Computational Linguistics in the Netherlands Journal 11 (2021) 141-159 Submitted 10/2021; Published 12/2021

MedRoBERTa.nl: A Language Model for Dutch Electronic Health Records

Stella Verkijk[*] STELLAVERKIJK@OUTLOOK.COM
Piek Vossen[*] P.T.J.M.VOSSEN@VU.NL

[*] *Vrije Universiteit Amsterdam, The Netherlands*

Abstract

This paper presents MedRoBERTa.nl as the first Transformer-based language model for Dutch medical language. We show that using 13GB of text data from Dutch hospital notes, pre-training from scratch results in a better domain-specific language model than further pre-training RobBERT. When extending pre-training on RobBERT, we use a domain-specific vocabulary and re-train the embedding look-up layer. We show that MedRoBERTa.nl, the model that was trained from scratch, outperforms general language models for Dutch on a medical odd-one-out similarity task. MedRoBERTa.nl already reaches higher performance than general language models for Dutch on this task after only 10k pre-training steps. When fine-tuned, MedRobERTa.nl outperforms general language models for Dutch in a task classifying sentences from Dutch hospital notes that contain information about patients' mobility levels.

1. Introduction

As the amount of language data available online grows, so does the interest from major tech companies in Natural Language Processing techniques. Consequently, computationally costly innovations emerge. The field of Natural Language Processing (NLP) has seen a major uplift since the rise of Deep Learning and especially the introduction of Transformer encoding (Vaswani et al. 2017), Google's language model BERT (Devlin et al. 2019) and Facebook's language model RoBERTa (Liu et al. 2019). BERT and RoBERTa are neural networks trained to model language by predicting masked words in sentences. These models learnt general patterns of language during pre-training and can be fine-tuned on any other more specific NLP task. BERT and RoBERTa are pre-trained on English Wikipedia texts (2,500M words) and the BookCorpus (Zhu et al. 2015) (800M words). However, not all language data is the same, and the language data openly available online does not represent all registers and genres (domains) a language may have. In this section, we will show how this led to the creation of many domain-specific models and why such a model made specifically for the Dutch medical language domain could be extremely useful as well.

1.1 The rise of domain-specific language models

Since the release of BERT and RoBERTa, many adaptations of these models have been presented. Language models that are specialized on a certain domain appear to outperform BERT and RoBERTa on tasks within that domain (Beltagy et al. 2019, Huang et al. 2019, Chalkidis et al. 2020, Lee et al. 2020, Gu et al. 2020, Müller et al. 2020). These domain-specific models all had data from a specific domain of the English language included in some way during pre-training. Examples of such domains are the scientific research domain (Beltagy et al. 2019), the legal domain (Chalkidis et al. 2020) and the biomedical domain (Gu et al. 2020, Lee et al. 2020).

Dutch language models such as BERTje and RoBBERT are trained on Wikipedia, news and web data. Up until this day, there is no specific language model for Dutch medical language, even though it could be a useful asset for various NLP tasks in the medical domain, for example when extracting valuable information from unstructured data in Electronic Health Records (EHRs) or for the creation of healthcare tools. Especially when the information is diverse and complex and the language is varied, we expect that a specialised medical language model can contribute enormously to develop fine-tuned information extraction software from large volumes of unstructured medical data. In this paper, we present MedRoBERTa.nl, the first language model trained on Dutch hospital notes, sourced from EHRs. An EHR contains the notes from care takers in hospitals about an individual patient.

Currently, MedRoBERTa.nl is used to find recovery patterns for COVID-19 patients by detecting functional levels of patients in various types of medical notes[1]. This task entails the labelling of sentences or complete notes with domains from the World Health Organization's International Classification of Functioning, Disability and Health (WHO ICF)[2]. For example, a sentence like 'Pt mobiliseert niet' (*Patient does not mobilzie*) would be labelled with the domain ICF code *d450* for *Walking* and with the level label 0, indicating that the patient has no walking ability (a level of 5 would mean the patient has complete walking ability).

1.2 Dutch hospital notes as a domain

Hospital notes are short texts that vary in their precise subject and form, but that are always about the status of a patient. The purpose of the notes is to communicate this status between different health-care professionals. As such they contain a mixture of general and specialised language. An example of such information would be a nurse writing down what a patient has eaten on a specific day, whether they were able to go to the bathroom themselves or needed help, whether they have taken their medication, etc. Other examples are a recollection of some type of communication between two doctors or a note on a patient's own observations about their well-being. Generally, a hospital note is an enumeration of different aspects of a patient's well being.

The language used in hospital notes differs considerably from standard Dutch. An important difference between general Dutch and the language used in Dutch hospital notes is the common usage of medical terms. Names of illnesses, medicines, treatments, but also symptoms are much more common than they are in Wikipedia texts, books or news articles. Apart from this difference, there are more incongruities. For example, the sentences in hospital notes tend to be much shorter than sentences in news or Wikipedia articles. This is one of the traits resulting from the tendency to use very efficient language in hospital notes, where functional words are left out, sentences are simplified and no special attention is given to grammaticality. This last aspect is exemplified in sentences that are common in hospital notes, but that are not regular in general Dutch, like in Example (1). This sentence is strictly speaking ungrammatical, since the transitive use of the verb 'imponeren' (*to impress*) in this sentence does not convey the meaning that it has in general Dutch. As for Example (2): 'possibel' is not an actual word in Dutch and a native Dutch speaker would use the word 'mogelijke' in this context. These examples show that the language in hospital notes sometimes adheres to a different syntax, vocabulary and meaning for common words. This makes this domain different from the biomedical domain, where inputs are sourced from carefully edited PubMed articles. It is important to note that the language used in hospital notes is also different from other domains where efficiency in writing and the absence of editing plays a role, such as social media posts. In hospital notes, constructions like those in examples 1 and 2 are grammaticalized and become part of the medical 'lingo'. Although social media posts also adhere to their own register, including grammatical and spelling mistakes, the grammaticalizations specific to the medical domain are not present in social media posts. As the language in hospital notes is such a unique domain, it raises the question of whether anything can be gained by fine-tuning an existing model or whether it is necessary to build a new language model, specific to the Dutch medical domain.

(1) De stemming **imponeert** normofoor, met een normaal modulerend affect
(*Mood **impresses** normophore, with normal modulating affect)*

[1] Effectiveness of allied healthcare in patients recovering from COVID-19, ZonMw project 10390062010001

[2] https://www.who.int/standards/classifications/international-classification-of-functioning-disability-and-health

.

(2) Patiënt met **possibele** pulmonale aspergillus met oplopend galactomannan onder vori mono
(*Patient with **possible** pulmonary aspergillus with ascending galactomannan under vori mono*

1.3 Research Questions and Contributions

This paper aims to answer several research questions. The main objective is to show to what extent a domain-specific language model for Dutch hospital notes is better at modeling the language specific to those notes than general language models for Dutch, and to see whether such a model performs better when fine-tuned on tasks specific to the medical domain. Attention will also be paid to the question whether the amount of data available for this study is sufficient to pre-train a new language model, and whether it is possible to build a successful model more economically than RobBERT and BERTje. We also investigate whether it is better to pre-train a model from scratch or to extend pre-training on an existing model for general Dutch (in this case RobBERT (Delobelle et al. 2020)). We release the best performing model, namely the one trained from scratch, as MedRoberta.nl.

For the creation of the extended model, we apply a new method which includes a domain-specific vocabulary, proposed by de Vries & Nissim (2020). In order to maintain the information stored in RobBERT's transformer layers when extending pre-training with a new vocabulary, we commence by freezing the transformer layers and only re-train the embedding look-up layer, after which we pre-train the complete model further.

The contribution of this research is fourfold. Firstly, MedRoBERTa.nl is released to the public so that it can be used in future research in the Medical NLP domain and for creating healthcare tools in hospital contexts. Secondly, we demonstrate how the model can be applied to assign functional categories in the World Health Organisation's International Classification of Functioning, Disability and Health (WHO ICF). Thirdly, we compare building a model from scratch with extending pre-training on an existing model for general Dutch, demonstrating that a model from scratch provides a language model with more accurate sentence embeddings for the domain. Finally, we demonstrate that such a dedicated model can be constructed with less extreme computational power than was used for the generic models.

The remainder of this paper will consist of a discussion of relevant literature and previous work in Section 2, followed by a description and discussion of the methods and data used for pre-training in Section 3 and for evaluating the medical language models in Section 4. After this, the results will be presented in Section 5, along with our analysis and discussion. The paper will end with conclusions in Section 6.

2. Related Work

Several studies demonstrate that domain-specific models give better results at tasks within their domain than general models. The models showing these results differ in their pre-training techniques and their evaluation methods.

There are two ways of pre-training a language model on new data: pre-training from scratch and extending pre-training on an existing model. The first implies that the model is initialized with random weights, whereas the second means that trained weights from an existing model are taken as a starting point as a means of transfer learning. Although the prevailing assumption is that it is preferable to start with trained weights (Gu et al. 2020), there are advantages and disadvantages to both these techniques. It seems to vary per domain and even per target task which method is preferable.

Chalkidis et al. (2020) experimented with training from scratch as well as with extending pre-training on BERT, and showed that it depends on the downstream task which model performs better. Both BioBERT (Lee et al. 2020) and ClinicalBERT (Huang et al. 2019) were made by extending pre-training on BERT, showing improved performance on downstream tasks in the biomedical domain.

However, (Gu et al. 2020) built a new domain-specific language model for the biomedical domain by pre-training from scratch, and show improved performance on most biomedical downstream tasks compared to both BioBERT and ClinicalBERT. They argue that transfer learning by extending pre-training on a general language model is only preferable when there is not much domain-specific data to pre-train on. They also state that a big advantage of training from scratch is the fact that a domain-specific vocabulary can be used during pre-training. The SciBERT model (Beltagy et al. 2019) was also trained from scratch and the authors experimented with using either the general vocabulary from BERT or their own domain-specific vocabulary during pre-training, concluding that using a domain-specific vocabulary indeed had a positive effect on the final performance of the model. Müller et al. (2020) had less data available for pre-training and extended pre-training on BERT to create their Covid-Twitter-BERT model. They showed that the more specific the downstream task, meaning the more *in-domain*, the less pre-training time their model needed to reach optimal performance.

As there are different ways of pre-training a language model, there are also different options with regard to the evaluation of the model. Since language models are mostly used to be fine-tuned on a specific task, this is also what they tend to be evaluated on. PubMedBERT (Gu et al. 2020), BioBERT (Lee et al. 2020) and Covid-Twitter-BERT (Müller et al. 2020) were evaluated after being fine-tuned on in-domain end tasks. The authors refrained from evaluating the quality of the models' embeddings, as a means of testing the raw model's value before it being fine-tuned. Beltagy et al. (2019) (SciBERT) did perform this extra step by feeding embeddings from their model to a BiLSTM. However, these types of probing tasks can sometimes also blur the outcome, as the classifier (in this case the BiLSTM) itself has an impact on the results (Belinkov and Glass 2019). Therefore, the optimal way to evaluate the model's intrinsic knowledge is to put the embeddings to the test without any other influencing factor. This is possible by performing a similarity test, as was done by Huang et al. (2019) for the evaluation of ClinicalBERT. They used two datasets of clinical concepts, provided with similarity scores by physicians, to see how well their model could approach human similarity judgement.

Thus, there is ample previous experience with building domain-specific language models, but there are still some open questions and unexplored opportunities. All biomedical and clinical domain-specific models discussed were based on the BERT model. This study presents the first clinical model based on the RoBERTa model (see Section 3.1). Also, there is no domain-specific model available yet for the Dutch clinical domain. Apart from providing the first model of this kind, this study aims to further investigate what the best methods for pre-training a domain-specific model are. We try a new pre-training method in which pre-training is extended on an existing model but a domain-specific vocabulary is included in the pre-training phase, to see if a domain-specific model for a domain where much data is available can still benefit from transfer learning. We also experiment with training time when pre-training from scratch to see if domain-specific models can be built more efficiently in the future. To ensure a thorough evaluation, this study aims to evaluate the model on in-domain as well as out of domain end tasks and to test the raw model before fine-tuning on a similarity test. By testing on a general, out of domain end task we expect to see a steep drop in performance from the medical models, thereby showing how different Dutch clinical language is from the general domain. To provide a fair and indicative evaluation, we created a domain-specific similarity test set (see Section 4.1).

3. Pre-training language models

Several decisions need to be made beforehand when pre-training a new language model, especially when there is not much room for experimenting due to limited computational power. It was decided to build models upon the RoBERTa architecture, to use a domain-specific vocabulary as opposed to a general language vocabulary, and to try two methods of pre-training: training from scratch and extending pre-training on the general Dutch language model RobBERT. Since it was decided to build both models following the RoBERTa architecture for continuity, RobBERT was taken as the Dutch general language model as opposed to BERTje to extend pre-training on. In this section, we describe the motivation behind our pre-training approaches.

3.1 BERT or RoBERTa?

It was decided to base the medical language models on RoBERTa as opposed to BERT because of i) a lack of necessity for the Next Sentence Prediction (NSP) objective, ii) the expectation that larger input sequences result in a better representation of long-term dependencies, iii) a preference of a byte-level BPE tokenizer over a character-level BPE tokenizer and iv) the expectation that a bigger batch size leads to better performance. These motivations will be further discussed next.

Training objectives

An important distinction between BERT and RoBERTa is the choice in training objectives. While BERT uses both Masked Language Modeling (MLM) and Next Sentence Prediction (NSP), RoBERTa only uses MLM. Research has shown that removing NSP does not hurt or even slightly improves the performance compared to BERT (Rogers et al. 2020). Lan et al. (2019) claim that the way NSP is implemented in BERT indeed teaches the model to predict which sentences are successive, but with the wrong motives. In BERT, for the sentence pairs fed to the model that are not successive (negative examples), the second sentence is a random sentence from the corpus (Devlin et al. 2019). This means that the sentence can be from a different document, and will probably be about a very different topic than the first sentence. Consequently, it does not learn about the 'relationship between sentences' as stated by Devlin et al. (2019), but rather learns topic modelling, which is also already partly covered in MLM (Lan et al. 2019). Apart from this, the input formatting that is needed for NSP, where two segments are separated and every token in the sentence is accompanied with a sentence index label, can be restraining. Joshi et al. (2020) show that pre-training on single segments and refraining from using NSP instead of pre-training on two half-length sentences with the inclusion of NSP, 'considerably improves performance on most downstream tasks' (p. 65). Mickus et al. (2019) find that BERT's word embeddings are influenced by the the NSP objective, as the introduction of systematic distinctions between first and second sentences impacts the output embeddings. In other words, the sentence indexing of the input influences the outcome of the model, which should not be the case.

Apart from previous research hinting that the inclusion of NSP is not necessary, a strong motive not to use NSP for this specific project is the way in which sentences are structured in the domain-specific data. Very often, one sentence in a hospital note has little connection to the sentence before and after. As mentioned in Section 1.2), a hospital note tends to be an enumeration of different aspects of a patient's well being. For example, one sentence might be about the walking capacities, the next about the patient's mood, and the next about their medication. For the same reasons, Sentence-Order-Prediction (SOP) is also not expected to be useful for hospital notes.

It can be concluded that NSP can lead to better performance in some down-stream tasks, such as Question Answering, that rely heavily on topic modelling, but that for a general-purpose language model trained on hospital notes, it is expected that taking out NSP will not hurt or even improve the language model.

Input formatting

The expectation is that the input format of RoBERTa, enabled by the deletion of NSP, captures the medical data in a better way than BERT's input format. The sentences in the data are sometimes extremely short ('Stemming: goed' (Mood: good)). Therefore, it can be expected that gathering as many sentences as possible in the input instead of dividing it into two separate segments is the best choice for capturing long-term dependencies that span multiple sentences.

Tokenization

For a language model, tokenization is in essence the process where it converts natural language into numbers by linking (parts of) words to token IDs. This makes it an important aspect of a language model's architecture. There are several methods in tokenization: word tokenization, character tokenization and subword tokenization. The principle of subword tokenization is that frequently used words should not be split into subwords and that rare words should be split into meaningful subwords. Although both BERT and RoBERTa opt for a hybrid approach between character and (sub)word tokenization using a Byte Pair Encoding (BPE) tokenizer, there are small differences between the two. Where BERT uses WordPiece, a character-level BPE tokenizer, RoBERTa uses a Byte-level BPE tokenizer. With the first, text is represented as a sequence of character n-grams, while with the second, text is tokenized into variable-length byte n-grams (Wang et al. 2020). The advantage of the byte-level tokenizer used by RoBERTa is that it can encode any input text and will therefore never encounter an unknown token (it can always be split up into bytes) (Liu et al. 2019, Wang et al. 2020). Also, BERT implements a vocabulary with 30k entries while RoBERTa's byte-level BPE vocabulary consists of 50k sub-word units.

Batch size

The RoBERTa model also distinguishes itself from BERT by using different hyperparameters for pre-training. The most significant of these is using a larger batch size. Through performing experiments with varying batch sizes, Liu et al. (2019) show that 'training with large batches improves perplexity for the masked language modeling objective, as well as end-task accuracy' (p. 6). Although the batch size you can use depends on the computational power available, and for this project the computational power to equal RoBERTa's batch size of 8k sequences was not available, it was expected that pre-training with a larger batch size than BERT would still be the better choice.

3.2 From scratch or extending?

As mentioned in Section 2, there are two ways of pre-training a language model on new data: pre-training from scratch and extending pre-training on an existing model. Since a model with trained weights has already learnt some general patterns of language it is generally thought that extending pre-training on an existing model is better than pre-training from scratch and having a random initialization. According to Müller et al. (2020), pre-training a domain-specific model from scratch requires a very large corpus to 'unlearn' random initialization, which is not available in all domains. The upside of having trained weights unfortunately comes with the downside of initializing a model that is already trained on a type of language that might not be like the domain-specific language. General Dutch adheres to certain rules for grammar and morphology that might be different for medical Dutch. Also, in general Dutch, medical terminology is not frequent compared to general terminology, of which the opposite is the case in Dutch hospital notes. Finally, common words may be used in very different meanings in hospital notes.

When extending pre-training on a general language model to make it domain-specific, there are two options: i) extending pre-training on this model with new training data but using the original vocabulary; ii) create a new vocabulary based on the new training data and extend pre-training with new data and a new vocabulary. In the first case, you risk that some or many frequent or important tokens in the training data are not represented in the vocabulary, while in the second case you lose the information on how the token IDs in the vocabulary are linked to their embeddings. Depending on the domain, the lexical variety in the domain-specific training data can differ so much from that of the original training data that the vocabulary of the original model becomes completely unrepresentative. The fact that every unknown domain-specific word that the model encounters will be split up into subwords makes the training much more challenging (Tai et al. 2020). Previous studies in the creation of domain specific models has shown that it is essential to the success of a domain-specific language model that it is trained with a vocabulary from the same domain (Beltagy et al. 2019, Gu et al. 2020). SciBERT (Beltagy et al. 2019) and PubMedBERT (Gu et al. 2020) were both trained from scratch with a new vocabulary. Both studies showed that the customized vocabulary resulted in higher performance in downstream tasks. Therefore, for the extended model trained in this study, it was decided to create a new domain-specific vocabulary for the training process.

Extending pre-training on an existing system is, in that sense, more complicated, because the off-the-shelf model does not recognise the new vocabulary. De Vries & Nissim (2020) present a new strategy to include a newly created domain-specific vocabulary in the process of extending pre-training on an existing model. We adopted this procedure, which included the following steps: we initialized the RobBERT model and the domain specific vocabulary and tokenizer (the same vocabulary and tokenizer as we use for pre-training from scratch). As in the standard RoBERTa architecture, the model has a total of 25 layers when taking the embedding lookup layer into account (the first layer). In the first pre-training phase, the whole model except for the embedding lookup layer was frozen and trained for 1 epoch. This means that only the embedding lookup layer was trained and therefore adapted, in order to align with the new domain-specific vocabulary. In the meanwhile, the transformer layers were frozen, meaning the information stored there remains intact. This ensures that the model will link the embedding stored in the transformer layers of a specific token to the same token in the new vocabulary, even though it has a different token id. However, this means no domain adaptation happens in the transformer layers. Therefore, as a second step, the model with the embedding look-up layer trained up until that point was initialized, none of the layers were frozen and the whole model was trained for one epoch. In Figure 1 this is represented schematically: the RobBERT model with the frozen transformer layers and the domain specific vocabulary are taken as input for the first epoch, after which this model is initialized and pre-trained further without freezing any layers. The model was trained for a total of 2 epochs following the strategy for the RobBERT model (Delobelle et al. 2020).

When training from scratch, you start with random weights. This means the model is not yet tuned to (Dutch) language. You cannot control a random weight initialization, which means that two pre-trained models initiated from scratch that had the exact same learning scheme and data might have different performances. However, if Dutch hospital notes are as drastically different from other Dutch text data as we think (see Section 1), it might be the case that a model that is already trained on general Dutch language has learned patterns that make it harder to understand hospital notes and that are hard to unlearn. This argues for a start with random weights instead of with trained weights from a general language model. Also, as mentioned in Section 2, there are examples of domain specific models that were pre-trained from scratch outperforming models trained with initialized weights (Chalkidis et al. 2020, Beltagy et al. 2019), especially when there is ample pre-training data available. In our case, there was 13GB of domain-specific language data available, which is slightly more than the complete amount of training data of BERTje (12.1GB). On top of that, including a domain-specific vocabulary is straight-forward when training from scratch. (see Figure 2).

To summarize, both training options introduced in this section have advantages and disadvantages. Comparing them will give some more insights in which method is more desirable. For a schematic overview of both methods, see Figures 1 and 2.

3.3 Hyperparameters

Both models were trained for a little over 2 epochs, which in our case meant 92.494 iterations (41.247 per epoch). Through gradient accumulation we reached a batch size of 400. Although this is much lower than RoBERTa's and RobBERT's batch size of 8k, it is considerably higher than BERT's batch size of 256. Both models started with a learning rate of 8e-4, lowered to 1e-5 after 10k steps. Adam optimization (Kingma and Ba 2014) was used with linear decay. Weight decay was set to 0.01 and the beta values were set to the default 0.9 and 0.98. Using mixed precision training, it took around

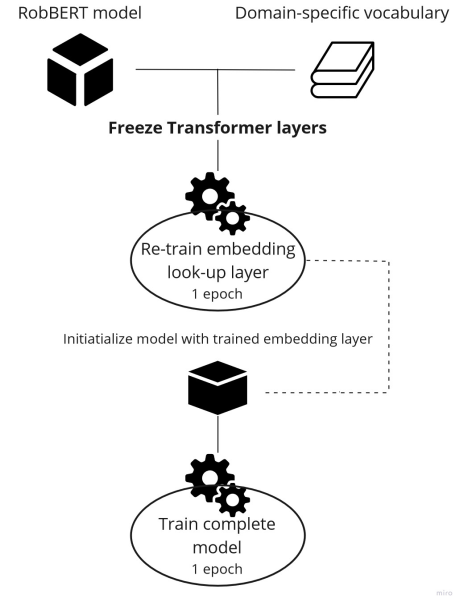
RobBERT model
Domain-specific vocabulary
Freeze Transformer layers
Re-train embedding look-up layer
1 epoch
Initiatialize model with trained embedding layer
Train complete model
1 epoch
miro

Figure 1: Training process of the extended model

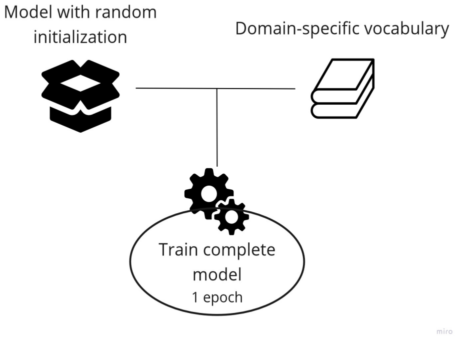

Figure 2: Training from scratch

12 days to train one model. The medical models were trained on four 16 GB Nvidia Tesla p100 PCIe GPUs on a highly secured server. To compare, RobBERT was built on a computing cluster with four of these GPUs per node, where the number of nodes could be dynamically adjusted. They could use up to 20 nodes, which means 80 GPUs (Delobelle et al. 2020). Roughly said, Delobelle et al. had access to about 20 times more computational power than we had access to.

3.4 Pre-training data

For the pre-training of a Dutch medical language model, data was sourced from hospital notes from the Medical Centre of the Vrije Universiteit (VuMC) and the Amsterdam Medical Centre (AMC). Hospital notes from 2017, 2018 and 2020 were used. For an overview of the volume and the sources of the pre-training data, see Table 1.

		2017	2018	2020
AMC	GB	2,8	3,0	2,0
	# notes	2.375.626	2.451.973	1.492.573
VuMC	GB	3,0	-	1,5
	# notes	2.545.515	-	1.111.116

Table 1: Amount of gigabytes and notes per year per hospital used for pre-training

4. Evaluating language models

As mentioned in Section 2, there are various ways to evaluate a language model. For example, there is a distinction between evaluating the raw model that has not been fine-tuned yet, and evaluating its performance on downstream tasks after fine-tuning. In this paper, we will refer to these methods as intrinsic and extrinsic evaluation, respectively. Another distinction that should be made when evaluating domain-specific models is in-domain and out-of-domain evaluation. To conduct a thorough evaluation of the language models created in this project, we conducted in-domain intrinsic, in-domain extrinsic and out-of-domain extrinsic evaluation. This section describes these approaches as well as the motivation behind them.

4.1 Intrinsic evaluation

Intrinsic evaluation means that the original model is evaluated by assessing the accuracy of the models' representations of text (the embeddings). To evaluate the medical models in this way, the model's similarity judgements of sentences will be compared to human similarity judgements.

The reason why the models are tested on similarity judgements is twofold. Firstly, previous literature shows that the way a system can model similarity is a good indicator of the model's overall performance. For example, Cer at al. (2017) state that "[a]ccurately modeling the meaning similarity of sentences is a foundational language understanding problem relevant to numerous applications". Also, Hill et al. (2015) point out that "similarity estimation constitutes an effective proxy evaluation for general-purpose representation-learning models whose ultimate goal is variable or unknown". Secondly, measuring sentence similarity implicates that the quality of the model's embeddings can be tested without the intervention of a classifier.

Because there are little to no data sets available in the Dutch medical domain to test language models on, let alone data sets created from private hospital notes, a similarity test set for intrinsic evaluation was created especially for this project. This test set was anonymized so it can be released to the public.

Testing on textual semantic similarity judgements has been a well-researched aspect of NLP. Important insights and results on this matter have been reported in papers of contestants of the SemEval 2017 Shared Task 1 (Cer et al. 2017). This shared task involved determining the similarity between two snippets of text. Systems were created to return a continuous valued similarity score on a scale from 0 to 5. In this scale, 0 indicates that the semantics of the sentences are completely independent and 5 indicates semantic equivalence. Performance for this task was assessed by computing the Pearson correlation between the similarity scores of the machine and human judgements. However, similarity judgements are highly complicated cognitive tasks (Hill et al. 2015) and it is not always a straightforward task to rate similarity in a scale. This was also shown in exploratory experiments that we performed for the creation of the similarity test set. In an attempt to gather more and less similar sentences, BERTje and RobBERT were used to rank pairs of phrases from hospital notes according to their similarity scores. These experiments showed that both BERTje and RobBERT assigned very high similarity scores to most pairs. The lowest similarity score assigned by RobBERT was around 0.74 (where 1 is perfect semantic similarity). For some examples on irrelevant sentences receiving high similarity scores by RobBERT, see Table 2. BERTje had scores on

Snippet 1	*Snippet 1 (Translated)*	Snippet 2	*Snippet 2 (Translated)*	Similarity score
Is aan het mobiliseren , loopt kleine stukjes met rollator	*Is mobilizing, walks small distances with a walker*	Vond het toen erg eng , wilt morgen graag met logopediste oefenen.	*Was scared at the time, would like to practice with a speech therapist tomorrow.*	0.94
Dhr raakte tijdens het vernevelen in paniek.	*Mr. panicked during the nebulization.*	Hij heeft een half uur in de stoel gezeten , ging goed maar is wel vermoeidend.	*He sat in the chair for half an hour, went well but is tiring.*	0.94
patiënt huilt	*patient is crying*	Heeft met Fysio trap gelopen.	*Walked some stairs with physiotherapist.*	0.87
Bij inspanning op bed daalt saturatie naar 86%	*With exercise in bed, saturation drops to 86%*	Gemotiveerd.	*Motivated.*	0.84
max	*max*	Gezien saturatie dip nu staan nog brug te ver , hierna wel snel herstel.	*Given the saturation dip standing up still a bridge too far, but quick recovery after dip.*	0.74

Table 2: Sample of some of the highest scoring snippet pairs (above the dashed line) and lowest scoring snippet pairs (underneath the dashed line) and their English translation for a preliminary experiment testing RobBERT on sentence similarity judgements in the SemEval 2017 Task 1 style.

a slightly broader range but performed in a similar manner. This meant that the general language models would hardly score anything lower than 4 or 5 on the SemEval 2017 scale. This lead to the conclusion that the SemEval scale was not suited for our experiment, since it raised the expectation that assessing the individual similarity scores would most probably not lead to a fair comparison between the general language models and the domain-specific medical language models.

Because of the complexity of a similarity ranking task, we created an odd-one-out similarity test set as an alternative. In an odd-one-out similarity judgement task, models are each time presented with three snippets of text (also referred to as sentences in this paper) and the task is to find the snippet that least fits the other two. Hence, the system has to indicate which of the three sentences has the lowest combined similarity score to the other two. In this way, the individual similarity scores assigned by the system do not need to be assessed, and each model has a fair chance of selecting the odd one out, even if the average similarity score of one system is far above the other. A second advantage of this simplified task is that it ensures more reliable labelling from human annotators. Rating similarity of sentence pairs on a 5-point is also difficult for people, as sentences may be complex and express mixtures of aspects to compare. Odd-one-out judgements are much easier because differences do not need to be translated to a scale and are limited to the three sentences and not to all other judgements to create a ranking as in pairwise scalar scores.

The goal of the data set curation was thus to gather sets of three sentences (henceforth referred to as triples) of which one out of three sentences is less similar to the other two then the other two are to each other. To do this, we made use of a dataset of hospital notes annotated in a pilot project by the VU University and the VU Medical Centre (the a-proof project[3]), where sentences were either labelled with one of four categories selected from the International Classification of Functioning, Disability and Health (ICF), or left with no label. The four selected ICF categories were d450: *Walking*; b152: *Emotional functions*; b455: *Exercise tolerance functions*, and d840-859: *Work and employment*. For more information on these categories, see the official WHO ICF online browser here: `https://icd.who.int/dev11/l-icf/en`. When a sentence was not about any of those categories, it was automatically ascribed to the *None* class. The sentences that were labelled with a domain also received a label for the so-called functional level of a patient. This meant that if a sentence was, for example, about the walking capabilities of a patient, the sentence would also receive a label on a scale of 0 to 5 indicating how well the patient could walk in that moment. We

[3] Ondersteunen van klinische beslissingen in de revalidatie van patiënten met Covid-19 m.b.v. datascience, funded by the UMC Covid fund

extracted all sentences labelled with an ICF category and matched them with sentences with the same label or with a different label to create combinations of similar and less similar sentences. It was also important to ensure that the level of difficulty to identify the odd one out differed per triple. Therefore, triples were selected so that various situations would hold, namely the following:

T1 All sentences are from the same ICF domain, but two of them contain overlapping keywords

T2 Two of the sentences are from the same ICF domain, one is from a different ICF domain and the sentences that are from the same domain do not contain overlapping keywords

T3 Two of the sentences are from the same ICF domain, one is from a different ICF domain and the sentences that are from the same domain also contain at least one overlapping keyword

T4 All sentences are from the same ICF domain, but the functional level of one sentence differs from the functional levels of the other two sentences. Sometimes the difference in functional level is small, other times bigger

A list of keywords was created per ICF domain. See Table 3 for an overview of the keywords used per domain. FAC is a widely used abbreviation (Functional Ambulation Categories) to indicate the level of mobility of patients by caretakers.

Domain	**Keywords**	**Translated keywords**
Walking	FAC0, FAC1, FAC2, FAC3, FAC4, FAC5, transfer, mobiliteit, tillift, rolstoel, stoel, bed, stapjes, stap, stappen	FAC0, FAC1, FAC2, FAC3, FAC4, FAC5, transfer, mobility, hoist, wheelchair, chair, bed, little steps, step, stepping
Emotional functions	modulerend affect, affect vlak, emotioneel, droevig, verdrietig, huilt, huilen, blij, tevreden, rustig, onrustig, apathisch, verward, somber, niet blij, vrolijk	modulating affect, affect plane, emotional, sad, sad, cries, crying, happy, content, quiet, uneasy, apathetic, confused, gloomy, not happy, merry
Exercise tolerance	saturatie, saturatiedip, saturatie dip, conditie, snel vermoeid, vermoeid, uitgeput, snel moe, sport	saturation, saturationdip, saturation sip, condition, quickly tired, tired, exhausted, tired quickly, sport
Work and employment	kantoor, bouw, niet naar school, les	office, construction, not going to school, class

Table 3: Keywords used for the selection of triples during the creation of the similarity test set.

Of each triple type, an equal amount of sentences were randomly selected to be annotated through stratified sampling. Annotators were then asked to identify the odd one out and anonymize the triples. Each triple was annotated and anonymized by three different people. A total of 12 annotators provided 1400 triples three times with a label. Of these triples, only those whereupon all three annotators agreed were selected, resulting in 824 triples to compare the models' performance to. The complete annotated data set is available online[4].

4.2 Extrinsic evaluation

Extrinsic evaluation on a domain-specific task was performed by fine-tuning the models on one of the ICF-classification data sets created for the a-proof project. As described in Section 4.1, annotators provided sentences or parts of sentences in hospital notes with one of five labels, of which four were

[4] https://github.com/cltl-students/verkijk_stella_rma_thesis_dutch_medical_langauge_model/tree/master/src/similarity_test/data

ICF categories and one the *None* class. If any (part of a) sentence provided information about one of these categories, they were labelled as such.

The models were tested on a sentence classification task where the goal is to extract sentences that are about one of the four ICF categories (*Walking*; *Emotional functions*; *Exercise tolerance functions*, and *Work and employment*). As discussed in Section 2, the classification of functioning in medical data has been an understudied area, even though it is highly useful information for clinical decision making. Testing the models on a task like this provides an outlook on how useful the models would be for any current or future project in the medical world.

The division of train and test data (80/20 split) as well as the amount of downsampling on the majority class *None* in the training data was copied from the a-proof project. The *None* class was downsampled to 25% of the original amount, as experiments performed during the a-proof project with different amounts of downsampling indicated that this amount led to the best performance. This resulted in a total of 42,444 sentences from 2,640 notes in the training set, and 40,214 sentences from 739 notes in the test set. It should also be noted that two of the ICF domains, namely *Exercise tolerance functions* and especially *Work and employment*, were underrepresented in this specific dataset.

For the evaluation of the models on a general, non-medical NLP task, we chose to test on Named Entity Recognition, since this is one of the most common sequence labelling tasks in NLP with available datasets in Dutch. We used the CoNLL 2002 data for Dutch Named Entity Recognition (Tjong Kim Sang 2002). The CoNLL-2002 data consist of news articles, namely of four editions of the Belgian newspaper 'DeMorgen' of 2000. These data can be downloaded online [5]. For this paper, the *ned_train* file was used for training and the *ned_testb* file was used for testing.

All fine-tuning for extrinsic evaluation was performed with the help of the simpletransformers library (`https://simpletransformers.ai/about`).

4.3 Evaluation of intermediate pre-training checkpoints

In order to see how much pre-training time was needed for the domain-specific model to reach optimal performance, the pre-training process was monitored by evaluating intermediate checkpoints. This was done by saving specific checkpoints and i) testing them on the intrinsic similarity test set and ii) fine-tuning and evaluating them on the extrinsic ICF data sets. In both cases, checkpoints were taken at 10k steps, 20k steps, 44k steps and 70k steps. The final model, at around 92k steps, was also included in the analysis.

4.4 Overview of evaluation data

All data used for the evaluation of the models were deleted from the data used for pre-training. For an overview of all data used for evaluation see Table 4.

task	Sim				ICF		NER	
subset of data	T1	T2	T3	T4	train	test	train	test
sentences	194	214	268	148	42444	40214	15806	5195
notes	-	-	-	-	2640	739	287	119

Table 4: Volume of data per data set used for evaluation. For the similarity test set, the amount of sentences per triple type is given. For the ICF and NER tasks, the amount of sentences per train and test set are given, as well as the amount of documents these sentences were sourced from.

[5] `https://www.clips.uantwerpen.be/conll2002/ner/`

5. Results

This section shows and discusses the results from the evaluation of the two medical language models compared to general models for Dutch (BERTje, RobBERT, Multilingual BERT (mBERT)). In order to maintain a clear distinction between the two medical language models, we will call them the From Scratch model and the Extended model. The From Scratch model is elsewhere in this paper also referred to as MedRoBERTa.nl. We start discussing the results on the intrinsic similarity test in 5.1 and proceed with the results on ICF classification and general Dutch NER in 5.2. The section will end with the results on the evaluation of intermediary checkpoints of the From Scratch model 5.3.

5.1 Intrinsic tests

In Table 5 the accuracy per model is reported on the complete test set and per triple type. The performance of the Medical Extended RobBERT model only after re-training the embedding look-up layer (Extended RobBERT Frozen) is also reported. For an overview of the triple types, see Section 4.1. For the keywords used in the selection of triple types 1 and 3, see Table 3.

	All triples	T1	T2	T3	T4
mBERT	0.57	0.56	0.48	0.68	0.52
BERTje	0.58	0.59	0.49	0.68	0.54
RobBERT	0.57	0.56	0.44	0.68	**0.57**
From Scratch	**0.65**	**0.65**	**0.56**	**0.76**	**0.57**
Ext. RobBERT Frozen	0.52	0.55	0.41	0.58	0.53
Ext. RobBERT Final	0.58	0.60	0.48	0.65	0.54
Support	824	194	214	268	148

Table 5: Accuracy per model on complete test set and per triple type

Looking at the scores over the complete dataset with all triples, Table 5 shows that the Medical From Scratch model outperforms all other models. It has an overall accuracy score that is .07 higher than the best performing general language model for Dutch, in this case BERTje. The Frozen RobBERT model has the worst performance, which might indicate that only retraining the lexical layer without adapting the transformer layers to the domain is not sufficient for this genre. The final Extended RobBERT model has the same performance as BERTje, the best performing general language model. It seems that even with one epoch of training for domain adaptation in the transformer layers has not been sufficient training to unlearn general patterns of Dutch. All general language models have similar performance.

Considering the scores per triple type in Table 5, the Medical From Scratch Model outperforms the best general Dutch language model with .06 for type 1, .07 for type 2 and .08 for type 3, but equals RobBERT in performance for type 4. This indicates that the language used to differentiate between different levels of functioning within a domain is less domain-specific than the language used to introduce an ICF domain. Unsurprisingly, all models perform best when judging type 3, which also seems intuitively easier to judge, as type 3 contains one sentence that is not only from a different domain but also lacks the overlapping keyword the other two sentences feature. Type 3 was also the type that was easiest to judge for the human annotators: it has the biggest support in the dataset, which means that it was the type that was most agreed upon by individual annotators.

A downside of the inclusion of overlapping keywords in the similarity dataset is that regardless of the quality of the representation the model has for these keywords, models will be cued by the fact that they match. For example, RobBERT might have an embedding for ‘transfer’ that does not match with what it often means in medical text (a transfer from the airport to your hotel versus a transfer of a patient from their bed to a chair, for example), but it would still match with the other embedding for ‘transfer’ in a second sentence.

In order to see how models perform when they are not cued by overlapping keywords, another test was performed where all keywords that were used to match triples on were masked. The results of this test are reported in Table 6. Triple types in boldface are triple types that originally contained overlapping keywords. Because all instances of these keywords were removed (and not only those appearing in triple types 1 and 3), the scores for triple types 2 and 4 also differ from those reported in Table 5. It was chosen to replace the keywords with the *mask* special token because the models recognise this as a token that could be anything (or link to any token-ID and thus to any embedding). Replacing them with any other token would have meant that the overlap between the two tokens remains.

	All triples	**T1**	T2	**T3**	T4
mBERT	0.51	0.44	0.49	0.57	**0.55**
BERTje	0.53	0.47	0.49	0.62	0.50
RobBERT	0.53	0.51	0.41	**0.63**	0.53
From Scratch	**0.57**	**0.55**	**0.55**	**0.63**	0.53
Ext. RobBERT Frozen	0.49	0.52	0.41	0.53	0.51
Ext. RobBERT Final	0.51	0.54	0.46	0.54	0.51
Support	824	194	214	268	148

Table 6: Accuracy per model on complete test set and per triple type, with keywords removed

The first observation that can be made by looking at the results in Table 6 is that the performance of all language models goes down once the keywords are removed from the test set. The performance of the From Scratch model and the final Extended RobBERT model drop the most, with 0.08 and 0.07 points respectively. RobBERT and the Frozen Extended RobBERT model drop the least, with 0.04 and 0.03 points respectively. We hypothesize that the reason behind the different drops in performance is that the From Scratch and the Final Extended RobBERT model had more accurate embeddings for these keywords than the other models, as the keywords tend to be either medical terms or frequent terms in the training data. Removing these keywords thus means a bigger loss of information for the medical models than for the general language models. Again, the low drop in performance from the Frozen Extended RobBERT model indicates that at this point, after only re-training the embedding look-up layer, the model does not yet have a good understanding of the medical language data. The Medical From Scratch model remains the model that performs best overall, though with less difference, having a combined accuracy score of .04 over the best performing general language models (BERTje and RobBERT). Looking at the performance for the different triple types, the From Scratch lost most advantage in triple type 3, indicating it relied more on the keywords to judge this triple than the other models.

Both the test on the original similarity test set as the one where the keywords were removed show that the Medical From Scratch model internalized most semantic knowledge on medical language: it delivers the most accurate sentence embeddings for this domain. The fact that BERTje performs slightly better than RobBERT might have to do with the fact that RobBERT was trained on much more data during the pre-training phase (39 GB vs. 12.1 GB), which might mean that RobBERT has internalised more patterns of general Dutch that are unproductive for understanding medical Dutch language.

5.2 Extrinsic tests

For the extrinsic in-domain task, each language model was fine-tuned 8 times on the ICF dataset. This resulted in 8 fine-tuned models per model type (a total of 32 models) that each predicted on the same test set. Each model was evaluated on these predictions at sentence level. Table 7 shows the average F1 scores over these 8 models per model type for classifying ICF domains on sentence level.

The results on the extrinsic out-of-domain task, namely Named Entity Recognition, did not differ much per fine-tuned model, so for this task, only three runs were performed. Results on the NER task are presented in Table 8.

	RobBERT	**BERTje**	**From Scratch**	**Ext. RobBERT**
Walking	0.62	0.62	***0.65**	0.62
Emotional functions	0.66	**0.69**	0.67	0.66
Exercise Tolerance	0.42	**0.45**	**0.45**	**0.45**
Work and employment	**0.40**	**0.40**	0.39	0.39

Table 7: F1 scores on sentence level per label per fine-tuned language model type. Scores are averaged over 8 runs. * p-value of 0.01

	Precision	**Recall**	**F1**
BERTje	**0.91**	**0.92**	**0.91**
RobBERT	0.84	0.85	0.84
From Scratch	0.64	0.68	0.66
Ext. RobBERT	0.68	0.72	0.70

Table 8: F1 scores on NER per fine-tuned language model type. Per language model type. Scores are averaged over 3 runs.

In general, Table 7 shows that all models score low on *Exercise tolerance functions* and *Employment and work.* This is because the dataset is very unbalanced: *Exercise tolerance functions* and *Employment and work* are underrepresented. Therefore, the analysis of the results will focus on *Walking* and *Emotional functions.* Among all 32 models that were tested, precision was consistently lower than recall for all ICF domains.

Looking at the performance on sentence level (Table 7), the only times where the best general language model and the best medical language model are more than 0.01 point apart in performance are for *Walking* and for *Emotional functions.* For *Walking*, the Medical From Scratch model is better than both RobBERT and BERTje. Because the scores per model also differed considerably per fine-tuning run, a statistical test was conducted to see if these differences were significant. Because the samples on which the models are trained and tested are not normally distributed, a Wilcoxon Signed Rank Test (WSRT) was conducted over the separate scores per run. We set a threshold for significance at a p-value of 0.05. For *Walking*, the Medical From Scratch Model was compared to BERTje, as BERTje performed slightly better than RobBERT looking at the unrounded averages. The WSRT showed that the From Scratch model differs from BERTje with a p-value of 0.01. As for *Emotional functions*, the WSRT showed that BERTje differed from the Medical From Scratch Model with a p-value of 0.03. This leads to think that the language used to describe the *Walking* class is more divergent from general Dutch than the language used to describe the *Emotional functions* class.

It seems that the fine-tuning process adapts models in such a way that it is able to skew even less appropriate models to do well on domain-specific downstream tasks. However, it is important to note that the medical models were trained in much less optimal environments with considerably less computational power. It can be said that when fine-tuned, a domain-specific language model will do better at classifying language that is extremely specific to the domain, but will be comparable to bigger, general language models for other tasks.

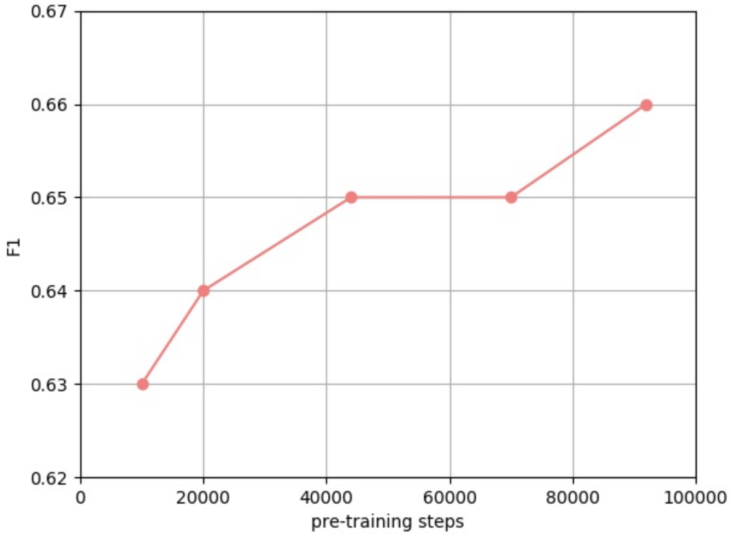

Figure 3: Performance of intermediate checkpoints of the From Scratch Medical Model on similarity test set

The results on Dutch NER presented in Table 8 show very clearly that the domain-specific medical language models perform worse than the general language models for Dutch, as expected. As opposed to the trends in the intrinsic evaluation and the evaluation on the domain-specific ICF classification, the Extended RobBERT model performs better at NER than the From Scratch model. This highlights even more that the From Scratch model internalised more domain-specific language, which makes it unsuited for tasks within the sphere of general Dutch. It demonstrates how different the language in hospital notes is, and how the language use in the training data has great influence on the model that it yields.

5.3 Testing intermediate checkpoints

As the results from the intrinsic evaluation showed that the medical model trained from scratch performed best, this model was taken to be analysed further through the evaluation of intermediate checkpoints (see Section 4.3). Table 3 shows the performance of intermediate checkpoints of the from scratch model on the odd-one-out similarity test , and 4 shows the performance of intermediate checkpoints of the from scratch model on the ICF sentence classification task. For the latter, only the performance on the two ICF domains that were sufficiently represented in the dataset are given.

Figure 3, evaluating intermediate checkpoints on the similarity task, shows a more or less steady increase as pre-training time passes. However, it is interesting to see that the model already reaches an accuracy of 0.63 after only 10k training steps. This is a performance of already .04 higher than mBERT and RobBERT and .05 higher than BERTje (See Table 5).

When looking at the performance of the intermediate checkpoints on the ICF task, we see that performance does not necessarily go up as training time passes. Performance simply does not change much. A similar observation can be made as with the intrinsic evaluation of the intermediate checkpoints: after only 10k steps, the Medical From Scratch model reaches similar performance to published general language models for Dutch (see Table 7).

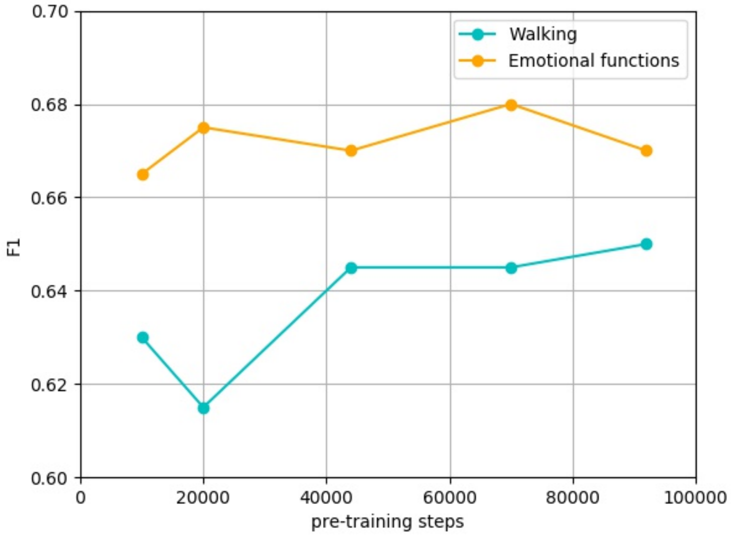

Figure 4: Performance at sentence level of fine-tuned models of intermediate checkpoints of the From Scratch Medical Model on classifying *Walking & Moving* and *Mood* from the ICF test set. Results are averaged over 2 fine-tuned models per intermediate checkpoint.

6. Discussion

This paper presented MedRoBERTa.nl as the first language model specifically for Dutch medical language. We have shown that it is possible to build a domain-specific language model outperforming general language models for domain-specific tasks with limited computational power. We also show that when in possession of sufficient in-domain data, it seems preferable to train from scratch than to extend pre-training on an existing model, even if you include a domain-specific vocabulary and re-train the embedding look-up layer in the case of the latter. Especially when testing the models' embeddings' accuracy, MedRoBERTa.nl shows improvement over general language models for Dutch. When fine-tuned, the domain-specific model only has an advantage if the task is very specific to the domain as well and features domain-specific language. We also show that domain-specific language models need much less pre-training time than general language models to perform better than them. We hope this finding motivates individual researchers with less computational resources available and will help in reducing excessive pollution caused by technology in the future.

For future research, it would be interesting to test whether the method we adopted for the creation of a domain-specific model leads to the same results for domains in other languages. It would also be interesting to repeat our experiment, but using a BERT architecture instead of a RoBERTa architecture, to see if our expectation that RoBERTa would be more fitting for this specific domain would prove to be true. It would also be worthwhile to test if other domain-specific models also need less pre-training time to reach near-optimal performance, as shown in this study and the one by Müller et al. (2020).

7. Acknowledgements

We would like to thank Edwin Geleijn and the a-proof team for making this research possible. We would like to thank Wietse de Vries and Pieter Delobelle for their advice on how to train the models.

The GPUs used in this research were financed by the NWO Spinoza Project assigned to Piek Vossen (project number SPI 63-260). The pilot study that created the ICF dataset that we used to fine-tune our models on was financed by the Corona Research Fund (project number 2007793 - COVID 19 Textmining).

References

Belinkov, Yonatan and James Glass (2019), Analysis methods in neural language processing: A survey, *Transactions of the Association for Computational Linguistics* **7**, pp. 49–72, MIT Press.

Beltagy, Iz, Kyle Lo, and Arman Cohan (2019), Scibert: A pretrained language model for scientific text, *EMNLP/IJCNLP*.

Cer, Daniel, Mona Diab, Eneko Agirre, Inigo Lopez-Gazpio, and Lucia Specia (2017), Semeval-2017 task 1: Semantic textual similarity-multilingual and cross-lingual focused evaluation, *arXiv preprint arXiv:1708.00055*.

Chalkidis, Ilias, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos (2020), Legal-bert: The muppets straight out of law school, *ArXiv*.

de Vries, Wietse and Malvina Nissim (2020), As good as new. how to successfully recycle english gpt-2 to make models for other languages, *arXiv preprint arXiv:2012.05628*.

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

Devlin, J., Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2019), Bert: Pre-training of deep bidirectional transformers for language understanding, *NAACL-HLT*.

Gu, Yu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon (2020), Domain-specific language model pretraining for biomedical natural language processing, *ArXiv*.

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.

Huang, Kexin, Jaan Altosaar, and Rajesh Ranganath (2019), Clinicalbert: Modeling clinical notes and predicting hospital readmission, *arXiv preprint arXiv:1904.05342*.

Joshi, Mandar, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy (2020), Spanbert: Improving pre-training by representing and predicting spans, *Transactions of the Association for Computational Linguistics* **8**, pp. 64–77, MIT Press.

Kingma, Diederik P and Jimmy Ba (2014), Adam: A method for stochastic optimization, *arXiv preprint arXiv:1412.6980*.

Lan, Zhenzhong, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut (2019), Albert: A lite bert for self-supervised learning of language representations, *arXiv preprint arXiv:1909.11942*.

Lee, Jinhyuk, Wonjin Yoon, Sungdong Kim, D. Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang (2020), Biobert: a pre-trained biomedical language representation model for biomedical text mining, *Bioinformatics* **36**, pp. 1234 – 1240.

Liu, Y., Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, M. Lewis, Luke Zettlemoyer, and Veselin Stoyanov (2019), Roberta: A robustly optimized bert pretraining approach, *ArXiv*.

Mickus, Timothee, Denis Paperno, Mathieu Constant, and Kees van Deemter (2019), What do you mean, bert? assessing bert as a distributional semantics model, *arXiv preprint arXiv:1911.05758*.

Müller, M., M. Salathé, and P. Kummervold (2020), Covid-twitter-bert: A natural language processing model to analyse covid-19 content on twitter, *ArXiv*.

Rogers, Anna, Olga Kovaleva, and Anna Rumshisky (2020), A primer in bertology: What we know about how bert works, *Transactions of the Association for Computational Linguistics* **8**, pp. 842–866, MIT Press.

Tai, Wen-Hsin, H. T. Kung, and Xin Dong (2020), exbert: Extending pre-trained models with domain-specific vocabulary under constrained training resources, *EMNLP*.

Tjong Kim Sang, Erik F. (2002), Introduction to the CoNLL-2002 shared task: Language-independent named entity recognition, *COLING-02: The 6th Conference on Natural Language Learning 2002 (CoNLL-2002)*. https://aclanthology.org/W02-2024.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin (2017), Attention is all you need, *Advances in neural information processing systems*, pp. 5998–6008.

Wang, Changhan, Kyunghyun Cho, and Jiatao Gu (2020), Neural machine translation with byte-level subwords, *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34, pp. 9154–9160.

Zhu, Yukun, Ryan Kiros, Richard S. Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler (2015), Aligning books and movies: Towards story-like visual explanations by watching movies and reading books, *2015 IEEE International Conference on Computer Vision (ICCV)* pp. 19–27.