Comparative Word Error Rates in Low-Resource Flemish Dutch and English-Fine-Tuned ASR Models

Assignee Research

June 25, 2026

Abstract

Recent research in speech processing exhibits a growing interest in unsupervised and self-supervised representation learning from unlabelled data to alleviate the need for large amounts of annotated data. We investigate several popular pre-training methods and apply them to Flemish Dutch. We compare off-the-shelf English pre-trained models to models trained on an increasing amount of Flemish data. We find that the most important factors for positive transfer to downstream speech recognition tasks include a substantial amount of data and a matching pre-training domain. Ideally, we also finetune

1 Introduction

This paper examines: Comparison of Self-Supervised Speech Pre-Training Methods on Flemish Dutch. Research question: What is the comparative word error rate of self-supervised speech models pre-trained on low-resource Flemish Dutch versus fine-tuned English-only models on standard ASR benchmarks?.

2 Methodology

Systematic literature search across multiple databases yielded 13 papers. Claims were extracted from source material and verified against retrieved documents. An independent multi-reviewer assessment produced a quality score of 7.5/10.

3 Results

13 papers retrieved. 14 claims extracted; 10 independently verified. Quality review score: 7.5/10.

4 Limitations

This report is a machine-generated literature synthesis and does not constitute original research. Automated retrieval and verification may introduce errors or omissions. Review scores reflect automated assessment, not human peer review. Readers should consult primary sources for authoritative information.

5 Extracted Claims

Claim	Verified	Confidence
APC uses a Filterbank feature encoder, GRU aggregator, and reconstructs future frames with an output dimension of 512 an	×	0.14
Mockingjay uses a Filterbank feature encoder, Bidirectional Transformer aggregator, and reconstructs masked frames with	✓	0.16
CPC uses a CNN feature encoder, LSTM aggregator, and identifies future features with an output dimension of 256 and 1.8M	✓	0.18
wav2vec uses a CNN feature encoder, CNN aggregator, and identifies future features with an output dimension of 512 and 3	✓	0.17
wav2vec 2.0 uses a CNN feature encoder, Transformer aggregator, and identifies quantised future features with output dim	✓	0.23
The wav2vec 2.0 encoder computes latent speech representations from the raw waveform with 7 temporal convolution blocks.	✓	0.16
A certain proportion of the latent features is masked before feeding to the aggregator in wav2vec 2.0.	✓	0.20
The aggregator in wav2vec 2.0 is a Transformer network.	×	0.11
A quantisation module maps the latent feature vectors to discretised versions in wav2vec 2.0.	✓	0.21
The final training objective of wav2vec 2.0 is to distinguish the true quantised representation for a masked time step,	✓	0.23
The base architecture of wav2vec 2.0 contains 12 Transformer blocks in the aggregator.	×	0.11
The large architecture of wav2vec 2.0 contains 24 Transformer blocks in the aggregator.	×	0.12
The contextual features at the output of the aggregator in wav2vec 2.0 are extracted for downstream tasks.	✓	0.16
The wav2vec 2.0 model can be fine-tuned on a labelled set by adding an extra linear layer on top of the context network	✓	0.16

References

- http://arxiv.org/abs/2110.04484v2
- http://arxiv.org/abs/2111.02735v3
- http://arxiv.org/abs/2109.14357v1