# **Phoneme Error Rate Dynamics in Flemish Dutch Speech Recognition Across Self-Supervised Pre-Training Objectives**

Assignee Research

June 28, 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: How does the choice of self-supervised pre-training objective (e.g., contrastive, masked prediction, or latent diffusion) impact phoneme error rate on Flemish Dutch speech recognition when scaling the pre-training data volume?.

## 2 Methodology

Systematic literature search across multiple databases yielded 10 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

10 papers retrieved. 15 claims extracted; 11 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, a GRU aggregator, and has an output dimension of 512 with 4.1M parameters. | ✓ | 0.16 |
| Mockingjay uses a Filterbank feature encoder, a Bidirectional Transformer aggregator, and has an output dimension of 768 | ✓ | 0.18 |
| CPC uses a CNN feature encoder, an LSTM aggregator, and has an output dimension of 256 with 1.8M parameters. | ✓ | 0.18 |
| wav2vec uses a CNN feature encoder, a CNN aggregator, and has an output dimension of 512 with 32.5M parameters. | ✓ | 0.17 |
| wav2vec 2.0 uses a CNN feature encoder, a Transformer aggregator, and has output dimensions of 768 (base) and 1024 (larg | ✓ | 0.23 |
| wav2vec 2.0 combines ideas from wav2vec, vq-wav2vec, and MLM. | × | 0.14 |
| 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. | × | 0.10 |
| A quantisation module in wav2vec 2.0 maps the latent feature vectors to discretised versions. | ✓ | 0.19 |
| 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. | ✓ | 0.17 |
| 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 | ✓ | 0.15 |

## References


- http://arxiv.org/abs/2407.05862v1
- http://arxiv.org/abs/2402.06923v1
- http://arxiv.org/abs/2109.14357v1