Computational Linguistics in the Netherlands Journal 14 (2025) 525-553 Submitted 12/2024; Published 07/2025

Intrinsic evaluation of Mono- and Multilingual Dutch Language Models

Daniel Vlantis[*] DJVLANTIS@GMAIL.COM
Jelke Bloem[*] J.BLOEM@UVA.NL

[*] *Institute for Logic, Language and Computation, University of Amsterdam, Science Park 900, 1098 XH Amsterdam, Netherlands*

Abstract

Through transfer learning, multilingual language models can produce good results on extrinsic, downstream NLP tasks in low-resource languages despite a lack of abundant training data. In most cases, however, monolingual models still perform better. Using the Dutch SimLex-999 dataset, we intrinsically evaluate several pre-trained monolingual stacked encoder LLMs for Dutch and compare them to several multilingual models that support Dutch, including two with parallel architectures (BERTje and mBERT). We also try to improve these models' semantic representations by tuning the multilingual models on additional Dutch data. Furthermore, we explore the effect of tuning these models on written versus transcribed spoken data. While we can improve multilingual model performance through fine-tuning, we find that significant amounts of fine-tuning data and compute are required to outscore monolingual models on the intrinsic evaluation metric.

1. Introduction

Automatic evaluation of language models is no easy task. With the lack of available language experts to manually evaluate models, it becomes increasingly important to have a variety of evaluation benchmarks and procedures for a variety of languages and domains. In natural language processing (NLP), there are two predominant schools of thought when it comes to evaluating the quality of language models: evaluation based on results on downstream NLP tasks and evaluation based on the core functionality of the models themselves. These are respectively known as extrinsic and intrinsic evaluation.

In intrinsic evaluation, the quality of the language model's language representations is most commonly tested by comparing how similar its representations of two words are to similarity scores elicited from human raters. This is made possible by the continuous nature of word embeddings. Word embeddings store words in a numeric vector space, meaning relationships between words can be calculated by the similarity between the encoded vectors. Traditionally, the cosine similarity metric is used for this task. Words that are deemed dissimilar will have a low cosine similarity and vice versa. Gold-standard datasets are readily available for high-resource languages such as English.

One of the early pioneers of these semantic similarity datasets is Wordsim353 (Finkelstein et al. 2001). This dataset contained 353 word pairs alongside a similarity score which was obtained from the judgements of human raters. Since its release, many new datasets that improved upon Wordsim in both quality and quantity have been constructed. SimLex-999 (Hill et al. 2015) refined the initial design by separating the concepts of relatedness and similarity. Until recently there was no such dataset for evaluating the quality of Dutch word embeddings. Dutch SimLex-999, which is based on the English one but with human-translated items that are re-rated by native speakers, makes intrinsic evaluation of the semantic representations of Dutch language models possible (Brans and Bloem 2024).

Large-scale language models like BERT (Devlin et al. 2019) have gained popularity due to their ability to easily be fine-tuned to downstream NLP-tasks with good results (Bahdanau et al. 2015).

The large amounts of training data and transfer learning capabilities of these models allow them to be applied to unseen or under-resourced languages. This begs the question of their performance compared to language models that have solely been trained on one language.

Extrinsic evaluations of Multilingual BERT (mBERT) compared to monolingual variants have shown mixed results in previous work (e.g. Virtanen et al. (2019) for Finnish, de Vargas Feijo and Moreira (2020) for Portugese, de Vries et al. (2023) for Dutch). Even in under-resourced settings, where we might expect mBERT to perform better, monolingual models can outperform mBERT on POS-tagging and NER in some settings (Wu and Dredze 2020). For Dutch, on the Alpino corpus, monolingual BERT outperforms mBERT at a labeled dependency parsing task, but mBERT performs better with unlabeled dependencies and on part-of-speech tagging (Wu and Dredze 2020). The Dutch Model Benchmark (DUMB, de Vries et al. (2023)) shows that mBERT outperforms the monolingual BERTje at part-of-speech tagging, named entity recognition, word sense disambiguation and question answering, while performing worse in natural language inference, sentiment analysis, abusive language detection, causal reasoning and pronoun resolution.

In this paper, we address this question for Dutch from the perspective of intrinsic evaluation. We compare mBERT, a large-scale multilingual model, to Dutch BERT variants, using Dutch SimLex-999 as a benchmark. Furthermore, we investigate to what extent the performance of the multilingual models can be improved by tuning them on additional Dutch data, and how much of it is needed. Lastly, we investigate whether the use of different types of Dutch training data provide different benefits.

The findings here could motivate the continued production of monolingual language models if it is discovered that the results are significantly better. To the best of our knowledge, no intrinsic evaluation of monolingual and multilingual models for Dutch has been done to this end.

2. Background

Modern language models attach semantic representations to tokens, so for studying meaning representations of specific words in specific languages, it is important to be aware of how the tokenization process works. During tokenization, input tokens are split into word or subword units to be fed into deep language models. State-of-the-art models like BERT (Devlin et al. 2019) and RoBERTa (Liu et al. 2019) employ different tokenizers with several benefits and limitations. BERT uses a WordPiece tokenizer which determines statistically likely groups of adjacent characters and stores them into vocabulary as subword units (Schuster and Nakajima 2012). RoBERTa uses BPE (byte-pair-encoding), a greedy algorithm, similar to WordPiece, which merges the most frequent character pairs into a new symbol in the vocabulary (Sennrich et al. 2016). In more recent models such as XLM-RoBERTa (Conneau et al. 2020), the SentencePiece tokenizer (Kudo and Richardson 2018) gained popularity, which generalizes BPE to no longer attach special meaning to spaces, among other additions. This is relevant to languages that are written without spaces.

These subword tokenization methods help the language models deal with unseen words and remove the possibility of out-of-vocabulary errors. These tokenizers are pre-trained on a large sample of text, learning statistical groupings of characters that are common in that training data. These groupings often deviate from morphological segmentation of words in a language, such as splitting "redo" at the prefix boundary into "re" and "do". This may cause issues in downstream tasks such as translation (Ataman et al. 2017, Bauwens and Delobelle 2024), especially in polysynthetic languages (Mager et al. 2022). It has been shown that applying more morphologically informed forms of subword tokenization leads to lower perplexity for trained BERT and GPT variants on several languages, though not always with clear performance gains on downstream tasks (Hou et al. 2023). Morphologically rich languages, such as Turkish, benefit from a more granular tokenizer with more subword units (Kaya and Tantuğ 2024).

Large multilingual models like mBERT use a single tokenizer for all languages. This means that languages with large linguistic differences must be split according to the same rules learned by the tokenizer. This can lead to large model vocabularies and word-splitting/merging behavior that adheres to neither morphological nor statistical boundaries for under-resourced languages. For English, Artetxe et al. (2023) have shown that monolingual models outperform equivalent multilingual ones in the context of machine translation. Some research has shown that pre-trained monolingual tokenizers outperform their multilingual counterparts and suggests using language-specific adapted tokenizers in multilingual models to improve performance on downstream tasks (Rust et al. 2021).

Dutch is slightly more morphologically rich than English, particularly by having compound words. Remy et al. (2023) have proposed mapping tokens of high-resource monolingual models to tokens from a tokenizer of under-resourced languages in their tik-to-tok approach. Their approach especially addresses the issue of tokenizing multiword compounds in Dutch and German, and this approach was used to create the RobBERT-2023 language model for Dutch by mapping English subword tokens to Dutch using fasttext embeddings (Delobelle and Remy 2024). Bauwens and Delobelle (2024) propose the BPE-knockout approach to morphologically inform BPE tokenizers. Their approach increases the morphological adherence of existing BPE tokenizers by removing subwords from the tokenizer's vocabulary that do not adhere to morphological decompositions in a reference lexicon. They show that this increases model performance in token-level tasks.

2.1 Non-Contextual Word Embeddings

Word embeddings are vector representations of words used in language modeling as a vocabulary store. Word embeddings have come a long way since their genesis, where each word was individually coded. As vocabulary size increased, vectors became longer, making efficient storage difficult. Since then many different word embedding models have been introduced; one of the first notable models was Word2Vec (Mikolov et al. 2013b, Mikolov et al. 2013a). Word2vec is an unsupervised word embedding model that stores vocabulary as dense word vectors. It is cheap and efficient and scales well with vocabulary size. Word2vec does not include contextualized embeddings; ambiguous words only get a single vector and different senses cannot be analysed separately. For example, the meaning of the word "tear" can change based on the context but will always have the same vector representation.

2.2 Contextual Word Embeddings

One of the largest revolutions in language modeling tasks was the introduction of the transformer architecture. The self-attention mechanism of the transformer allows for more efficient handling of longer texts which lends itself naturally to all sorts of NLP tasks. This led to the development of language models with contextual word embeddings, in which different tokens of the same word type may have different numeric representations depending on its context, addressing the problem of representing ambiguous words and leading to better performance at most NLP tasks (Peters et al. 2018). One such contextual word embedding model is BERT (Bidirectional Encoder Representations from Transformers, Devlin et al. (2019)) which includes contextualized information about the word being encoded. BERT uses the transformer architecture, which includes a self-attention mechanism to weigh the importance of different words in an input. Research has found contextual word embeddings to be better suited for tasks pertaining to complex language structure, ambiguous word usage and unseen words (Arora et al. 2020).

BERT models are pre-trained on the masked language modeling (MLM) and next-sentence prediction (NSP) tasks. BERT models can be easily fine-tuned to many downstream NLP tasks such as sentiment analysis, question answering, and more, and in doing so, have obtained state-of-the-art results (Devlin et al. 2019). Building upon the success of BERT, the RoBERTa model was developed (Liu et al. 2019). The NSP task was found to be less beneficial than MLM and dropped from the pre-training process. The MLM method was tweaked so that masked tokens changed between training epochs. This is known as dynamic masking and allows the model to encounter more variability in the pre-training stage, leading to a more robust embedding model. Additionally, while BERT uses the WordPiece tokenizer (Schuster and Nakajima 2012) to split words into subtokens, RoBERTa uses byte-pair encoding (BPE, Liu et al. (2019)). More recently, ModernBERT (Warner et al. 2024) was developed, with a larger context length, larger-scale pretraining, more advanced positional encoding and SentencePiece tokenization.

2.2.1 Dutch-language models

For the most popular contextual word embedding models, Dutch-specific variants have been developed. BERTje (De Vries et al. 2019) is a BERT variant pretrained only on Dutch data, specifically 2.4 billion tokens of material from the SoNaR-500 (Oostdijk et al. 2013) corpus, Wikipedia and a few other sources. Its architecture is identical to BERT, having 110M parameters, 12 layers and a hidden size of 768. RobBERT is the Dutch equivalent of the English RoBERTa model (Delobelle et al. 2020), with more pre-training data and byte-pair encoding. It is trained specifically on Dutch portions of the OSCAR corpus, a large multilingual dataset gathered from Common Crawl (Ortiz Su'arez et al. 2020). RobBERT-v1 uses the English RoBERTa tokenizer, and RobBERT-v2 is based on a Dutch BPE tokenizer. Subsequently, it was updated as RobBERT-2022 on an update of the OSCAR corpus, and then RobBERT-2023. The 2023 version consisted of another pretraining dataset update but also the release of a Large variant with 355M parameters (similar to English RoBERTa-large), and a new Dutch tokenizer that uses the Tik-to-Tok method (Remy et al. 2023). This is the most recent Dutch contextual word embedding model that we are aware of, and its Large variant shows state-of-the-art performance on several Dutch NLP tasks (de Vries et al. 2023).

2.2.2 Multilingual models

As discussed in the introduction, with Dutch being a language with fewer resources than English, yet with a lot of typological similarity to English, multilingual models show competitive performance on various Dutch NLP tasks (de Vries et al. 2023). BERT received a multilingual version, mBERT, soon after its release (Devlin et al. 2019), based on the same architecture but with pretraining data from the 100 largest language editions of Wikipedia at the time, including Dutch. On the Dutch Model Benchmark, mBERT outperforms its nearest Dutch equivalent BERTje on the tasks of part-of-speech tagging, named entity recognition, word sense disambiguation and question answering, with BERTje performing better on the other five tasks in the benchmark.

A large multilingual version of Roberta, XLM-RoBERTa (Conneau et al. 2020), also gained widespread use after performing well on various NLP tasks, still based on the masked language modelling objective. Like RoBERTa, this model was trained on crawled web data, but for 100 languages instead of just English. The 295 billion token pretraining dataset includes 5 billion tokens of Dutch. On the Dutch Model Benchmark, XLM-RoBERTa outperformed monolingual Dutch models at part-of-speech tagging, named entity recognition, word sense disambiguation and abusive language detection (de Vries et al. 2023).

The DeBERTa architecture was also applied to the same multilingual pretraining data as XLM-RoBERTa to create mDeBERTa (He et al. 2023). This model was shown to outperform XLM-RoBERTa for many tasks. In the Dutch Model Benchmark, it did not reach state of the art in any task, but performed well on average. The English DeBERTa-v3-large outperformed all other models on the question answering task and was the second-best performing model on average, so we include it in our comparison even though it is technically an English monolingual model.

Similar to ModernBERT, EuroBERT (Boizard et al. 2025) aims to apply modern optimizations to encoder language modelling, also aiming to extend them to all European languages. The current version spans 15 languages and was trained on 5 trillion tokens. It does include Dutch with 50.6B tokens. This model again uses the masked language modelling pretraining objective, but with a subsequent annealing phase to adjust the data distribution towards that of higher-quality data, including non-English.

2.3 Intrinsic Evaluation

The aforementioned model comparisons are largely based on extrinsic evaluation - the performance of the models when applied to a different task than the pre-training task, such as question answering or sentiment analysis. For encoder transformer models, this typically involves tuning a classification head for the task to be evaluated on top of the model to be evaluated. Performance on these tasks is often evaluated using task-specific metrics and can demonstrate the ability of the language model to be tuned to specific downstream tasks. However, task-specific performance can vary strongly between models and depends on the tuning approach. When one is interested in applying a model to a new task or new domain for which no task-specific evaluation data exists, intrinsic evaluation of a model's lexical-semantic representations can at least provide evidence as to whether the model correctly represents word similarity in a language. It has been noted that intrinsic and extrinsic evaluation scores of word embedding models correlate poorly (Bakarov 2018).

For word embedding models, intrinsic evaluation means testing the quality of the semantic representations. This is often done with reference to human-rated benchmarks or with post-hoc evaluations of model outputs by human raters. Many methods have been proposed to evaluate the quality of language models intrinsically. Bakarov (2018) identifies and rates sixteen methods. They classify intrinsic methods into the following four classes: methods of conscious evaluation, methods of subconscious evaluation, thesaurus-based methods, and language-driven methods (Bakarov 2018). Our work focuses on word semantic similarity, a conscious intrinsic evaluation method whereby raters are given time to make informed decisions. In word semantic similarity, human raters judge the similarity of word pairs and assign a score, often from 1 to 5. These similarities are compared to the cosine similarity of the word embeddings to estimate the ability of the word embeddings to capture semantic meaning, using a correlation metric. Many datasets have been created to this end; earlier datasets like MEN (Bruni et al. 2012) and WordSim-353 (Finkelstein et al. 2001) have been improved upon by SimLex-999 (Hill et al. 2015), later extended to multilingual MultiSimLex (Vulić et al. 2020). SimLex separated the notions of relatedness and similarity. For example, the words 'coffee' and 'cup' are related but dissimilar. Despite this, they are rated more similar than 'car' and 'train' in datasets such as WordSim-353.

Word Semantic Similarity datasets are abundant for high-resource languages such as English but, until recently, did not exist for Dutch, and Dutch is not included in MultiSimLex. Recently, Dutch SimLex-999 was created (Brans and Bloem 2024) with the same word pairs as the English one, manually translated and/or culturally adapted to Dutch, and re-rated by Dutch native speakers.

2.4 Related Work

Several studies have compared and contrasted the intrinsic evaluation scores of language models, particularly in higher resource languages like English and Mandarin (Pranav et al. 2024), but also in Finnish (Venekoski and Vankka 2017), Polish (Mykowiecka et al. 2018) and many others (e.g. Vulić et al. (2020)). The majority of work comparing monolingual and multilingual language models comes from extrinsic evaluation. This is often done by evaluating model performance on different downstream NLP tasks. For example, Litake et al. (2023) compared the results of monolingual and multilingual BERT models on named entity recognition (NER) in Hindi and Marathi. They found that results depended on the language, with the monolingual MahaRoBERTa model performing best in Marathi and the multilingual XLM-RoBERTa performing best in Hindi. Despite these results, the authors claim monolingual models do not outperform multilingual models in this context. Similar work has been done in Portuguese by de Vargas Feijo and Moreira (2020), who found that monolingual models as a whole performed better, if only slightly, than multilingual models on a wide range of NLP tasks. These tasks included question answering, recognizing textual entailment, semantic textual similarity (sentencewise) and many more. The aforementioned Wu and Dredze (2020) focused on several under-resourced languages, but found mixed results regarding the two types of models. Also for Dutch, as mentioned before, such comparisons have been performed in the context of the Dutch Model Benchmark (de Vries et al. 2023) and in the context of the development of models such as RobBERT-2023 (Delobelle and Remy 2024), but always on extrinsic tasks.

2.4.1 Fine-tuning multilingual models

There has been some work on the effect that fine-tuning has on word embeddings. Zhou and Srikumar (2022) showed that tuning multilingual BERT models results in slight changes in the underlying embeddings of words. They argue that, while the spatial dimensions of embeddings remain largely similar, fine-tuning for specific tasks alters the embeddings slightly to capture the nuances of those tasks better. This is supported by other research as well. Cheong et al. (2021) studied how tuning multilingual BERT for code-switching between English and Chinese achieved better results on intrinsic evaluation and the downstream speech recognition task.

In the space of generative decoder LLM models, all major Dutch models were created by continued pretraining on predominantly English models, such as the 2.8B parameter Fietje model (Vanroy 2024), where Phi 2 was trained on 28 billion tokens of Dutch, and the 7.2B parameter GEITje-7B model (Rijgersberg and Lucassen 2023), which was based on the English Mistral-7B (Jiang et al. 2023) by training on the Dutch Gigacorpus and a webcrawling corpus. This shows that multilingual generative decoder LLM models can be improved by language-specific continued pretraining, but again this was mostly evaluated using extrinsic classification tasks (Vanroy 2024). We are not aware of any research on how the altered word embeddings obtained from fine-tuning multilingual stacked encoder models affect intrinsic evaluation scores.

2.4.2 Written versus spoken language

One of our ideas is to tune on transcribed spoken language rather than written language, as this may have a higher density of basic vocabulary compared to average crawled web text. There have been several studies on the differences between written and spoken languages. In Dutch, additional context on the speaker and subject can be derived on the basis of spoken language (Keune et al. 2005). It has also been found that there are significant differences in the linguistic phenomena between written and spoken Dutch. Research has shown that spoken Dutch tends to rely more heavily on using personal pronouns, among other methods, in subject-object ambiguous sentences, to disambiguate (Jansen 2005). It has also been shown that Dutch LSTM models better predict word associations (as opposed to homophone control words) when trained on 1M tokens of transcribed spoken data as opposed to written data (Bay and Bloem 2023). This study used the dataset of Drieghe and Brysbaert (2002), the only Dutch word similarity dataset available before Dutch SimLex-999. Knowing this, we can intuit that pre-training or tuning language models on transcripts of spoken text could lead to different representations of words.

3. Methodology

In this section, the experimental setup is described in detail. In Subsection 3.1, we describe the intrinsic evaluation. Here we introduce the pre-trained models used in our work. This section also explains how word embeddings are extracted from these models, how word similarity is calculated, and how we compare these results to our our evaluation set to get an intrinsic evaluation score. In Subsection 3.2, we describe the experimental setup used to finetune our language models. This includes language models used, hyperparameters, datasets on which they are tuned, and more.

3.1 Intrinsic Evaluation

To perform intrinsic evaluation, word embeddings that encode tokens in various language models must be extracted. Extraction methods differ between models, but as we only include BERT-based stacked encoder models, we are able to use similar evaluation pipelines (besides accounting

Model	**Type**	**Parameters**	**Layers**	**Data**	**Tokenizer**
mBERT	Multi	178M	12	?	WordPiece
BERTje	Mono	110M	12	2.4B	WordPiece
RobBERT-v2	Mono	117M	12	6.6B	BPE
RobBERT-2023-large	Mono	355M	24	19.6B	Tik-to-Tok
XLM-RoBERTa-large	Multi	561M	24	295B (5B)	SentencePiece
mDeBERTa	Multi	276M	12	295B (5B)	SentencePiece
DeBERTa-v3-large	English	304M	24	?	SentencePiece
EuroBERT-210m	Multi	210M	12	5T (50.6B)	BPE (Llama 3)
EuroBERT-610m	Multi	610M	26	5T (50.6B)	BPE (Llama 3)
EuroBERT-2.1B	Multi	2.1B	32	5T (50.6B)	BPE (Llama 3)

Table 1: Pretrained language models evaluated for monolingual vs. multilingual performance

for different special tokens that the different models use). We evaluate the ten model variants described in Table 3.1. This includes a BERT and RoBERTa-based model for both the monolingual and multilingual cases, although only mBERT and BERTje have parallel architectures for direct comparison. While their training data may differ, their intrinsic evaluation scores could inform a conclusion about the efficacy of domain-specific training data rather than large multilingual training sets.

We use the Hugging Face transformer package (Wolf et al. 2020) to extract word embeddings from all models. We benchmark the cosine similarities of word pairs from the Dutch SimLex-999 benchmark (Brans and Bloem 2024) against human similarity ratings from that benchmark.

Each word in each pair is embedded without any context, following Brans and Bloem (2024), adding only the necessary special tokens for each model using *add_special_tokens=True*. While this does not take advantage of the contextual capabilities of contextual word embedding models, it is more faithful to the task that human raters got and enables comparisons to static (non-contextual) word embedding models such as Word2Vec.

Words in the benchmark may be subtokenized into several subword units by the various models that each have their own representations. Therefore, we take the average over all subtoken embeddings (special tokens excepted) as our word embedding for evaluation. The embedding is taken as the average over all tokenized embeddings. This is done since, during tokenization, words can be split into several subword units that are encoded separately.

Next, we calculate the similarity between our paired word embeddings using the cosine similarity metric:

$$\text{cosine similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} \tag{1}$$

This returns a score between -1 and 1 for each word pair in Dutch Simlex-999. A high score indicates highly similar words, and a low score, the opposite. Scores typically do not go far below 0 as the models are not optimized to encode non-association or antonymy. Based on these results, we generate a correlation score between the model-predicted, and human-rated similarity scores in Dutch SimLex-999. This is done using the Spearman rank correlation coefficient ρ:

$$\rho = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)} \tag{2}$$

where n is the number of data points and d_i is the difference between the ranks of the ith element in the dataset. We calculate ρ for each layer of each model, allowing us to gauge the models performance over several layers. Prior research has suggested that embeddings extracted from earlier layers have a stronger correlation to contextless human semantic similarity ratings (or benchmarks of lexical semantics in general) and that the final contextual layer has a weaker correlation (Bommasani et al. 2020, Brans and Bloem 2024).

3.2 Fine-tuning

Next, we aim to investigate the effect that fine-tuning language models has on their intrinsic evaluation scores. We do this for several scenarios. First, we fine-tune language models and explore how the quantity of tuning data affects the scores achieved. Second, we compare how the medium of tuning data affects intrinsic evaluation score. To do so, we finetune models on tuning sets from written sources and transcribed, spoken sources.

Our tuning data comes from the SoNaR (Oostdijk et al. 2013) and CGN (*Corpus Gesproken Nederlands*, Spoken Dutch Corpus) (Oostdijk 2000) corpora. The SoNaR corpus contains over 500 million words of Dutch written text from the Netherlands and Flanders, obtained from various sources. These include native sources but also sources translated by experts. The second corpus we use, CGN, contains over 900 hours of transcribed audio text. This amounts to approximately 9 million words. The sources of the transcriptions come from interviews, news broadcasts, and more from both the Netherlands and Flanders. Both datasets are described in Table 3.2.

Corpus	**Domain**	**# Words**
Corpus Gesproken Nederlands (CGN) (Oostdijk 2000)	Spoken	9M
SoNaR Corpus (Oostdijk et al. 2013)	General	500M

Table 2: Corpora used for fine-tuning multilingual language models

We tune the mBERT, XLM-RoBERTa, mDeBERTa and EuroBERT-2.1B models to discover the effect that tuning these models on additional Dutch data has on intrinsic evaluation scores, if any. To do so, we created several subsets of the SoNaR corpus. The subsets contain approximately 10k, 100k, 1M, 10M and 100M words. These subsets were created by randomly drawing sentences from the SoNaR corpus until the threshold number of words was reached. Various preprocessing steps were also taken to remove HTML and non-text information in the corpus files.

Once the subsets are created we proceed to fine-tuning the models. This is done on the masked language modeling (MLM) task, where random words in the textual input are replaced with a MASK token, and the language model must predict which words fit in their spot. This is done using the 'transformers' package and creating a pipeline which loads the models and their tokenizers. The pipeline then tokenises the tuning data, masks 15% of words, and performs continued training on the MLM task. The tokenizers are not affected by our fine-tuning process. The hyperparameters for fine-tuning all models are kept consistent for fair comparison and can be seen in Table 3.

Next, we investigate the effect of the medium of the tuning data on intrinsic evaluation scores. We hypothesize that the relationship between words embedded by the model fine-tuned on spoken language will more closely resemble that of the Dutch SimLex-999 human evaluation set. Meaning that they will score higher on the intrinsic evaluation task, and thus provide a more suitable base for training or tuning than written text.

To explore this, we finetune the mBERT, XLM-RoBERTa (base and large), mDeBERTa and EuroBERT-2.1B models on written and spoken text. This too, is done on the masked language modeling (MLM) task, where words in sentences are masked, and the language model must predict the masked words. For the written text model, we used the variants of the previously defined models that were tuned on the 1M word subset of the SoNaR corpus. For the spoken model, we created an additional 1M word subset, this time from the CGN corpus. One issue in distinguishing spoken and written language is that language may be written to be spoken. This type of data may be less representative of spontaneous speech. Therefore, for the CGN subset, we have only sampled from

Hyperparameter	**Value**
Epochs	3*
Learning Rate	5e-5
Weight Decay	0
Train Batch Size	64
Optimizer	adamw
adam_beta1	0.9
adam_beta2	0.999
adam_epsilon	1e-8

Table 3: Hyperparameter settings for all fine-tuned models. * EuroBERT was only trained for 1 epoch on the largest tuning dataset due to resource restrictions.

corpus components A to I, which include more spontaneous forms of speech such as face-to-face conversations, interviews, business negotiations and sports commentary. For a full overview of these components, see Appendix E. SoNaR also includes written-to-be-read components, but these are relatively small and we consider them similar to written language, so they were not excluded from our sampling.

Again, we randomly selected sentences from the chosen components of the corpus until we reached an equal number of token as in the SoNaR 1M sample (respecting sentence boundaries). Additionally, transcription-specific cleaning steps had to be taken here to ensure data quality was not impacted. This means removing transcript-specific code data such as [UNKNOWN]-, [ggg]-, and [xxx]-tokens, and more. These codes were originally added for annotation purposes but are not necessary for tuning. A complete list of the filtered codes is included in Appendix F.

We are left with 30 tuned models (5 models x 6 data samples). All of these finetuned models undergo the same intrinsic evaluation that the pretrained language models underwent in Section 3.1. For each word pair in Dutch SimLex-999, each word is embedded separately, similarity scores are calculated between the words of the pair for each layer, and a correlation score per layer is calculated. We report results for the first layer and the best-scoring layer, which may differ per model.

4. Results

In Section 4.1, we present the intrinsic evaluation scores obtained by the pre-trained monolingual and multilingual language models. In 4.2, we explore the results obtained from mBERT, XLM-RoBERTa, mDeBERTa and EuroBERT-2.1B when finetuned on additional Dutch text of varying amounts. In Section 4.3, we present the results of the medium study, where we compare the performance of these four multilingual models when finetuned on written vs spoken Dutch. Finally, in section 4.4, we present some additional results related to tokenisation.

4.1 Pretrained Model Results

Using the Dutch Simlex-999 evaluation data we were able to benchmark the intrinsic evaluation scores obtained by the models in Table 3.1. For each model, we calculate the Spearman correlation between similarity scores obtained by our model and those reported in Dutch Simlex. This is done for all layers of the model. In Table 4, we show the correlation score for the first layer and the best-performing layer of each model.

Model	**First Layer**	**Highest Score**	Best Layer	Layers
BERTje	**0.396**	**0.396**	1	12
RobBERT-v2	0.179	0.188	5	12
RobBERT-2023-large	0.157	0.157	1	24
mBERT	0.135	0.137	5	12
XLM-RoBERTa	**0.184**	0.184	1	12
XLM-RoBERTa-large	0.141	0.180	3	24
mDeBERTa	0.178	**0.313**	6	12
DeBERTa-v3-large	0.100	0.116	9	24
EuroBERT-210m	0.037	0.052	2	12
EuroBERT-610m	0.029	0.034	4	26
EuroBERT-2.1B	0.058	0.058	1	32

Table 4: Peak results achieved by all pretrained models. Highest mono- and multilingual scores are highlighted in bold.

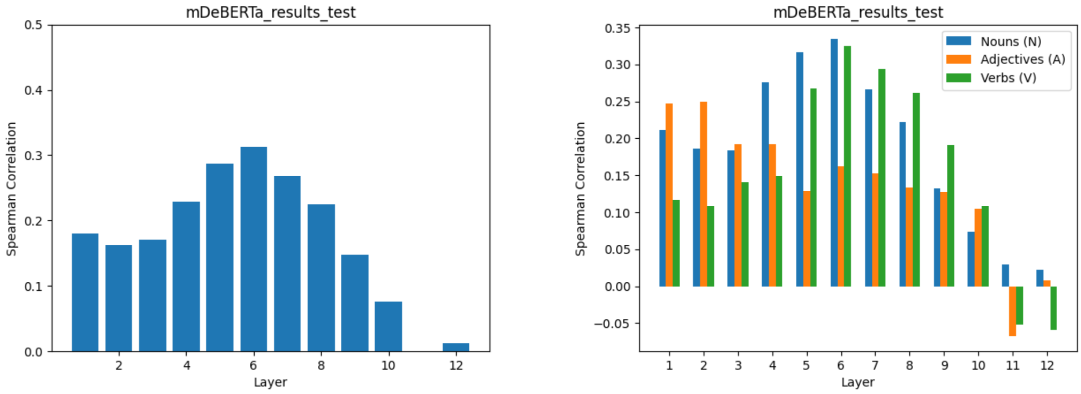

Figure 1: Aggregate and POS-based layer-wise intrinsic evaluation scores for mDeBERTa model

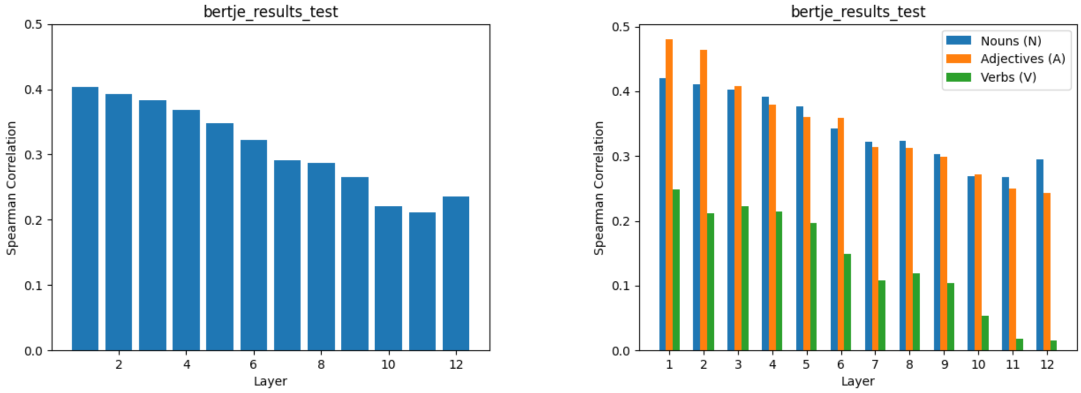

Figure 2: Aggregate and POS-based layer-wise intrinsic evaluation scores for BERTje model

In Figure 1, we see the scores obtained by the mDeBERTa model, the best-performing pretrained multilingual model, and in Figure 2 we see BERTje, the best-performing pretrained monolingual model. Similar figures for other models can be found in Appendix C. The same BERTje result is discussed by Brans and Bloem (2024), who note the best performance in the first layer. As for mDeBERTa, in contrast to BERTje and most other pretrained models, middle layers (4 through 8) achieve the highest aggregate correlation score. As expected, the contextualized layers score worst, with layer 11 being both the lowest and the only negative scoring layer. The reported intrinsic evaluation scores of the pre-trained models also reveal interesting results.

To derive more useful insights from the reported scores, we separate these results by part-of-speech label. This allows us to judge the performance of each model on different word types. The POS evaluation scores from Figure 2 show a fairly regular pattern, with nouns and adjectives performing similarly and verbs trailing behind while all categories decline across layers. Patterns in mDeBERTa are more varied, with adjectives scoring best in the initial layers, while verbs and nouns peak in the middle layers. Interestingly, verbs have nearly as good correlations with human judgements as nouns here, while usually language models struggle more with verbs in this task.

In Table 4, we see that in both the RoBERTa and the BERT cases, the monolingual models scored higher than their multilingual variants on this task. For the BERT models, BERTje achieves a peak score at the first layer with a Spearman correlation of 0.396, compared to mBERT's peak of 0.135 at layer 1. The results of the RoBERTa models are similar, with the monolingual RobBERT model achieving a peak score of 0.179 compared to XLM-RoBERTa's 0.141. Interestingly, the older RobBERT v2 with the BPE tokenizer scores better than the newer 2023 version with the Tik-to-Tok tokenization approach here.

We can also observe that the smaller but multilingual mDeBERTa outperforms the larger English DeBERTa-v3-large, even though on the Dutch Model Benchmark (de Vries et al. 2023), DeBERTa-v3-large outperforms on five out of nine tasks. As noted by de Vries et al. (2023), this may be due to English interference from the automatic translation involved in creating some of the task-specific benchmarks, making a largely English model perform well. Lastly, we see that the EuroBERT models perform very poorly on this benchmark, despite being the most modern and largest ones.

4.2 Fine-tuning and ablation study

In this section, we describe the results obtained from the fine-tuning study, where we tune mBERT, XLM-RoBERTa-large, mDeBERTa and EuroBERT-2.1B on increasing quantities of Dutch sentences from the SoNaR corpus. The models were fine-tuned on the MLM objective, with 15% of words being masked. We tested the following amounts of additional training data: 10k, 100k, 1M, 10M and 100M tokens. All models were tuned separately, so it is not the case that the 100k model is a continuation of the 10k model. Every model used the same sample of text for every data size, so it is not the case that a different 10k tokens were sampled each time. The peak intrinsic evaluation scores over all layers for each fine-tuned model are presented in Figure 3, and with numbers for the largest tune of 100M tokens in Table 5.

As we would expect, we see that the peak scores achieved by each tuned model increases with the size of the tuning data. For mBERT, the strongest correlation was achieved by the model trained on the 100M token subset of the SoNaR corpus, which achieved a peak score of 0.298, a significant improvement over all the other SoNaR-tuned mBERT models and an improvement of approximately 0.161 over the peak result obtained by the pre-trained mBERT model. Nevertheless, this score trails behind the comparable monolingual model of BERTje, which achieved a correlation of 0.396.

We can also see that more modern versions of the architecture struggle to match this result with this amount of tuning data, with only mDeBERTa and XLM-RoBERTa-large outperforming

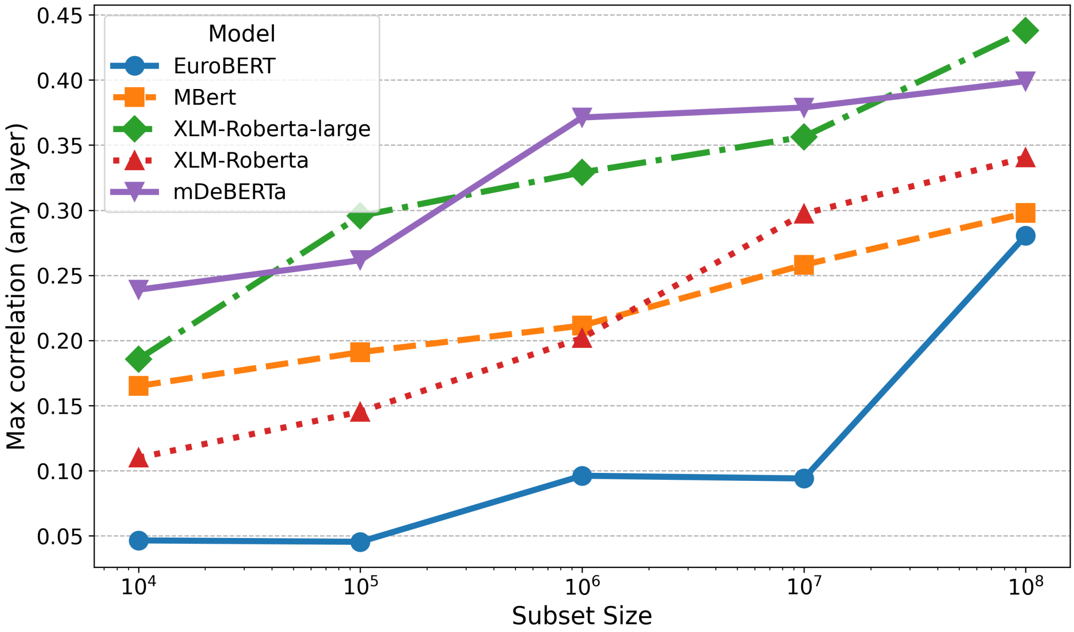

Figure 3: Peak (best layer) correlation scores per model per ablation

Model	First Layer	Highest Score	Tune-diff	Best Layer	Layers
mBERT	0.206	0.298	+0.161	6	12
XLM-RoBERTa	0.224	0.341	+0.157	5	12
XLM-RoBERTa-large	0.181	**0.438**	**+0.281**	14	24
mDeBERTa	0.190	0.399	+0.086	7	12
EuroBERT-2.1B	0.070	0.281	+0.223	9	32

Table 5: Peak results achieved by all models tuned on 100M tokens of written Dutch from the SoNaR corpus. The EuroBERT-2.1B model was tuned for one epoch due to resource limitations, the others for three. Tune-diff is the absolute difference in correlation score between the untuned model and the tuned model

Model	Data	**First Layer**	**Highest Score**
mBERT	Written	0.157	0.211
mBERT	Spoken	0.173	**0.222**
XLM-RoBERTa-large	Written	0.153	0.329
XLM-RoBERTa-large	Spoken	0.169	**0.351**
mDeBERTa	Written	0.178	**0.371**
mDeBERTa	Spoken	0.166	0.305
EuroBERT-2.1B	Written	0.057	**0.096**
EuroBERT-2.1B	Spoken	0.013	0.082

Table 6: Results for models tuned on an equivalent amount of written or spoken data

BERTje. mDeBERTa, a model with the same number of layers and only around double the parameters of BERTje, barely outperforms it with a correlation score of 0.399 on its layer 7 when tuned with 100M tokens. XLM-RoBERTa-large also only outperforms BERTje when tuned with 100M tokens of data, but with a solid correlation score of 0.438 on layer 14, the highest in our study. This model is significantly larger than BERTje, however.

The base-sized XLM-RoBERTa shows similar results to mBERT, with a more rapid rise in performance when tuned on more data, and a peak correlation of 0.340 on the 100M subset. Tuning had a similar impact on the intrinsic evaluation score as in mBERT's case, with an improvement of 0.157 in terms of absolute correlation score over the pre-trained XLM-RoBERTa model compared to mBERT's 0.161. XLM-RoBERTa-large showed the largest absolute improvement from tuning on 100M tokens of written Dutch text with 0.281.

The case of EuroBERT is interesting – despite being the largest and most modern model, its base model performs the worst and requires a significant amount of fine-tuning to reach a reasonable correlation with human semantic similarity ratings. With 100M tokens of fine-tuning data, it is still outperformed by all other tuned multilingual models, though we were only able to tune it on this data sample for one epoch instead of three. Due to the large scale of this model, further improvements can be expected with significantly more tuning data.

In all cases, the results obtained suggest that we can improve the intrinsic evaluation scores of multilingual models by tuning on additional Dutch data, but quite a lot of data is required. Tuning these models on 100M tokens for three epochs also comes at a significant computational cost, requiring several days to a week when using a single modern high-end GPU (and longer for EuroBERT).

4.3 Media Study

In this section, we present the results of the media study. Here, we evaluate pairs of tuned mBERT, XLM-RoBERTa, mDeBERTa and EuroBERT-2.1B models. These models are tuned on either written or spoken Dutch data taken from the SoNaR and CGN corpus, respectively. The models are tuned on the MLM tasks, with 15% of words being masked.

These results are shown in Table 6. We see inconsistent results between models. For mBERT and XLM-RoBERTa-large, the model tuned on spoken data performs slightly better, but for the best multilingual model mDeBERTa, the model tuned on written data is clearly better. For EuroBERT, the written data model is also slightly better, although this study would probably have to be done with 100M tokens of tuning data or more for reliable results. A layer-wise comparison for mDeBERTa, the model with the largest difference between spoken and written, is shown in Figure 4. For the other models, the layer-wise patterns also look similar, though XLM-RoBERTa-large loses more performance in the last layers when tuned on written data than when tuned on spoken data.

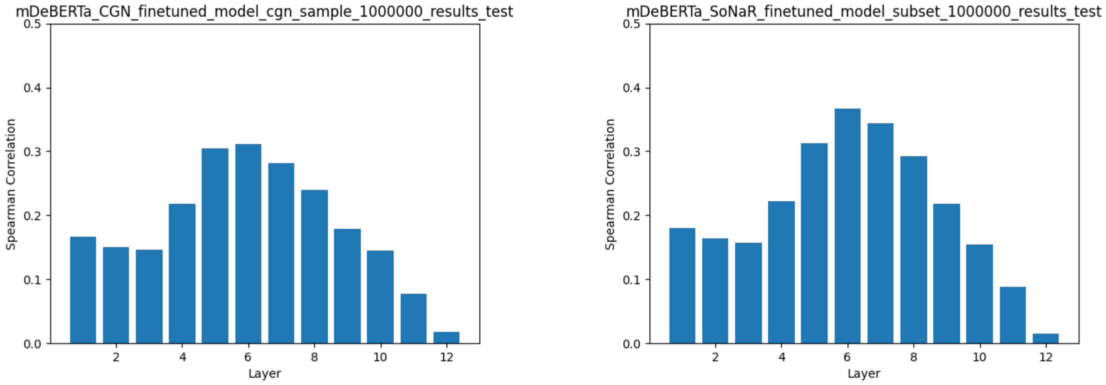

Figure 4: Layer-wise Dutch SimLex-999 correlation scores of mDeBERTa when tuned on written (SoNaR) versus spoken (CGN) Dutch data.

Model	Input	Tokenization
mBERT	razendsnel	[‘raz’, ‘##end’, ‘##sne’, ‘##l’]
BERTje	razendsnel	[‘razendsnel’]
RobBERT-v2	razendsnel	[‘ra’, ‘zen’, ‘ds’, ‘nel’]
RobBERT-2023	razendsnel	[‘razendsnel’]
XLM-Roberta	razendsnel	[‘raz’, ‘end’, ‘s’, ‘nel’]
mDeBERTa	razendsnel	[‘raz’, ‘end’, ‘s’, ‘nel’]
DeBERTa-v3-large	razendsnel	[‘raz’, ‘ends’, ‘nel’]
EuroBERT	razendsnel	[‘raz’, ‘ends’, ‘nel’]

Table 7: Manual inspection of tokenization of Dutch input

4.4 Tokenisation

Our results indicate that pre-trained monolingual language models outperform multilingual models in terms of intrinsic evaluation, unless they are tuned with significant amounts of additional Dutch data.

Several potential obstructions prevent these models from achieving high intrinsic evaluation scores. One such obstruction is the tokenizer. An important benefit of training language models on a monolingual corpus is the performance of the tokenizer. Tokenizers in multilingual models such as mBERT need to be one-size-fits-all. This means that they should be able to create effective subword units in several languages at the same time. The problem with this is that languages with significant linguistic differences need to be tokenized by the same tokenizer. For example, in the Dutch language, compounding is used to add meaning. Take the word ‘razendsnel’ (English: lightning fast). It comprises two Dutch words compounded: ‘razend’ (English: furious or enraged) and ‘snel’ (English: fast or quick). Splitting this word at the compound word boundary during tokenization might be beneficial for effective meaning representation (Bauwens and Delobelle 2024), and as we discussed in section 2, such benefits have been observed in the literature for various morphologically rich languages.

Language models like RobBERT, which are solely trained in Dutch, may have tokenizers that are better attuned to the morphological structure of Dutch. This could explain why, in our results, monolingual models outperform multilingual models on the intrinsic evaluation score. This is the case in both RoBERTa and BERT models. To verify this claim, we perform a small manual investigation into the tokenization of words by the four pre-trained models.

In Table 7, we see how the different models tokenize this particular example. There are significant differences between the model strategies for this task. BERTje, the model with the highest intrinsic evaluation score, represents the entire word as one token. While this tokenization strategy can lead to good scores, it is not as efficient at scale. With this technique the words need to be stored individually in vocabulary which scales poorly with vocab size, additionally, it makes it more difficult for the model to handle unseen words.

Language models employ sub-wording algorithms, like WordPiece or BPE, to split words into bytes. This can be seen in the tokenization of 'razendsnel' (lightning fast) in Table 7. This is an important feature, as adding different languages can cause the vocabulary size to explode. Storing subword bytes allows you to reconstruct words using common subword units. However, the subword units generated by the tokenizers are, therefore, trained on data from several languages. This leads to the creation of common subword units that may not necessarily correspond to any useful semantic subword units in Dutch. Meaning can be lost as a result. We can see that none of the major models tokenize 'razendsnel' morphologically into 'razend' and 'snel'. We list some further examples in Appendix D.

4.4.1 Compound words experiment

One interesting observation from the manual inspection is that the multilingual models seem to tokenize shorter words according to morphological boundaries but struggle with compound words. To test whether this property of the multilingual model's tokenizer affects the reported intrinsic evaluation score, we divided the Dutch Simlex-999 benchmark into two subgroups. These groups are determined by the number of compound words in the word pairs: 0, or 2. We define a compound word as a word that consists of two Dutch words that can each exist independently. For example, 'huiswerk' (homework) is a compound word because 'huis' (house) and 'werk' (work) can both exist on their own.

An example of a pair with no compounding from the dataset is 'boek-informatie' (book-information), examples of pairs with one compound word are 'verjaardag-jaar' (birthday-year) and 'nemen-achterlaten' (take-leave_behind), and examples of pairs with two compound words are 'vliegtuig-luchthaven' (airplane-airport), 'aanmoedigen-ontmoedigen' (encourage-discourage) and 'uitgang-deuropening' (exit-doorway). The categories are of course quite imbalanced — there are 846 pairs with no compound words, 129 pairs involving one compound word and 24 pairs with two compound words. In the 24 pairs with two compounds, seven are pairs of particle verbs, two are pairs of adjectives and the others are nouns. These pairs include both highly related and unrelated words, and of course the words involved are longer than average.

For each of the pre-trained models from Table 3.1, we calculated intrinsic evaluation scores for each compound word group subset of Dutch Simlex-999. The results are visualized in Figure 5. We observe that for BERTje, mBERT and XLM-RoBERTa-large, performance on pairs with 2 compounds is worse — two models with WordPiece tokenization and one SentencePiece model. However, mDeBERTa, which tokenizes quite similarly to XLM-RoBERTa, does not show this effect. Furthermore, EuroBERT and DeBERTa-v3-large show an opposite pattern where 2 compound words perform better. These two models also tokenize similarly and their tokenizers appear largely influenced by English (DeBERTa-v3-large being a monolingual English model) so it is interesting that they perform better on pairs of Dutch compound words.

Similar results for the tuned models are shown in Figure 6. After tuning, EuroBERT and mDeBERTa perform worse at 2 compound word pairs, despite overall performance gains. EuroBERT, mBERT and XLM-RoBERTa-large score worse on pairs of compound words than on pairs of non-compound words. Since the tokenizers do not change during tuning, this could be due to limitations

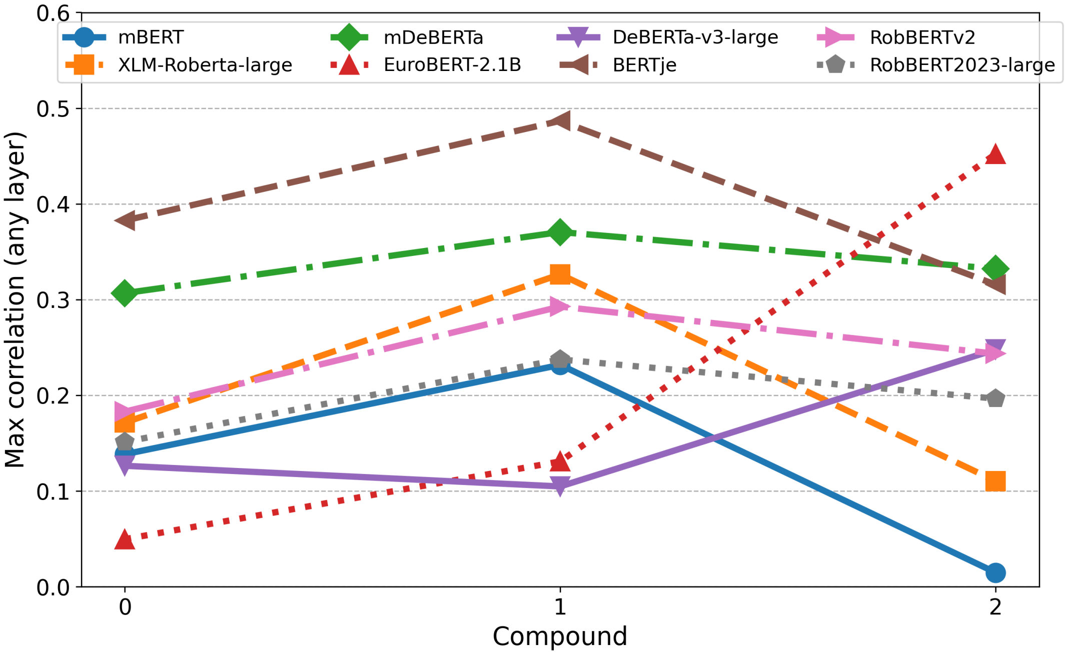

Figure 5: Peak (best layer) correlation scores per model per compound size

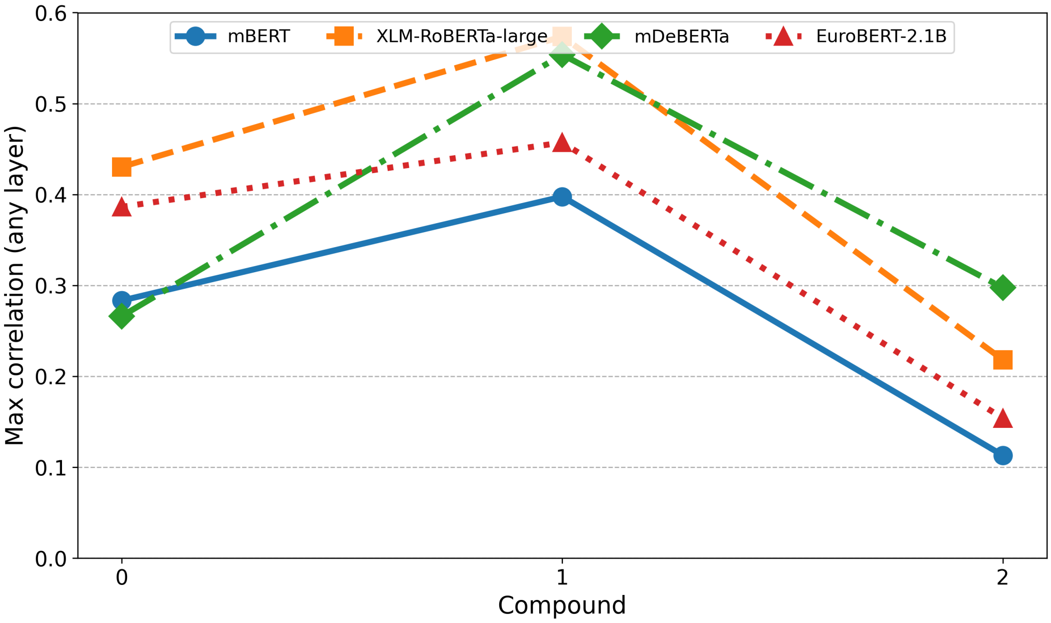

Figure 6: Peak (best layer) correlation scores per model per compound size, for models tuned on 100M tokens from the SoNaR corpus

Model	Input	Tokenization
RoBERTa-BPE-39k	razendsnel	[’razendsnel’]
RoBERTa-BPEko-39k	razendsnel	[‘razend’, ‘snel’]
RoBERTa-BPE-30k-BPEko-9k	razendsnel	[‘razend’, ‘snel’]

Table 8: Tokenization of Table 7 example with BPE-knockout

Model	First Layer	Highest Score	Best Layer	Layers
RoBERTa-BPE-39k	-0.015	**0.005**	5	12
RoBERTa-BPEko-39k	-0.042	-0.026	6	12
RoBERTa-BPE-30k-BPEko-9k	-0.042	-0.032	7	12

Table 9: Results for models from the BPE-knockout paper (Bauwens and Delobelle 2024)

of the multilingual tokenizers. This further investigation somewhat suggests that performance gains from language-specific tuning of multilingual models are limited by the multilingual tokenizer. However, these results are inconclusive and for clearer evidence, an experiment with a controlled dataset of compound words of different complexity would be needed.

4.4.2 BPE-knockout experiment

The BPE-knockout approach of Bauwens and Delobelle (2024) aims to make tokenization more morphological by pruning trained BPE tokenizers to remove subtoken merge steps that result in subtokens that span morpheme boundaries. This results in better adherence to derivational and compound boundaries during tokenization, and might be a solution for the lack of language-specific morphological information that multilingual tokenizers have.

As well as English and German, Bauwens and Delobelle (2024) test their approach on small Dutch monolingual RoBERTa models and show better performance on downstream tasks with those models, both when pretrained with a BPE-knockout tokenizer (RoBERTa-BPEko-39k) and when taking a model pretrained with a regular tokenizer and tuning it with BPE-knockout (RoBERTa-BPE-30k-BPEko-9k). In Table 8, we see that on our ‘razendsnel’ example, these models perform the expected morphological tokenization, unlike all models from Table 7. RoBERTa-BPE-39k is Bauwens and Delobelle (2024)’s base model without BPE-knockout.

In Table 9, we also evaluated these models on Dutch SimLex-999, but found that the base BPE model outscores the BPE-knockout models and that correlations are near-zero. The general poor performance is likely to be caused by the small amount of training data used by Bauwens and Delobelle (2024). It would be very interesting to evaluate whether the BPE-knockout approach can improve the score of multilingual models or tuned multilingual models on the Dutch SimLex-999 benchmark, but none of the popular multilingual models use a regular BPE tokenizer (see Table 3.1). In future work, it would be interesting to adapt the BPE-knockout approach for SentencePiece tokenizers, which also often have a BPE component, to evaluate whether language-specific adaptation of the multilingual tokenizer can improve the quality of semantic representations for a target language.

5. Discussion

We evaluated a number of pretrained models on a semantic similarity benchmark for Dutch. The results of this study are intended to motivate the continued development of monolingual or multilingual models depending on the obtained results. We find that monolingual models outperform multilingual models in this intrinsic evaluation, and multilingual models require a significant amount of fine-tuning to catch up to a base-BERT-equivalent monolingual model on the Dutch SimLex-999 intrinsic evaluation benchmark. In our setup with up to 100M tokens of fine-tuning data, only mDe-

BERTa and XLM-RoBERTa-large managed to catch up, with mDeBERTa being more optimized and XLM-RoBERTa-large being bigger. The recent EuroBERT-2.1B model, while officially supporting the Dutch language, shows quite poor results on semantic similarity, except when tuned on larger amounts of data.

5.1 Fine-tuning and ablation study

While the pre-trained model results indicate that monolingual models outperform multilingual models, it is not always a practical solution to train a domain-specific and language-specific model for every situation. Additionally, the transfer learning capabilities of these models mean that very low-resource languages could benefit from some learned semantic knowledge obtained from other languages. Therefore, we investigated whether fine-tuning large-scale pre-trained multilingual language models on domain-specific data could provide similar results as their monolingual counterparts.

Our results indicate that tuning on additional Dutch data does improve the intrinsic evaluation scores of both models, and it is possible to surpass the performance of BERTje with 100M tokens of fine-tuning data while 2.4B tokens were used to train BERTje. However, this was done with more modern architectures, for which there are no Dutch pre-trained equivalents. To establish exactly how much tuning data is necessary for a multilingual model to surpass an equivalent monolingual model, a larger-scale follow-up experiment would be required in which a Dutch model is pretrained on large quantities of data with the same setup as mDeBERTa or XLM-RoBERTa. Nevertheless, our results are promising especially for more under-resourced languages where the amount of available data is sufficient for fine-tuning a multilingual model but not for pre-training a monolingual model.

As expected, the results improved with the quantity of fine-tuning data used. In future work, we would like to see if this trend continues by increasing the quantity of fine-tuning data even further to be similar to the amount of data that monolingual models are pretrained with. In particular, for larger-scale models such as EuroBERT, it might be necessary to tune with much larger volumes of data to bring out reasonable performance. This parallels the approach taken for Dutch decoder language models such as Fietje (Vanroy 2024).

5.2 Media Study

Equally, if not more important than data quantity is data quality. Traditionally language models are trained using written text from large corpora, scraped from the web, documents, books, and more. This is the case since this data is easier to collect and requires little transformation and formatting adjustments. Whether this text provides a better base for training language models in Dutch is unclear. Thanks to projects like the CGN (Oostdijk 2000), we can access large amounts of transcription data, including many hours of transcribed text, formatted and annotated. Thus, we can investigate the effects of tuning multilingual models on written vs spoken text.

To this end, we fine-tuned the multilingual models on equally sized subsets (10000 words) of written and spoken text. We found mixed results: mBERT and XLM-RoBERTa-large did better with spoken data, mDeBERTa and EuroBERT did better with written data. These results suggest that older and smaller models benefit more from transcribed spoken data, though this is speculative. If there is any such effect, it is model-dependent, and a more controlled study would be needed to figure out what aspects of model architecture affect this, such as the tokenizer, the type of pre-training data used or the language mix in the multilingual model. We do observe a clear difference when we compare the type-token ratios of our 1M sample of the SoNaR corpus and of the CGN. Excluding punctuation SoNaR has a TTR of 0.104 while the CGN sample has 0.044, indicating a larger diversity of vocabulary in the written text.

It is possible that spoken data is more beneficial for tuning smaller models, as spoken language is also what humans initially learn from in their language acquisition process. It may be more similar to child-directed speech, which is also used to train language models more efficiently, e.g. in the

BabyLM challenge (Hu et al. 2024). It also better reflects what native speakers would hear and learn from.

5.3 Implications

Our study dealt with investigating the performance of Dutch language models and comparing multilingual and monolingual performance in this domain. We chose this domain for several reasons. Firstly, Dutch is a relatively high-resource language with several good-quality corpora and other linguistic resources available, allowing us to experiment with different sizes and types of fine-tuning data. Additionally, the Dutch language has some interesting linguistic phenomena which we wanted to explore. Among these phenomena are compounding which we explored in further detail. Despite these specific choices, our research has further reaching implications. Insights from our work can help develop the field of natural language processing by motivating the creation of language-specific language models which use higher-quality embeddings. This is not only applicable to higher resource languages like Dutch and English, but even to under-resourced languages with few small-scale corpora on which to train.

While large-scale multilingual language models use single tokenizers, we hypothesize that developing tokenizers for individual languages or adapting them to target languages will improve the quality of word embeddings used by LMs. This is not a tall task for languages with many resources, such as Dutch. However, lower-resource languages still struggle due to a lack of data on which to train tokenizers. In this case, dictionary-based approaches like BPE-knockout may help, though we are not aware of any applications of this method to (tuned) multilingual models so far.

Our work pertains to stacked encoder LLMs, but decoder LLMs such as GPT-4o (OpenAI 2024) are gaining prominence and these are often adapted from English or multilingual models by necessity. Little work has been done on intrinsic evaluation of these models against semantic similarity benchmarks, though it has been found that OpenAI's text-embeddings-3-large can achieve a 0.41 correlation with Dutch SimLex-999 (Snelder et al. 2025), which is similar to BERTje and our tuned multilingual models. For English, it has also been shown that prompt-based decoder LLMs can achieve far higher correlations (0.86) with human word similarity ratings when prompted to perform a rating task (Trott 2024), although this task is a bit different than the typical intrinsic evaluation setup as the model receives more context. It would be interesting to investigate whether tuning a multilingual model or pre-training a monolingual model yields better word embeddings also for generative decoder LLMs, although for Dutch we are not aware of any large monolingual models of this type to compare to. Also, this type of evaluation can only be performed on open-source models. Therefore, it is difficult to speculate how our findings would generalize to this class of models. Further research is needed to evaluate the quality of lexical-semantic representations of such models intrinsically.

6. Conclusion

In this work, we evaluated the performance of monolingual and multilingual language models for Dutch using a benchmark of human semantic similarity ratings for intrinsic evaluation. Furthermore, we propose and test methods by which we can improve the language-specific semantic representations of multilingual language models through tuning. By tuning models on target language data using the masked language modelling task, we were able to improve intrinsic evaluation scores compared to pre-trained mBERT, XLM-RoBERTa, mDeBERTa and EuroBERT models. We found that models fine-tuned with the most additional Dutch data showed the greatest improvements. Notably, XLM-RoBERTa-large and mDeBERTa fine-tuned on 100M tokens from the Dutch SoNaR corpus outscored the monolingual BERTje model on the benchmark. Interestingly, the recent EuroBERT-2.1B model did not score well, but shows potential for larger amounts of fine-tuning data.

We also compared tuning the multilingual models on transcribed spoken and written textual data. Here, we found model-specific effects - the older mBERT and XLM-RoBERTa-large did better on spoken data, while the newer mDeBERTa and EuroBERT did better with written data.

We noted significant differences in tokenization between the models and potential issues in applying multilingual tokenizers for Dutch, which may cause a performance bottleneck when tuning multilingual models, as the tuning process does not adapt the tokenizer to the language. Other work suggests that incorporating more morphological tokenization may help to improve performance. Existing solutions could be evaluated for adapting multilingual tokenizers to Dutch in future work.

Overall, we found that monolingual models BERTje and RobBERT outperformed multilingual models in terms of intrinsic evaluation, and that a significant amount of computational resources is required to fine-tune multilingual stacked encoder models to the same level, even ones that have more modern architectures. Nevertheless, we did observe our highest benchmark score from a XLM-RoBERTa-large model tuned on 100M tokens of Dutch written text, outperforming BERTje.

We suggest several directions for future work to build upon our findings. Firstly, pretraining monolingual language models using both written and spoken text data could provide further insights into their performance across intrinsic and extrinsic tasks, including various downstream NLP applications. Because this is computationally costly, we could not conduct the ablation study to its fullest extent. We would like to research further the effect of tuning data quantity on intrinsic evaluation scores to see if the trend continues. We were also not able to explore hyperparameter tuning for the model tuning process as this would be computationally costly, but in future work, this would be a way to get more performance from a limited amount of language-specific tuning data.

Finally, further exploration of the limitations of multilingual language model tokenization and its interaction with language-specific morphology, such as negation, compounding, and prefix/suffix usage, could reveal critical factors affecting model performance and guide the development of more effective multilingual models with language-specific tokenizer adaptations. Our compound word experiment suggests that it would be interesting to evaluate morphological tokenization for Dutch large language models in future work.

References

Arora, Simran, Avner May, Jian Zhang, and Christopher Ré (2020), Contextual embeddings: When are they worth it?, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 2650–2663.

Artetxe, Mikel, Vedanuj Goswami, Shruti Bhosale, Angela Fan, and Luke Zettlemoyer (2023), Revisiting machine translation for cross-lingual classification, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 6489–6499.

Ataman, Duygu, Matteo Negri, Marco Turchi, and Marcello Federico (2017), Linguistically motivated vocabulary reduction for neural machine translation from Turkish to English., *The Prague Bulletin of Mathematical Linguistics* **108** (1), pp. 331–342.

Bahdanau, Dzmitry, Kyung Hyun Cho, and Yoshua Bengio (2015), Neural machine translation by jointly learning to align and translate, *3rd International Conference on Learning Representations, ICLR 2015.*

Bakarov, Amir (2018), A survey of word embeddings evaluation methods, *arXiv preprint arXiv:1801.09536.*

Bauwens, Thomas and Pieter Delobelle (2024), BPE-knockout: Pruning pre-existing BPE tokenisers with backwards-compatible morphological semi-supervision, *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 5810–5832.

Bay, Serkan and Jelke Bloem (2023), Speech versus script: a language model analysis, The 33rd Meeting of Computational Linguistics in the Netherlands (CLIN 33). https://clin33.uantwerpen.be/abstract/speech-versus-script-a-language-model-analysis/.

Boizard, Nicolas, Hippolyte Gisserot-Boukhlef, Duarte M Alves, André Martins, Ayoub Hammal, Caio Corro, Céline Hudelot, Emmanuel Malherbe, Etienne Malaboeuf, Fanny Jourdan, et al. (2025), EuroBERT: Scaling multilingual encoders for European languages, *arXiv preprint arXiv:2503.05500.*

Bommasani, Rishi, Kelly Davis, and Claire Cardie (2020), Interpreting pretrained contextualized representations via reductions to static embeddings, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4758–4781.

Brans, Lizzy and Jelke Bloem (2024), Simlex-999 for Dutch, *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pp. 14832–14845.

Bruni, Elia, Gemma Boleda, Marco Baroni, and Nam-Khanh Tran (2012), Distributional semantics in technicolor, *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 136–145.

Cheong, Sik Feng, Hai Leong Chieu, and Jing Lim (2021), Intrinsic evaluation of language models for code-switching, *in* Xu, Wei, Alan Ritter, Tim Baldwin, and Afshin Rahimi, editors, *Proceedings of the Seventh Workshop on Noisy User-generated Text (W-NUT 2021)*, Association for Computational Linguistics, Online, pp. 81–86. https://aclanthology.org/2021.wnut-1.10.

Conneau, Alexis, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Édouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov (2020), Unsupervised cross-lingual representation learning at scale, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 8440–8451.

de Vargas Feijo, Diego and Viviane Pereira Moreira (2020), Mono vs multilingual transformer-based models: a comparison across several language tasks.

De Vries, Wietse, Andreas van Cranenburgh, Arianna Bisazza, Tommaso Caselli, Gertjan van Noord, and Malvina Nissim (2019), Bertje: A Dutch BERT model, *arXiv preprint arXiv:1912.09582.*

de Vries, Wietse, Martijn Wieling, and Malvina Nissim (2023), DUMB: A benchmark for smart evaluation of Dutch models, *in* Bouamor, Houda, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, Singapore, pp. 7221–7241. https://aclanthology.org/2023.emnlp-main.447.

Delobelle, Pieter and François Remy (2024), RobBERT-2023: Keeping Dutch language models up-to-date at a lower cost thanks to model conversion, *Computational Linguistics in the Netherlands Journal* **13**, pp. 193–203. https://www.clinjournal.org/clinj/article/view/180.

Delobelle, Pieter, Thomas Winters, and Bettina Berendt (2020), Robbert: a dutch roberta-based language model.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2019), BERT: Pre-training of deep bidirectional transformers for language understanding, *in* Burstein, Jill, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Association for Computational Linguistics, Minneapolis, Minnesota, pp. 4171–4186. https://aclanthology.org/N19-1423.

Drieghe, Denis and Marc Brysbaert (2002), Strategic effects in associative priming with words, homophones, and pseudohomophones., *Journal of Experimental Psychology: Learning, Memory, and Cognition* **28** (5), pp. 951, American Psychological Association.

Finkelstein, Lev, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin (2001), Placing search in context: The concept revisited, *Proceedings of the 10th international conference on World Wide Web*, pp. 406–414.

He, Pengcheng, Jianfeng Gao, and Weizhu Chen (2023), DeBERTav3: Improving deBERTa using ELECTRA-style pre-training with gradient-disentangled embedding sharing, *The Eleventh International Conference on Learning Representations.* https://openreview.net/forum?id=sE7-XhLxHA.

Hill, Felix, Roi Reichart, and Anna Korhonen (2015), Simlex-999: Evaluating semantic models with (genuine) similarity estimation, *Computational Linguistics* **41** (4), pp. 665–695, MIT Press One Rogers Street, Cambridge, MA 02142-1209, USA journals-info . . . .

Hou, Jue, Anisia Katinskaia, Anh-Duc Vu, and Roman Yangarber (2023), Effects of sub-word segmentation on performance of transformer language models, *The 2023 Conference on Empirical Methods in Natural Language Processing.*

Hu, Michael Y., Aaron Mueller, Candace Ross, Adina Williams, Tal Linzen, Chengxu Zhuang, Ryan Cotterell, Leshem Choshen, Alex Warstadt, and Ethan Gotlieb Wilcox (2024), Findings of the second BabyLM challenge: Sample-efficient pretraining on developmentally plausible corpora, *in* Hu, Michael Y., Aaron Mueller, Candace Ross, Adina Williams, Tal Linzen, Chengxu Zhuang, Leshem Choshen, Ryan Cotterell, Alex Warstadt, and Ethan Gotlieb Wilcox, editors, *The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning*, Association for Computational Linguistics, Miami, FL, USA, pp. 1–21. https://aclanthology.org/2024.conll-babylm.1/.

Jansen, Frank (2005), Subject–object ambiguities in spoken and written Dutch, *Linguistics in the Netherlands* **22** (1), pp. 99–109, John Benjamins.

Jiang, Albert Q., Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed (2023), Mistral 7b. https://arxiv.org/abs/2310.06825.

Kaya, Yiğit Bekir and A Cüneyd Tantuğ (2024), Effect of tokenization granularity for Turkish large language models, *Intelligent Systems with Applications* **21**, pp. 200335, Elsevier.

Keune, Karen, Mirjam Ernestus, Roeland van Hout, and R. Harald Baayen (2005), Variation in Dutch: From written MOGELIJK to spoken MOK, *Corpus Linguistics and Linguistic Theory* **1** (2), pp. 183–223. https://doi.org/10.1515/cllt.2005.1.2.183.

Kudo, Taku and John Richardson (2018), SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 66–71.

Litake, Onkar, Maithili Sabane, Parth Patil, Aparna Ranade, and Raviraj Joshi (2023), *Mono Versus Multilingual BERT: A Case Study in Hindi and Marathi Named Entity Recognition*, Springer Nature Singapore, p. 607–618. http://dx.doi.org/10.1007/978-981-19-6088-8_56.

Liu, Yinhan, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov (2019), RoBERTa: A robustly optimized BERT pretraining approach.

Mager, Manuel, Arturo Oncevay, Elisabeth Maier, Katharina Kann, and Thang Vu (2022), Bpe vs. morphological segmentation: A case study on machine translation of four polysynthetic languages, *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 961–971.

Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean (2013a), Distributed representations of words and phrases and their compositionality, *in* Burges, C.J., L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, Vol. 26, Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2013/file/9aa42b31882ec039965f3c4923ce901b-Paper.pdf.

Mikolov, Tomas, Kai Chen, Greg Corrado, and Jeffrey Dean (2013b), Efficient estimation of word representations in vector space, *arXiv preprint arXiv:1301.3781.*

Mykowiecka, Agnieszka, Malgorzata Marciniak, and Piotr Rychlik (2018), Simlex-999 for Polish, *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018).*

Oostdijk, N (2000), Het Corpus Gesproken Nederlands, *Nederlandse Taalkunde* **5** (3), pp. 280–284.

Oostdijk, Nelleke, Martin Reynaert, Véronique Hoste, and Ineke Schuurman (2013), The construction of a 500-million-word reference corpus of contemporary written Dutch, *Essential speech and language technology for Dutch: Results by the STEVIN programme* pp. 219–247, Springer Berlin Heidelberg.

OpenAI (2024), Hello GPT-4o — OpenAI. Retrieved June 20, 2024 from `https://openai.com/index/hello-gpt-4o/`.

Ortiz Su'arez, Pedro Javier, Laurent Romary, and Benoit Sagot (2020), A monolingual approach to contextualized word embeddings for mid-resource languages, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, Online, pp. 1703–1714. https://www.aclweb.org/anthology/2020.acl-main.156.

Peters, Matthew E., Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer (2018), Deep contextualized word representations, *in* Walker, Marilyn, Heng Ji, and Amanda Stent, editors, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, Association for Computational Linguistics, New Orleans, Louisiana, pp. 2227–2237. https://aclanthology.org/N18-1202/.

Pranav, A, Yan Cong, Emmanuele Chersoni, Yu-Yin Hsu, and Alessandro Lenci (2024), Comparing static and contextual distributional semantic models on intrinsic tasks: An evaluation on Mandarin Chinese datasets, *in* Calzolari, Nicoletta, Min-Yen Kan, Veronique Hoste, Alessandro Lenci, Sakriani Sakti, and Nianwen Xue, editors, *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, ELRA and ICCL, Torino, Italia, pp. 3610–3627. https://aclanthology.org/2024.lrec-main.320.

Remy, François, Pieter Delobelle, Bettina Berendt, Kris Demuynck, and Thomas Demeester (2023), Tik-to-tok: Translating language models one token at a time: An embedding initialization strategy for efficient language adaptation, *arXiv preprint arXiv:2310.03477.*

Rijgersberg, Edwin and Bob Lucassen (2023), Geitje: een groot open Nederlands taalmodel. https://github.com/Rijgersberg/GEITje.

Rust, Phillip, Jonas Pfeiffer, Ivan Vulić, Sebastian Ruder, and Iryna Gurevych (2021), How good is your tokenizer? On the monolingual performance of multilingual language models, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 3118–3135.

Schuster, Mike and Kaisuke Nakajima (2012), Japanese and Korean voice search, *2012 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, IEEE, pp. 5149–5152.

Sennrich, Rico, Barry Haddow, and Alexandra Birch (2016), Neural machine translation of rare words with subword units, *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Association for Computational Linguistics, p. 1715.

Snelder, Xander, Yunchong Huang, and Jelke Bloem (2025), Prompting instruction-tuned llms for semantic similarity values, Unpublished manuscript.

Trott, Sean (2024), Can large language models help augment english psycholinguistic datasets?, *Behavior Research Methods* pp. 1–19, Springer.

Vanroy, Bram (2024), Fietje: An open, efficient LLM for Dutch, *arXiv preprint arXiv:2412.15450.*

Venekoski, Viljami and Jouko Vankka (2017), Finnish resources for evaluating language model semantics, *Proceedings of the 21st Nordic conference on computational linguistics*, pp. 231–236.

Virtanen, Antti, Jenna Kanerva, Rami Ilo, Jouni Luoma, Juhani Luotolahti, Tapio Salakoski, Filip Ginter, and Sampo Pyysalo (2019), Multilingual is not enough: BERT for Finnish, *arXiv preprint arXiv:1912.07076.*

Vulić, Ivan, Simon Baker, Edoardo Maria Ponti, Ulla Petti, Ira Leviant, Kelly Wing, Olga Majewska, Eden Bar, Matt Malone, Thierry Poibeau, et al. (2020), Multi-SimLex: A large-scale evaluation of multilingual and crosslingual lexical semantic similarity, *Computational Linguistics* **46** (4), pp. 847–897, MIT Press One.

Warner, Benjamin, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, Nathan Cooper, Griffin Adams, Jeremy Howard, and Iacopo Poli (2024), Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference. https://arxiv.org/abs/2412.13663.

Weijers, Erik (2004), *Een verkenning van COREX. Introductie van het Exploitatieprogramma bij het Corpus Gesproken Nederlands.* www.mpi.nl/corpus/manuals/tutorial-corex.pdf¿.

Wolf, Thomas, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush (2020), Huggingface's transformers: State-of-the-art natural language processing.

Wu, Shijie and Mark Dredze (2020), Are all languages created equal in multilingual BERT?, *Proceedings of the 5th Workshop on Representation Learning for NLP*, pp. 120–130.

Zhou, Yichu and Vivek Srikumar (2022), A closer look at how fine-tuning changes BERT, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1046–1061.

Appendix A. Code and notebooks

All code and raw results used in this study can be found at:
`https://github.com/bloemj/dutch_mono_multi_bert`.

Appendix B. Full layer-wise results of pretrained models

These figures show layer-wise correlation scores with Dutch SimLex-999 for all pretrained models in the study.

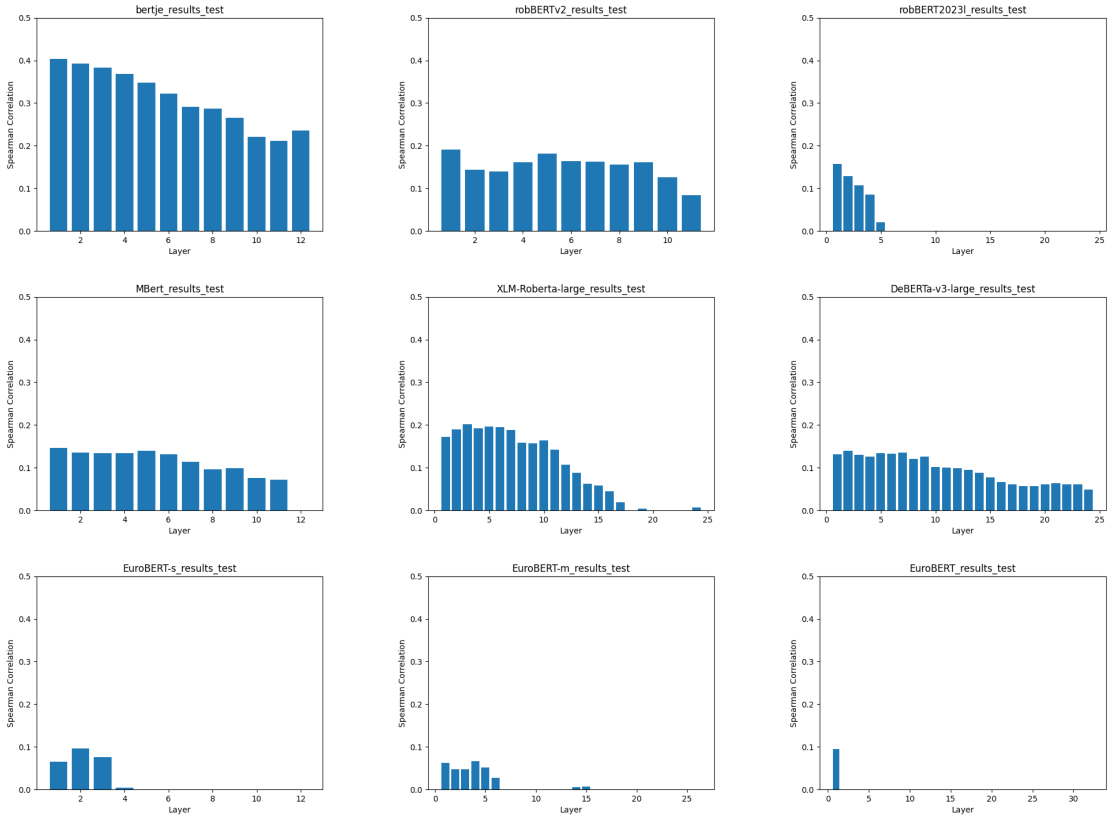

Figure 7: Aggregate layer-wise intrinsic evaluation scores for all pretrained models, excluding mDeBERTa which is shown in Figure 1

Appendix C. Layer-wise results of tuned models

These figures show layer-wise correlation scores with Dutch SimLex-999 for the models tuned on the largest (100M tokens) SoNaR written text dataset.

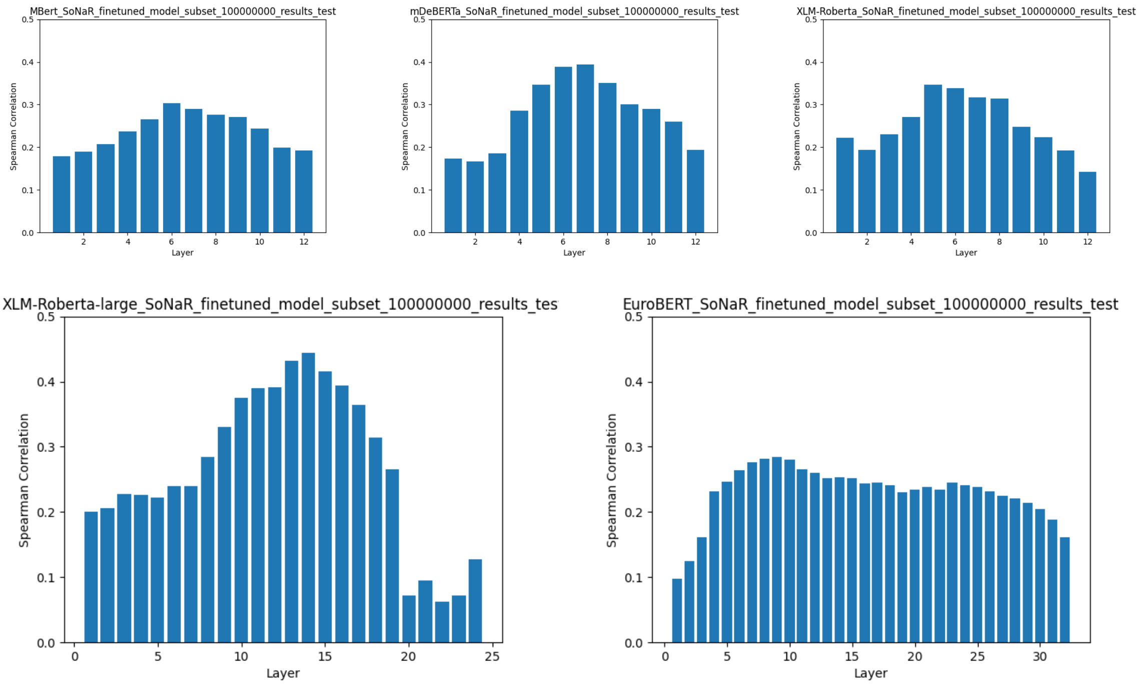

Figure 8: Aggregate layer-wise intrinsic evaluation scores for all models tuned on the 100M token SoNaR corpus sample

Appendix D. Manual Inspection of Tokenisers

Model	**Input**	**Tokenization**	Tokenizer
mBERT	werknemer	[‘werk’, ‘##nemer’]	WordPiece
BERTje	werknemer	[‘werknemer’]	WordPiece
RobBERT	werknemer	[‘werk’, ‘nemer’]	BPE
RobBERT-2023	werknemer	[‘werknemer’]	Tik-to-Tok
XLM-RoBERTa	werknemer	[‘werk’, ‘nemer’]	SentencePiece
mDeBERTa	werknemer	[‘werk’, ‘nemer’]	SentencePiece
DeBERTa-v3-large	werknemer	[‘we’, ‘rk’, ‘ne’, ‘mer’]	SentencePiece
EuroBERT	werknemer	[‘wer’, ‘kn’, ‘emer’]	BPE (Llama 3)
RoBERTa-BPE-39k	werknemer	[’werk’, ’nem’, ’er’]	BPE-knockout
RoBERTa-BPEko-39k	werknemer	[‘werknemer’]	BPE
RoBERTa-BPE-30k-BPEko-9k	werknemer	[’werk’, ’nem’, ’er’]	BPE+knockout
mBERT	aanwezigheid	[‘aanwezig’, ‘##heid’]	WordPiece
BERTje	aanwezigheid	[‘aanwezigheid’]	WordPiece
RobBERT	aanwezigheid	[‘aan’, ‘we’, ‘zigheid’]	BPE
RobBERT-2023	aanwezigheid	[‘aanwezigheid’]	Tik-to-Tok
XLM-RoBERTa	aanwezigheid	[‘aanwezig’, ‘heid’]	SentencePiece
mDeBERTa	aanwezigheid	[‘aan’, ‘wezi’, ‘gheid’]	SentencePiece
DeBERTa-v3-large	aanwezigheid	[‘a’, ‘an’, ‘we’, ‘zig’, ‘heid’]	SentencePiece
EuroBERT	aanwezigheid	[‘aan’, ‘we’, ‘zig’, ‘heid’]	BPE (Llama 3)
RoBERTa-BPE-39k	aanwezigheid	[’aanwezig’, ’heid’]	BPE-knockout
RoBERTa-BPEko-39k	aanwezigheid	[‘aanwezigheid’]	BPE
RoBERTa-BPE-30k-BPEko-9k	aanwezigheid	[’aanwezig’, ’heid’]	BPE+knockout
mBERT	schuim	[‘sc’, ‘##hu’, ‘##im’]	WordPiece
BERTje	schuim	[‘schuim’]	WordPiece
RobBERT	schuim	[‘schuim’]	BPE
RobBERT-2023	schuim	[‘schuim’]	Tik-to-Tok
XLM-RoBERTa	schuim	[‘sch’, ‘u’, ‘im’]	SentencePiece
mDeBERTa	schuim	[‘schui’, ‘m’]	SentencePiece
DeBERTa-v3-large	schuim	[‘sch’, ‘u’, ‘im’]	SentencePiece
EuroBERT	schuim	[‘sch’, ‘u’, ‘im’]	BPE (Llama 3)
RoBERTa-BPE-39k	schuim	[‘schuim’]	BPE-knockout
RoBERTa-BPEko-39k	schuim	[‘schuim’]	BPE
RoBERTa-BPE-30k-BPEko-9k	schuim	[‘schuim’]	BPE+knockout

Table 10: Manual inspection of tokenization of Dutch input with different models and tokenizers

Appendix E. Sampled CGN subcorpora

Text type	Description
tta	Spontaneous conversations (face-to-face)
ttb	Interviews with teachers of Dutch
ttc	Spontaneous telephone dialogues (recorded via a switchboard)
ttd	Spontaneous telephone dialogues (recorded on MD with local interface)
tte	Simulated business negotiations
ttf	Interviews/discussions/debates (broadcast)
ttg	(political) discussions/debates/meetings (non-broadcast)
tth	Lessons recorded in a classroom
tti	Live (e.g. sport) commentaries (broadcast)

Table 11: Text types sampled from CGN (Weijers 2004, p. 37), representing relatively spontaneous speech

Text type	Description
ttj	Newsreports/reportages (broadcast)
ttk	News (broadcast)
ttl	Commentaries/columns/reviews (broadcast)
ttm	Ceremonious speeches/sermons
ttn	Lectures/seminars
tto	Read speech

Table 12: Text types **not** sampled from CGN (Weijers 2004, p. 37), representing prepared speech

Appendix F. Transcript specific codes

Symbol	Description
*v	foreign (= non-Dutch) word
*d	dialect
*a	incomplete word
*u	slip of the tongue or onomatopoeia
*z	word with dialectal pronunciation
*x	word difficult to hear
ggg	a non-speech sound produced by the speaker
xxx	one or more incomprehensible words or partial words
Xxx	an incomprehensible word that is clearly a title or proper name
”.”	the full stop marks the end of a sentence
”...”	the ellipsis sign marks the end of an incomplete sentence
”?”	the question mark indicates the end of an interrogative sentence

Table 13: CGN codes removed from training data and their definitions

Appendix G. Hardware specifications

GPU: 8x NVidia RTX6000 Ada
GPU Memory: 48GB

CPU: 2x AMD 9554
Memory: 768 GB