should improve the student.

2.3 Language models in CTC decoding

A CTC acoustic model emits a probability distribution over characters for each frame independently. Greedy decoding simply takes the most likely character per frame; it ignores any knowledge of which character sequences form plausible words. An external language model can be added during beam-search decoding, scoring candidate transcriptions by their linguistic plausibility. The combined score balances the acoustic model against the language model through a weight α and a word-insertion bonus β (Hannun et al., 2014). Bartelds et al. (2023) deliberately did not use a language model due to the scarcity of Gronings text.

An n-gram language model estimates the probability of a word from the previous $n-1$ words, using counts collected from a text corpus. Despite their simplicity, n-gram models remain well suited to CTC beam-search rescoring: they are fast to query, integrate naturally into the left-to-right beam search, and, unlike large neural language models, can be trained reliably on a few million words, which is roughly the amount of Gronings text available in this study. KenLM (Heafield, 2011) provides an efficient implementation with modified Kneser–Ney smoothing, which redistributes probability mass to unseen n-grams and is important when the corpus is small. The decoder used in this work, pyctcdecode (Kensho Technologies, 2021), combines the acoustic and language model scores during beam search.

3 Data

All speech data are sampled at 16 kHz, and all transcriptions are normalised to the lowercase Gronings character set used by the acoustic model. Table 3.1 summarises the resources used.

3.1 Labelled speech

The labelled speech is drawn from the read-aloud Gronings corpus used by Bartelds et al. (2023), consisting of book texts read by a small number of speakers. Following their protocol, the smallest training subset of approximately 24 minutes of transcribed speech is used for self-training the teacher model. A held-out validation set is used for two purposes: to select the best fine-tuning checkpoint of every model, and to tune the language model decoding hyperparameters α and β. Evaluation then uses two test sets throughout, one in-domain and one out-of-domain. The in-domain test set (213 utterances) consists of read-aloud material of the same domain as the training data, with speaker overlap. The out-of-domain test set (71 utterances) contains unseen speakers and is used to test generalisation.

3.2 Text corpus for the language model

To train the language model, a corpus of written Gronings was assembled from three sources: a collection of short literary stories, the transcriptions of a Gronings audiobook, and a large set of published books and magazine issues (columns and prose). Only the transcriptions of the audiobook are used as text for the language model and its audio is not added as labelled speech, because it is read by a single narrator and would bias the acoustic model towards one voice. Because Gronings and Dutch are closely related and the source material is mixed, files that were flagged as Dutch or as low quality (for example containing scanning errors) were excluded using a metadata sheet; training the language model on Dutch text would bias the decoder towards Dutch spellings. Each remaining line was normalised to match the acoustic model's output: it was lower-cased, Unicode-normalised, split on hyphens, and stripped of any character outside the model's alphabet. After normalisation the corpus contains 2,508,884 words and 104,931 unique word forms. The list of unique word forms is also given to the decoder, which uses it when scoring words that the language model has not seen. This corpus is entirely text, requiring no audio, and is the resource that makes language model integration possible.

3.3 Unlabelled audio pool

Two unlabelled audio pools are used for self-training. The first is the in-domain pool: the re-

Table 3.1: Overview of the data used. The in-domain pool has gold transcriptions that are withheld during pseudo-labeling; the new audio is genuinely unlabelled. ST = self-training, ID = in-domain, OOD = out-of-domain.

Resource	Role	Size	Source
Gold train (24 min)	teacher training	24 min	Bartelds et al. (2023)
Validation set	selection, tuning	213 utt.	Bartelds et al. (2023)
In-domain test	evaluation (ID)	213 utt.	Bartelds et al. (2023)
Out-of-domain test	evaluation (OOD)	71 utt.	Bartelds et al. (2023)
In-domain pool	ST pool (Ph. 1)	168 min, 1,487 utt.	Bartelds et al. (2023)
New audio pool	ST pool (Ph. 2–3)	22.3 h, 20,828 seg.	collected for this study
Text corpus	language model	2.5 M words	collected for this study

maining 168 minutes of the read-aloud training recordings, treated as if they were unlabelled by withholding their transcriptions from the teacher.

The second is a collection of new, previously untranscribed Gronings audio: 101 long recordings (several minutes to tens of minutes each) of regional journalism and biographical material. Long recordings cannot be fed to the model directly, so they were split into utterance-sized chunks with the Silero voice activity detector (Silero Team, 2024), which detects speech regions and cuts at silences. Each chunk was resampled to 16 kHz. This yielded 20,828 segments totalling 22.27 hours of speech, with a mean segment length of about 3.8 seconds, which is comparable to the labelled training utterances. This material is unlabelled and differs from the training data in content (journalism rather than literary fiction). Pseudo-label confidence reflects this shift. Scored by the model that labels each pool (Section 4.3), the median non-blank confidence (the model's average certainty over the frames that emit a character rather than the CTC blank symbol) is 0.983 for the in-domain pool but only 0.930 for the new audio. Figure 3.1 shows the full distributions: the in-domain pool is concentrated near the maximum confidence, whereas the new audio has a wider, lower tail. It is precisely this low-confidence tail that confidence filtering is intended to remove.

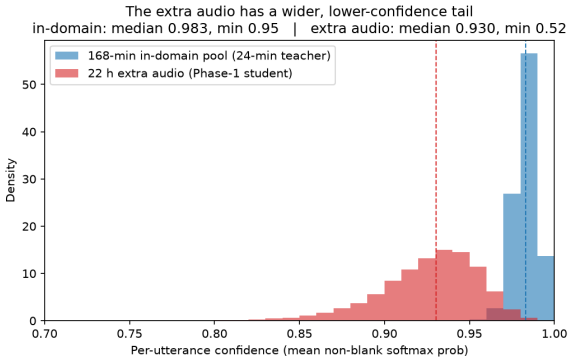

Figure 3.1: Per-utterance confidence (mean non-blank max-softmax probability) for the two unlabelled pools, each scored by the model that labelled it: the 24-min teacher for the in-domain pool and the Phase-1 student for the 22-hour pool (Section 4.3). These are the confidence signals actually used for filtering.

4 Methods

This study makes two changes to the pipeline of Bartelds et al. (2023), and the two act at different points. Language-model integration (Section 4.2) changes only how a finished acoustic model is decoded, and leaves its weights untouched. Confidence-based self-training (Section 4.3) changes what the acoustic model is trained on, by adding automatically transcribed speech and filtering it by confidence. Because they act at different points, the two were implemented and measured separately. All hyperparameter settings appear in Table 4.1.

4.1 Acoustic model and fine-tuning

The acoustic model is the 300-million-parameter XLS-R model (Babu et al., 2021), fine-tuned to predict Gronings characters with a CTC head. The teacher is the 24-minute Gronings model released by Bartelds et al. (2023). Every student is fine-tuned in this study, each one starting from the pre-

Table 4.1: Main hyperparameters used throughout the experiments. The same fine-tuning configuration is used for the teacher and all students. LM = language model.

Parameter	Value
Acoustic model	XLS-R-300M (CTC)
Learning rate	1×10^{-5}
Training steps	13,000
Batch size	8×4 (grad. accum.)
LR schedule	10/40/50% warmup/flat/decay
Precision	fp16
Feature encoder	frozen
Model selection	best dev WER
LM toolkit	KenLM ($n = 2, 3, 4$)
Decoder	pyctcdecode beam search
LM weight α	$\{0.3, 0.5, 0.7, 0.9, 1.1\}$
Word bonus β	$\{0, 0.5, 1, 1.5, 2\}$
Confidence measure	mean non-blank max-softmax
Keep fractions	100, 90, 75, 50, 25%
Added audio (Phase 2)	3 h, 10 h, 22.27 h
Evaluation metrics	WER, CER (`jiwer`)

trained XLS-R checkpoint, so that the students differ only in the data they are trained on. The full list of settings is given in Table 4.1. Each run keeps the checkpoint with the lowest validation set WER rather than the lowest validation loss since WER is the metric of interest, so the checkpoint that minimizes the loss is not necessarily the one that transcribes best. In addition, a fixed random seed is used throughout this study.

4.2 Language model integration

Language-model integration proceeds in three steps: building a text corpus, training the language model on it, and integrating that model into the decoder.

Step 1: corpus construction. The corpus of written Gronings described in Section 3.2 is assembled from published books, short stories and columns, cleaned of Dutch and low-quality files, and normalised to the character set the acoustic model predicts. It contains 2.5 million words. No audio is involved in this step, which makes the approach cheap and feasible.

Step 2: language model training. Count-based n-gram models of order 2, 3 and 4 are trained on that corpus with KenLM (Heafield, 2011), an existing toolkit for building n-gram language models. Orders 2, 3 and 4 are all trained, so that the effect of using a longer word context can be measured.

Step 3: decoder integration. The baseline decoding strategy is greedy decoding: at each output frame the character with the highest probability is selected independently, and the resulting sequence is collapsed into a transcription by the CTC rule, deleting repeats and then blanks. Greedy decoding is fast but purely acoustic. It has no notion of which character sequences form real words, so it produces non-words, or splits and merges words at the wrong places. Beam-search decoding instead keeps a fixed number of candidate transcriptions as it proceeds, and can therefore let a linguistically plausible hypothesis overtake a higher-scoring but implausible one. Adding a language model to the beam search supplies exactly this linguistic knowledge, at the level of whole word sequences rather than individual frames.

Beam search is implemented with pyctcdecode (Kensho Technologies, 2021), which combines the acoustic model's frame-level log-probabilities with the language model. Each candidate transcription is scored by

$$\text{score} = \log p_{\text{AM}} + \alpha \log p_{\text{LM}} + \beta \, n_{\text{words}}, \qquad (4.1)$$

where p_{AM} is the acoustic-model probability of the model's predicted transcription, p_{LM} its language-model probability, α weights the language model, and β is a word-insertion bonus that counteracts the language model's tendency to prefer shorter sequences. Both α and β have to be set, and their best values are not known in advance. For each n-gram order, every one of the 25 combinations of the five candidate values of α and the five of β listed in Table 4.1 is evaluated on the validation set, and only the single best combination is then applied once to the test set. Tuning in the validation set and reporting on the test set ensure that the reported improvements are not an artifact of fitting the decoder to the evaluation data.

For every acoustic model and test set, the model is decoded once with greedy decoding and once with each of the three n-gram orders. Greedy decoding, which uses no language model and is the strategy used by Bartelds et al. (2023), is the baseline against which the three orders are compared.

4.3 Confidence-based self-training

The procedure of Bartelds et al. (2023), which serves as the starting point, has three steps. A teacher model, trained on the 24 minutes of transcribed speech, transcribes a pool of unlabelled audio. A student model is then fine-tuned on the 24 minutes of gold data together with the whole pseudo-labelled pool. The student can improve on the teacher because the extra audio exposes it to more acoustic variation than the 24 gold minutes contain, even though some of its labels are wrong. The teacher has a fairly high error rate (in-domain WER around 0.33), so a substantial share of the pseudo-labels it produces are incorrect, and in this baseline procedure all of them are kept regardless of quality.

The contribution of this study is to attach a confidence score to each pseudo-label and to use that score to decide which labels to keep. The underlying hypothesis is that discarding the least reliable labels leaves the student with cleaner supervision. Whether it does turns out to depend on the pool, which is what the three phases below are designed to separate.

The confidence score is read off the model's own output. When the labelling model transcribes an utterance, it produces, for every short time frame of the audio, a probability for each possible character. A model that is sure of itself puts nearly all of that probability on one character, while an unsure model spreads it across several characters. Taking the highest probability in each frame and averaging those values over the utterance therefore gives a single number expressing how certain the model was. It is worth noting that most frames of a CTC model emit the blank symbol, almost always with very high probability, so including them would influence the average and make every utterance look equally confident. The average is therefore taken over the non-blank frames only. Algorithm 4.1 states the procedure. Utterances are ranked by this score, and only the most confident top-X percent are kept.

The self-training experiments are organised into three phases.

Phase 1: filtering the in-domain pool. The teacher transcribes the 168-minute in-domain pool, producing 1,487 pseudo-labelled utterances, each with a confidence score. These are ranked by confidence, and the most confident 100, 90, 75, 50 and 25 percent are retained in turn. For each of the five cut-offs a student is fine-tuned on the 24 minutes of gold data together with the retained pseudo-labels. The keep-100 percent student retains everything and therefore reproduces the unfiltered model, which is the baseline against which the four filtered students are judged.

Algorithm 4.1 Computing the confidence of one pseudo-label.

Require: a labelling model M and one audio clip x

1: Run M on x. For every time frame t, this gives a probability for each possible character.
2: For each frame t, let c_t be the character with the highest probability, and let p_t be that probability.
3: Build the transcription by writing down $c_1, c_2, \ldots$ and applying the CTC rule: first delete repeated characters, then delete the blanks.
4: Keep only the frames whose chosen character c_t is not the blank symbol.
5: Set the confidence to the average of p_t over those kept frames.
6: **return** the transcription and its confidence.

Phase 2: scaling the pool with new audio. The 22.27 hours of new audio (Section 3.3) are pseudo-labelled not by the 24-minute teacher but by the keep-100 percent Phase-1 student. Relabelling further unlabelled audio with an improved student rather than with the original teacher is the iterative form of self-training: following Xu et al. (2020), Bartelds et al. (2023) likewise used each student to transcribe the next set of unlabelled speech. Among the five Phase-1 students the keep-100 percent model was chosen as the labeller because it generalises best to unseen speakers, having the lowest out-of-domain WER.

The training set of that same student, namely the 24 minutes of gold data together with the 168-minute pseudo-labelled pool, forms a fixed base to which the new audio is added. To decide which new segments to add, the segments are shuffled once with a fixed random seed and then taken in that order until their cumulative duration reaches 3 hours,

10 hours, or the whole 22.27 hours, producing three training sets. This isolates the effect of quantity.

Phase 3: filtering the large pool. Phase 3 varies quality instead of quantity. All 20,828 segments of the new pool are ranked by the labelling student's confidence, the same score shown for this pool in Figure 3.1, and the most confident 90, 75, 50 and 25 percent are retained. Each retained set is added to the same fixed base of 24 minutes of gold data plus the 168-minute pool, and a student is fine-tuned on the result. The amount of new audio shrinks as the cut-off tightens, exactly as in Phase 1, but the pool being filtered is now large and noisy rather than small and clean. Comparing the two phases therefore isolates the effect of the pool itself on the value of confidence filtering, which is the second half of the research question.

4.4 Design choices and alternatives

The acoustic model is fixed to XLS-R-300M rather than a Dutch monolingual model, because Bartelds et al. (2023) found multilingual pre-training to transfer better to Gronings, and because reusing their model keeps the present results comparable to theirs. A count-based n-gram language model was preferred over a neural one: a corpus of 2.5 million words is small enough that a neural model would tend to over-fit it, and such a model would also be slow to query inside a beam search, whereas a smoothed n-gram model is stable, fast, and cheap enough to train at three different orders. Finally, filtering was tested at four levels, keeping the top 90, 75, 50, and 25 percent, which span mild to aggressive filtering. Only a few levels were used because each one requires a separate, full fine-tuning run.

4.5 Evaluation

All systems are evaluated by WER and, as a finer-grained complement, CER, both computed with the `jiwer` library.* WER is the proportion of word substitutions, insertions and deletions needed to turn the hypothesis into the reference; CER is the same quantity at the character level, and is informative here because the acoustic model operates on characters and because a single misrecognised character already counts as a whole word error under WER. For the language-model experiments, greedy decoding and language-model decoding transcribe exactly the same utterances, so their transcriptions can be compared one utterance at a time, counting how many the language model improves, leaves unchanged or worsens relative to greedy decoding. Because the test sets are small (213 and 71 utterances) and each condition was trained only once, the WER differences should be interpreted as trends rather than as exact rankings.

The two test sets play complementary roles and are reported side by side throughout. The in-domain set measures how well a system does on the same kind of read-aloud material, and from the same overlapping speakers as the training data, it is the easier setting and reflects performance under matched conditions. The out-of-domain set contains unseen speakers and is the harder and more informative setting, because any real deployment of a Gronings ASR system will meet speakers and recording conditions absent from the training data. A method that improves only the in-domain number risks merely fitting the training distribution more tightly, whereas one that also improves the out-of-domain number is generalizing better. Both numbers are therefore reported for every system.

4.6 Experimental setup

The labelled corpus is split into training, validation and test sets in an 80/10/10 ratio, following Bartelds et al. (2023). The in-domain validation and test sets contain 213 utterances each. The teacher was trained on a 24-minute subset of the training split, and the remaining 168 minutes form the in-domain unlabelled pool. The out-of-domain test set contains 71 utterances from speakers not present in training.

Twelve acoustic models were fine-tuned for this study: the five Phase-1 students, the three Phase-2 scaling models and the four Phase-3 filtering models. Each uses the identical configuration of Table 4.1, so that the systems differ only in the data they were trained on. Two further models are used but not trained here, being checkpoints released by Bartelds et al. (2023) that were downloaded and evaluated: the 24-minute teacher, and a stronger 192-minute acoustic model obtained by continued

*`https://github.com/jitsi/jiwer`

pre-training on Gronings followed by fine-tuning on the full 192 minutes.

The language-model experiments require no fine-tuning at all, since they only change how a finished model is decoded. Two acoustic models are used, the 24-minute teacher and the stronger 192-minute model, and each is decoded on both test sets with greedy decoding and with each of the three n-gram orders. The 2-, 3- and 4-gram models are therefore compared against the same greedy baseline, on the same two test sets, for both a weak and a strong acoustic model.

Table 5.1: WER with and without the n-gram language model, for the 24-minute teacher and the stronger 192-minute reference model. For each row n is the best-performing n-gram order and α, β its tuned weights; Rel. is the relative change from greedy to LM decoding. ID = in-domain, OOD = out-of-domain, LM = language model.

Model	Test	n	α	β	Greedy WER	LM WER	Rel.
24 min	ID	2	0.9	2.0	0.334	0.295	−11.8%
24 min	OOD	3	0.9	0.5	0.454	0.409	−9.7%
192 min	ID	4	1.1	0.0	0.148	0.132	−11.0%
192 min	OOD	3	1.1	2.0	0.315	0.283	−10.3%

5 Results

The results are presented in the order of the two contributions: first language-model integration, then the three self-training phases. For the language-model experiments the point of comparison is greedy decoding of the same acoustic model. For the three self-training phases the baseline is the unfiltered keep-100 percent student.

5.1 Language model integration

Table 5.1 shows the effect of adding the language model. For the 24-minute teacher, the best language model reduces the in-domain WER from 0.334 to 0.295 (11.8 percent relative) and the out-of-domain WER from 0.454 to 0.409 (9.7 percent). The language model's main effect is to repair word boundaries and spellings that greedy decoding produces. For the reference "ze stroalt mit aal dat gefriemel op schoot", greedy decoding writes the verb as "stroalte" and runs two words together as "aaldat"; the language model, having seen the correct forms in the text corpus, restores "stroalt" and splits "aal dat", cutting the utterance WER from 0.50 to 0.12. For the reference "hai hoalt zien snoepies te veurschien", greedy decoding splits one word into "hoal t" and the language model rejoins it into "hoalt", giving the correct transcription.

The benefit is also not sensitive to the exact n-gram order. On the in-domain test set, the three orders give almost identical relative reductions (11.8, 11.7 and 11.6 percent for the 2-, 3- and 4-gram models, respectively), and on the out-of-domain test set they range only between 8.4 and 9.7 percent. Higher orders did not help further, which is expected given the corpus size: with 2.5 million words, there are too few examples of any specific four-word sequence for a 4-gram to add much over a 2- or 3-gram. The best settings found on the validation set used a fairly high language model weight (α around 0.9 to 1.1), confirming that the language model is contributing substantially to the decision. Table 5.1 reports, for each model and test set, the single best-performing order.

A natural concern is that a language model only helps a weak acoustic model. To test this, the same procedure was applied to a much stronger Gronings model (a continued-pre-trained model fine-tuned on the full 192 minutes, with a greedy in-domain WER of 0.148). As the lower half of Table 5.1 shows, the language model still reduces WER by 11.0 percent in-domain and 10.3 percent out-of-domain. The benefit is therefore robust to acoustic model strength and is best understood as a standalone gain.

5.2 Phase 1: filtering the in-domain pool

Training on the 24 minutes of gold data together with the unfiltered 168-minute pseudo-labelled pool lowers the in-domain WER from 0.334 for the teacher to 0.315. This keep-100 percent student is the baseline against which the filtering conditions are judged.

Confidence filtering, however, provides no further benefit and eventually hurts. As Figure 5.1 shows, keeping the top 90 or 75 percent leaves the WER essentially unchanged (0.315 and 0.314), while keep-

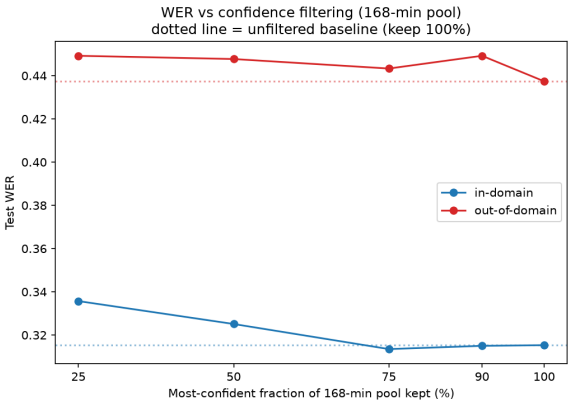

Figure 5.1: Phase 1. Test WER as a function of the most-confident fraction of the 168-minute in-domain pool that is kept. The dotted lines mark the unfiltered (keep-100 percent) baseline. Filtering does not help and hurts at aggressive cut-offs.

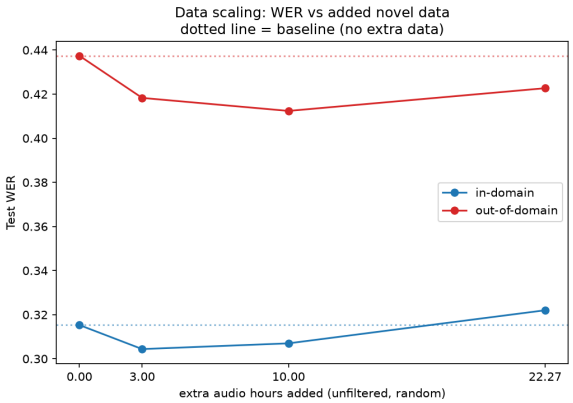

Figure 5.2: Phase 2. Test WER as a function of the amount of new (unfiltered) out-of-domain audio added to the fixed base. The dotted lines mark the baseline. Gains appear at 3 to 10 h and reverse at 22 h.

ing only the top 50 or 25 percent raises it to 0.325 and 0.336, clearly worse than the baseline.

The reason is a trade-off between quality and quantity. Keeping only the most confident pseudo-labels discards a large part of the training data. Any gain in label quality from dropping the least confident labels is evidently too small to make up for this loss, because on a pool that is already fairly clean and quite small the students trained on less data end up worse. The effect is monotonic in the wrong direction: the more aggressively the pool is filtered, the fewer training minutes remain and the worse the student becomes.

5.3 Phase 2: scaling with new audio

Adding new out-of-domain audio to the unfiltered pool helps, but only up to a point (Figure 5.2). Adding roughly 3 hours lowers the in-domain WER to 0.304, the best in-domain result among all self-training systems, and the out-of-domain WER to 0.418. Adding 10 hours gives a similar in-domain WER (0.307) and the best out-of-domain WER of the scaling experiment (0.412). Adding the full 22 hours, however, reverses part of the gain (in-domain 0.322, out-of-domain 0.423). More data is therefore not uniformly better, and the curve is consistent with a data-saturation effect.

Two features of this result deserve emphasis. First, improvement from even a few hours of extra data appears on both test sets, including the out-of-domain one, which is encouraging. Second, the turning point lies somewhere between 10 and 22 hours, meaning that the best approach is not as much data as possible but a moderate amount. For a low-resource setting this is practically useful, because collecting and pseudo-labelling tens of hours of audio is far more costly than collecting a few, and the extra effort is shown here to bring no benefit and even a small penalty.

5.4 Phase 3: filtering the large pool

When confidence filtering is instead applied to the full 22-hour pool, the result differs from Phase 1 (Figure 5.3). Every filtered variant beats the baseline on the out-of-domain test set, where the baseline sits at 0.437, and the best out-of-domain WER overall, 0.401, is obtained by keeping the top 75 percent. In-domain WER stays close to the baseline across the sweep. In other words, on a large and noisy pool, removing the least confident pseudo-labels removes enough harmful noise to improve performance, while still leaving a large amount of useful data. The benefit shows up almost entirely on the out-of-domain test set, while the in-domain test set barely moves.

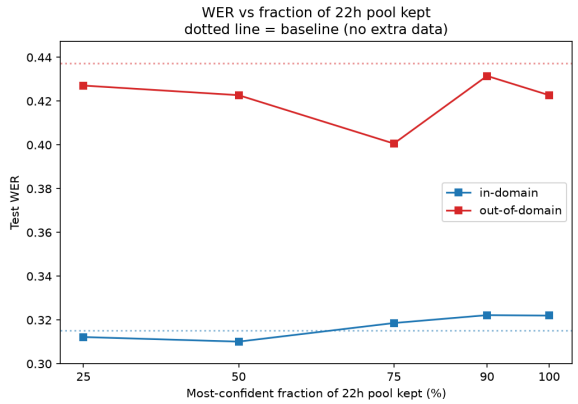

Figure 5.3: Phase 3. Test WER as a function of the most-confident fraction of the 22-hour pool that is kept. The dotted lines mark the baseline. Out-of-domain WER is lowest at the top-75 percent cut-off.

Table 5.2: Self-training results (test set). ID = in-domain, OOD = out-of-domain. Phase 1 filters the 168-min pool; Phase 2 scales the pool with new audio; Phase 3 filters the 22-h pool.

		WER		CER	
System	**Ph.**	**ID**	**OOD**	**ID**	**OOD**
Teacher (24 min)	—	0.334	0.454	0.084	0.135
Keep 100% (unfilt.)	1	0.315	0.437	0.078	0.128
Keep 90%	1	0.315	0.449	0.079	0.127
Keep 75%	1	0.314	0.443	0.078	0.128
Keep 50%	1	0.325	0.448	0.080	0.130
Keep 25%	1	0.336	0.449	0.084	0.130
+3 h	2	0.304	0.418	0.076	0.119
+10 h	2	0.307	0.412	0.075	0.115
+22 h (all)	2	0.322	0.423	0.080	0.116
22 h, top 90%	3	0.322	0.432	0.080	0.116
22 h, top 75%	3	0.319	0.401	0.078	0.112
22 h, top 50%	3	0.310	0.423	0.076	0.116
22 h, top 25%	3	0.312	0.427	0.078	0.119

6 Discussion and conclusion

This study asked to what extent language model integration and confidence-based self-training improve low-resource Gronings ASR, and how the size and noise of the data pool affect confidence filtering. Three conclusions follow.

First, an external n-gram language model is a robust and cheap improvement. Trained only on available written Gronings, it lowers WER by roughly 10 to 12 percent relative on both test sets, and the gain holds for both a weak and a strong acoustic model. The improvement is consistent and robust: it appears at every n-gram order. Its mechanism is also interpretable, since it repairs word boundaries and implausible spellings that greedy decoding produces. Because the language model is applied only at decoding time, it can be combined with any existing acoustic model at no training cost, which makes it a good first step for any low-resource variety that has some written material.

Second, more unlabelled audio helps, but only up to a point. A moderate amount (3 to 10 hours) of new out-of-domain audio produced the best in-domain self-training result, but adding the full 22 hours gave diminishing and even negative returns. For a low-resource setting this is an encouraging message, since a few hours of easily collected audio already help, but it also warns that indiscriminately piling on noisy data does not help. A similar pattern of diminishing returns was reported by Bartelds et al. (2023) when repeatedly adding self-labelled data.

Third, the value of confidence filtering depends on the pool. On the small, clean in-domain pool (Phase 1) filtering only traded away useful data and did not help. On the large, noisy out-of-domain pool (Phase 3) the same filtering removed enough harmful noise to yield the best out-of-domain result. Confidence filtering is therefore best understood as a trade-off between quality and quantity depending on the pool size and noise: it is not helpful when data are scarce and clean, and beneficial when data are plentiful and noisy. This reconciles the two intuitions that filtering should help because it removes errors, and that it would hurt because it removes data, what matters is how much noise there is to remove and how much data can be spared. A practical implication for low-resource ASR is that confidence filtering should be reserved for settings where the unlabelled pool is large and of uncertain quality since applying it to a limited pool is likely to waste data.

In conclusion, the two contributions answer the research question in complementary ways. The first part of the question, to what extent the two techniques improve Gronings ASR, is answered positively but with different profiles: the language

model gives a reliable and model-agnostic reduction of errors, while self-training gives a larger but more conditional improvement that depends on how much and what kind of extra audio is used. The second part, how the size and noise of the pool affect confidence filtering, has a more interesting answer: filtering is not a technique that is simply good or bad, but one whose usefulness is determined by the pool it is applied to. The language model can be adopted unconditionally, since it is cheap and never hurt performance in these experiments. Self-training should be adopted with judgement: collect a moderate amount of additional audio rather than as much as possible, and filter it by confidence only when it is plentiful and its quality is uncertain.

Limitations There are several limitations in this study. The test sets are small (213 and 71 utterances), so small WER differences should be read as trends rather than precise rankings, and every condition was trained once, without repeated runs. The three techniques, language model, confidence filtering and data scaling, were evaluated separately and never combined, so their joint effect is unknown. The in-domain pool (Phase 1) and the new audio (Phases 2 and 3) were pseudo-labelled by different models, which are the 24-minute teacher and the Phase-1 student respectively.

Future work The most natural next step is to combine the two contributions into a single pipeline: a student trained on a moderate amount of confidence-filtered out-of-domain audio, and then decoded with the language model. However, it cannot be assumed that the gains would simply add up. The language model largely repairs word boundaries and implausible spellings, and a better-trained acoustic model may already make fewer of exactly those mistakes, in which case the language model's benefit would be less. The reverse is also possible, since a stronger acoustic model produces better predicted transcriptions for the beam search to rank. The experiments reported here, which vary one component at a time, cannot distinguish these possibilities, and only a combined run would settle it. A further direction concerns the teacher–student coupling. Here the teacher was fixed within each phase (the 24-minute model in Phase 1, the Phase-1 student in Phases 2 and 3), so the unlabelled data influenced the student only through the pseudo-labels. A tighter scheme with a mean-teacher setup in which the teacher weights are updated as a moving average of the student's weights could let the unlabelled data influence the model more directly, at the cost of a higher risk of reinforcing the teacher's errors.

References

Babu, A., Wang, C., Tjandra, A., Lakhotia, K., Xu, Q., Goyal, N., ... others (2021). XLS-R: Self-supervised cross-lingual speech representation learning at scale. *arXiv preprint arXiv:2111.09296*.

Baevski, A., Zhou, Y., Mohamed, A., & Auli, M. (2020). wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in neural information processing systems*, *33*, 12449–12460.

Bartelds, M., San, N., McDonnell, B., Jurafsky, D., & Wieling, M. (2023). Making more of little data: Improving low-resource automatic speech recognition using data augmentation. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: Long papers)* (pp. 715–729).

Conneau, A., Baevski, A., Collobert, R., Mohamed, A., & Auli, M. (2020). Unsupervised cross-lingual representation learning for speech recognition. *arXiv preprint arXiv:2006.13979*.

Graves, A., Fernández, S., Gomez, F., & Schmidhuber, J. (2006). Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on machine learning* (pp. 369–376).

Hannun, A., Case, C., Casper, J., Catanzaro, B., Diamos, G., Elsen, E., ... others (2014). Deep Speech: Scaling up end-to-end speech recognition. *arXiv preprint arXiv:1412.5567*.

Heafield, K. (2011). KenLM: Faster and smaller language model queries. In *Proceedings of the sixth workshop on statistical machine translation* (pp. 187–197).

Kahn, J., Lee, A., & Hannun, A. (2020). Self-training for end-to-end speech recognition. In *Icassp 2020-2020 ieee international conference on acoustics, speech and signal processing (icassp)* (pp. 7084–7088).

Kensho Technologies. (2021). *pyctcdecode: A fast and lightweight Python-based CTC beam search decoder for speech recognition.* `https://github.com/kensho-technologies/pyctcdecode`. GitHub.

Silero Team. (2024). *Silero VAD: pre-trained enterprise-grade voice activity detector (VAD), number detector and language classifier.* `https://github.com/snakers4/silero-vad`. GitHub.

Xu, Q., Likhomanenko, T., Kahn, J., Hannun, A., Synnaeve, G., & Collobert, R. (2020). Iterative pseudo-labeling for speech recognition. *arXiv preprint arXiv:2005.09267*.