Berliner Hochschule für Technik

Domain Adaptation of Transformer-based Language Models via Low Layer Information Integration

Master Thesis of

Dennis Fast
Matriculation Number: 930034

Berliner Hochschule für Technik (BHT)
Department VI - Data Science (Master of Science)
Data Science and Text-based Information Systems (DATEXIS)

First Reviewer:	Prof. Dr.-Ing. habil. Alexander Löser
Second Reviewer:	Prof. Dr. rer. nat. Felix Bießmann
Advisors:	Paul Grundmann, Tom Oberhauser

September 04, 2023

Abstract

The thesis presents an in-depth exploration of strategies to enhance the performance of transformer-based language models in the context of domain adaptation. The study focuses on biomedical language understanding tasks. It investigates the modification approaches of vocabulary and token embedding matrix of the BERT-based models to improve their adaptation to a specific domain.

The main approaches examined in this work, include vocabulary reduction and augmentation combined with token embedding re-initialisation aiming to investigate the knowledge transfer capability of model vocabularies. Through comprehensive experimentation, we evaluate the impact of these approaches on performance on individual tasks from the Bio-LM benchmark.

Vocabulary reduction strategies, particularly the "last" heuristic, reveal benefits in noise reduction and performance improvement in specific tasks. However, the trade-off between vocabulary reduction and the potential loss of relevant terms is acknowledged.

Knowledge transfer approaches are explored, emphasising syntactic and semantic transfer approaches. The experiments emphasize the complexity of initializing embeddings of domain-specific tokens, highlighting the challenge of capturing the semantic meaning by averaging embeddings of the sub-tokens or the contextual neighbours.

Overall, the findings reveal complex relationships between model modifications and task performance. While some strategies lead to improvements in specific tasks, no single approach consistently surpasses the source model across all tasks.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In the context of modern Natural Language Processing (NLP), a Transformer-based language model refers to a type of artificial intelligence model that was designed to understand and generate human language.

Transformer architecture has revolutionized the field of NLP. By leveraging self-attention and multi-head attention mechanisms, transformer-based language models can capture long-range dependencies in text data, rather than relying only on sequential processing as in traditional recurrent neural networks (RNNs). The use of transformer architectures has led to significant improvements in language understanding and generation tasks, making them the backbone of many state-of-the-art NLP systems.

Transformer-based models have demonstrated remarkable performance improvement in various textual tasks compared to previous generations of NLP models, including text classification, sequence labelling, machine translation, and language generation. These models are typically pre-trained on large corpora of text data, such as book collections or data from the internet, to learn the underlying language patterns. This pre-training phase is followed by fine-tuning on specific downstream tasks, enabling the models to adapt to specific domains and tasks.

1.1 Motivation

Despite the outstanding capabilities, the Transformer-based language models often lack domain-specific knowledge, which can be crucial for specialized domains such as biomedical research. Domain adaptation techniques offer a promising solution to bridge this gap and improve the performance of general models in specific domains.

The biomedical domain, in particular, presents unique challenges due to its specialized vocabulary, complex terminology, and large amounts of domain-specific text data. By adapting a general language model to biomedical research using domain-specific knowledge, we can unlock its full potential in tasks like document classification, entity recognition, and question-answering within the biomedical field. This has the potential to significantly benefit researchers, healthcare professionals, and other stakeholders in the domain.

1.2 Objective

The objective of the master's thesis is to explore the feasibility of knowledge transfer from a domain-specific to a general transformer-based language model through a transfer of domain-specific tokens and their corresponding token embeddings without further fine-tuning the general model on the domain-specific corpora.

Specifically, we aim to investigate the potential benefits of adapting a general model BERT [1] (source model), using the vocabulary of the biomedical model PubMedBERT [2] (target model). By leveraging the transfer learning approaches proposed in this thesis, we seek to enhance the performance of the general model on domain-specific tasks, particularly on the Bio-LM benchmark.

1.3 Outline

This thesis is organized into several chapters, each addressing a specific aspect of domain adaptation and knowledge transfer for transformer-based language models. The structure of the thesis is as follows:

Chapter 2 “Background” provides a short introduction to transformer-based models used in the course of experiments, we focus on their pre-training specifics regarding acquiring and storing domain-specific knowledge. We then list the main knowledge transfer techniques utilized by transformer-based models, addressing their application areas, limitations, and how our approach involving contextual token embeddings for knowledge transfer fits in. Finally, we present an overview of previous research papers in the areas of vocabulary reduction, adaptation and transfer.

Chapter 3 “Preliminary Analysis” deals with an in-depth exploration of model vocabularies and contextual word embeddings in transformer-based language models. We examine the relationship between vocabulary size and model size, analyze common tokens across models, and study token length distribution. Then, we visualize the token embeddings using t-SNE to reveal syntactic and semantic patterns. The section also examines cosine similarity’s ability to capture the contextual meaning of the tokens.

Chapter 4 ”Methodology” is structured to systematically investigate the capability of token embeddings in transferring domain-specific knowledge across pre-trained language models. It begins by detailing the overall research approach, focused on measuring the performance of models on target domain tasks after the knowledge transfer step. Based on previous research and the preliminary analysis chapter, we present a series of hypotheses and provide a guiding framework for evaluating the impact of adaptation, pre-initialization, vocabulary reduction, and knowledge transfer techniques.

In Chapter 5 ”Implementation” we focus on the practical aspects of our study. We explain how we set up and conducted our experiments, provide details on the utilized models and present the evaluation pipeline designed to ensure a systematic and consistent evaluation process. Additionally, we address the hyperparameter optimization search space.

In Chapter 6 ”Evaluation and Discussion” we present our experiment outcomes and discuss the implications, limitations, and potential drawbacks of our experimental setup and findings. We first reveal the most stable set of hyperparameters, which we then use in the course of our experiments. Next, we go deeper into the evaluation and discussion of the experiment results, followed by an analysis of the possible causes for insufficient model performance.

Chapter 7 "Conclusion" evaluates our hypotheses from Chapter "Methodology", summarizes the main findings of the presented work, and presents possible directions for future work.

Chapter 2

Background and Related Work

This chapter provides an introduction to the essential concepts we will need to understand the recent research in areas of knowledge distillation and transfer and design novel approaches to improve state-of-the-art techniques.

2.1 Transformer-based Language Models

The field of Natural Language Processing (NLP) has undergone a transformative evolution, marked by significant breakthroughs in understanding and generating human language. Before the steep popularity rise of transformer-based models, the NLP landscape was dominated by techniques that heavily relied on recurrent neural networks (RNNs) and convolutional neural networks (CNNs). While these approaches showed promise, they struggled to capture long-range dependencies and contextual nuances present in the human language.

The landmark paper ”Attention is All You Need” [3] by Vaswani et al. (2017) introduced the Transformer architecture - an innovation that marked a new era for NLP. At its core, the Transformer introduced the concepts of self-attention and multi-head attention mechanisms, allowing the model to efficiently weigh the significance of each word in a sequence relative to all other words. This breakthrough design tackled the challenges of capturing contextual relationships across long distances within the text, leading to remarkable improvements in tasks like machine translation, text generation and sentiment analysis.

The Transformer architecture consists of two key components: the self-attention mechanism and feed-forward neural networks. The self-attention mechanism allows each word in a sequence to focus on relevant words, both nearby and distant, enabling the model to comprehend the contextual nuances of language. The feed-forward neural networks process the attended representations, further refining the model's understanding and generating more accurate predictions.

Since the introduction of the Transformer, the NLP landscape has experienced the next paradigm shift. Pretrained transformer-based language models, such as BERT [1] and GPT series [4][5][6][7], and their subsequent iterations, have become the cornerstone of modern NLP research and applications. These models have demonstrated remarkable capabilities in a wide range of tasks, including text classification, sequence labelling, question answering, and even tasks beyond traditional NLP, like protein folding prediction and code generation.

Moreover, the Transformer architecture has given rise to the concept of "transfer learning" in NLP. Researchers and practitioners now commonly employ pre-trained models on large text corpora and fine-tune them on specific tasks with limited task-specific data. This approach has democratized NLP advancements, making it accessible to a broader audience and accelerating progress in various applications.

Next, we explore the inner workings of three transformer-based language models, exploring their architecture, training strategies and applications.

2.1.1 BERT

Bidirectional Encoder Representations from Transformers (BERT) [1] is a model based on the transformer architecture. Developed by researchers at Google AI in 2018, the model marked a significant advancement in the domain of pre-trained language models. BERT was designed to address the limitations of previous models that only considered left-to-right or right-to-left context in a text sequence and had just a few applications.

There are three main innovations in BERT compared to the original transformer.

1. **Bidirectional self-attention mechanism**: The self-attention mechanism of BERT is bidirectional rather than unidirectional. The original transformer model only received the previous context of a token, meaning the self-attention was applied only to the tokens before. In contrast, the BERT model can apply self-attention to both the left and right from the current token, allowing it to receive more context during the training of the model.

2. **Training objectives**: BERT utilizes the following training objectives as part of its training routine:

    - Masked-language modelling (MLM): predicting missing tokens in a sentence by considering both the preceding and following words, which allows capturing of richer contextual information);

    - Next-sentence prediction (NSP): determining if two sentences are consecutive in the original text, improve understanding of longer-term dependencies across sentences).

3. **Large training dataset**: The original transformer model was not explicitly associated with a specific training dataset. Its effectiveness was demonstrated through experiments on machine translation tasks using various language pairs from publicly available datasets, such as WMT and IWSLT. In contrast, BERT was explicitly trained on a massive and diverse text corpus that consisted of a mixture of books, articles, and web pages. The training dataset included the BooksCorpus dataset (800 Mio. words), which contained text from a wide array of books, and the English Wikipedia corpus (2.5B words), providing information on a diverse range of topics and allowing BERT to learn rich semantic representations of language.

In implementing these changes to the original Transformers, the researchers were able to achieve state-of-the-art performance on a range of different tasks, like text classification, named entity recognition, and sentiment analysis.

2.1.2 BlueBERT

In the research paper ”Transfer Learning in Biomedical Natural Language Processing: An Evaluation of BERT and ELMo on Ten Benchmarking Datasets” [8] by Peng et al. (2019) introduced two models with the same architecture as the original BERT and used its model weights as starting point. Then, the models were further pre-trained on two different biomedical corpora: PubMed abstracts (ca. 4B words) and MIMIC-III (ca. 500 Mio words). The researchers used the same model configuration and training parameters as Devlin et al. in the original BERT paper. They evaluated the models' performance on the biomedical benchmark BLUE which was introduced in the same paper.

The researchers could show that further domain-specific pre-training of the general-domain language model enables the transfer learning ability and improves the performance of the model on that specific domain significantly.

BlueBERT's applications cover various tasks such as named entity recognition for medical entities, relation extraction, and biomedical text classification.

2.1.3 PubMedBERT

The research paper ”Domain-Specific Language Model Pretraining for Biomedical Natural Language Processing” [2] by Gu et al. (2020) showed that domain-specific pre-training from scratch outperforms continual pre-training of generic language models (e.g. BlueBERT) and demonstrated that the previous assumption that mixed-domain pre-training supports transfer learning is not always applicable.

Both pre-training paradigms are illustrated in Figure 2.1 below (from [2]).

The researchers introduced two language models that have the same architecture as BERT but they discarded its original model weights and pre-trained two models on the biomedical corpora from scratch. The first model was pre-trained on the PubMed ab-

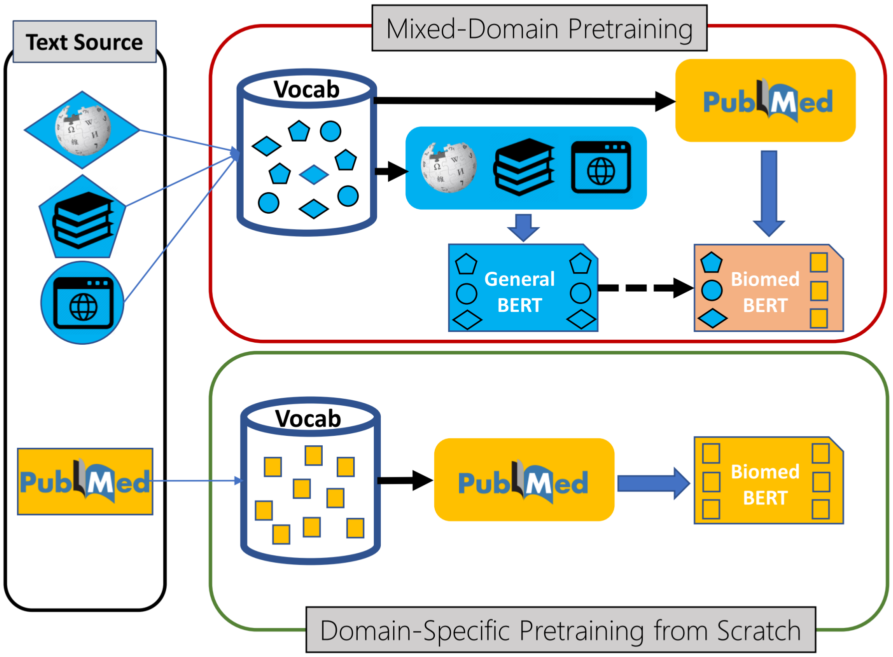


Figure 2.1: Two paradigms for pre-training of transformer-based language models

stracts only (ca. 3.1B words as of February 2020), and the second one was pre-trained on the PubMed abstracts and PMC full-text additionally (size of PMC full-text corpus not stated).

They followed the standard pre-training procedure introduced in the BERT paper, the only difference was that they used whole-word masking (WWM) with a masking rate of 15% instead of the standard MLM procedure. They noticed that when a word is only partially masked, it is relatively easy to predict the masked portion given the observed ones. In contrast, WWM enforces that the whole word must be masked if one of its sub-words is chosen, forcing the language model to capture more contextual semantic dependencies and improving its performance on the downstream tasks.

They evaluated both models on a new benchmark BLURB (Biomedical Language Understanding & Reasoning Benchmark) introduced in the same paper, which is comprised of a comprehensive set of biomedical NLP tasks from publicly available datasets, including named entity recognition (NER), evidence-based medical information extraction (PICO), relation extraction, sentence similarity, document classification, and question answering.

2.2 WordPiece Tokenizer

Tokenizers are crucial components of all modern NLP models that convert text into individual units, or tokens, for further analysis by NLP models. These tokens can range from words over subword units to single characters. All models utilized in our experiments use the so-called WordPiece tokenizer. Next, we explain how it works.

2.2.1 Vocabulary Creation

If we want to train a transformer-based language model on text data from scratch, the first thing we need to do is create a corresponding vocabulary.

- **Token ID**: Each token in the vocabulary is assigned to a unique integer ID for consistency

- **Vocabulary Initialization**: The process starts with creating a small vocabulary including all single characters occurring in the text data and the special tokens: [CLS] (beginning of a sequence), [SEP] (separator between sequences), and [MASK] (used for masked language modelling tasks in pre-training).

- **Filling the Vocabulary**: After initialization, WordPiece learns merge rules. Since it identifies subwords by adding a prefix ##, each word is initially split by adding that prefix to all the characters inside the word. After that, the tokenizer computes a score for each subtoken pair, using pairwise appearance frequency:

$$score = \frac{f(t_1 t_2)}{f(t_1) \cdot f(t_2)} \tag{2.1}$$

  By dividing the frequency of the pair by the product of the frequencies of each of its parts, the algorithm prioritizes the merging of pairs where the individual parts are less frequent in the vocabulary. The size of the vocabulary is arbitrary and is often optimized by researchers as a hyperparameter in pre-training for the best balance between computational effort and performance.

2.2.2 Tokenization

When a text is fed into the tokenizer after initial training, it's split into sequences of subword units. For example, the word "playing" might be tokenized into {'play', '##ing'}, where '##' indicates that 'ing' is a continuation of the previous subword.

2.2.3 Out-of-Vocabulary Handling

If the tokenizer encounters a word not present in the vocabulary, it further breaks down the word into characters or smaller subword units. This ensures that even rare or unknown words can be represented using the available subword tokens.

2.2.4 Limitation

While the WordPiece Tokenizer is effective, it has limitations. Long words can be broken into many subword tokens, potentially impacting model efficiency and interpretation. Also, the tokenizer may sometimes segment words in ways that seem linguistically odd, like 'unhappiness' being tokenized as {'un', '##h', '##appy', '##ness'}.

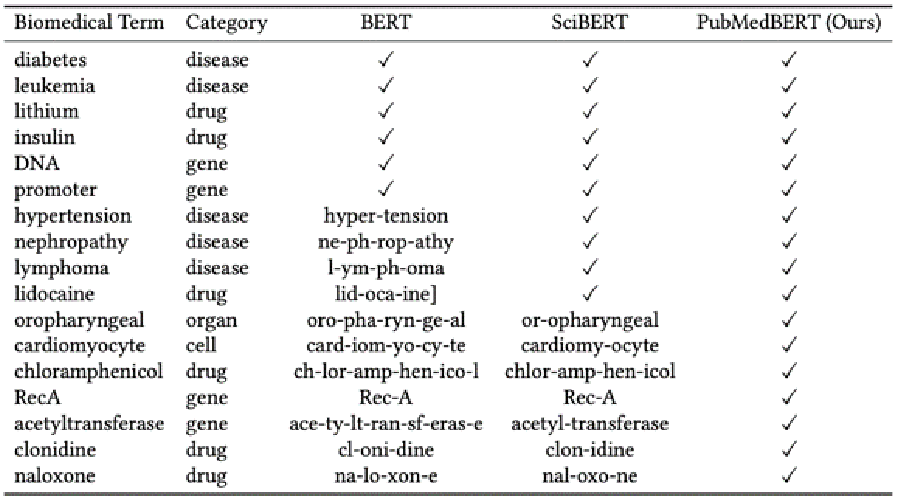
2.3 Domain-specific Vocabulary

Since a model vocabulary represents the set of unique tokens or sub-word units, it directly impacts the model's ability to understand text, as it determines which linguistic units the model can recognize and manipulate during processing. The vocabularies of different models highly vary based on factors such as the model's training data, language, and domain focus. Here are the main differences in the vocabularies of the models we use in our experiments:

- **BERT**: BERT's vocabulary includes a wide range of sub-word units, making it capable of representing both common words and specialized terms, but its understanding of domain-specific language is very limited.

- **BlueBERT**: Since BlueBERT is fully based on BERT and was further pre-trained on the biomedical text data, their vocabularies are identical.
- **PubMedBERT**: As PubMedBERT was designed for the biomedical domain specifically and pre-trained from scratch, its vocabulary reflects this specialization. As a result, the vocabulary of PubMedBERT includes a diverse set of biomedical terms, domain-specific terminology, medical jargon, and scientific terms that are essential for understanding and processing biomedical text.

This difference was shown by the researchers in the PubMedBERT paper [2] and can be seen in Table 2.1. The table shows that the more specialised the term is, the larger the amount of sub-tokens that BERT outputs.



Biomedical Term	Category	BERT	SciBERT	PubMedBERT (Ours)
diabetes	disease	✓	✓	✓
leukemia	disease	✓	✓	✓
lithium	drug	✓	✓	✓
insulin	drug	✓	✓	✓
DNA	gene	✓	✓	✓
promoter	gene	✓	✓	✓
hypertension	disease	hyper-tension	✓	✓
nephropathy	disease	ne-ph-rop-athy	✓	✓
lymphoma	disease	l-ym-ph-oma	✓	✓
lidocaine	drug	lid-oca-ine]	✓	✓
oropharyngeal	organ	oro-pha-ryn-ge-al	or-opharyngeal	✓
cardiomyocyte	cell	card-iom-yo-cy-te	cardiomy-ocyte	✓
chloramphenicol	drug	ch-lor-amp-hen-ico-l	chlor-amp-hen-icol	✓
RecA	gene	Rec-A	Rec-A	✓
acetyltransferase	gene	ace-ty-lt-ran-sf-eras-e	acetyl-transferase	✓
clonidine	drug	cl-oni-dine	clon-idine	✓
naloxone	drug	na-lo-xon-e	nal-oxo-ne	✓

Table 2.1: Tokenization of biomedical terms by different models

2.4 Word embeddings

Word embeddings have significantly improved the NLP field of study by providing a powerful way to represent words as dense, low-dimensional vectors. The development of word embeddings marked a shift from traditional one-hot encoding representations to continuous vector representations that capture semantic and syntactic relationships between words.

2.4.1 Static Word Embeddings

Static word embeddings represent each word with a fixed vector regardless of its context within a sentence or document. They are pre-trained on large corpora and can be directly utilized in downstream NLP tasks.

Mathematically, let's denote the vocabulary as V and the dimensionality of word vectors as d. For a static word embedding model, each word $w \in V$ is associated with a d-dimensional vector representation, denoted as $E(w) \in \mathbb{R}^d$.

One of the early milestones in static word embeddings was the introduction of word2vec [9], [10] by Mikolov et al. in 2013. The first paper introduces two novel neural network models: Skip-gram and CBOW. The second paper introduces negative sampling which is an efficient way to train word2vec.

The word2vec model popularized the concept of distributed word representations by training neural networks on large text corpus to predict surrounding words given a target word or vice versa. This approach resulted in word vectors that showcased meaningful relationships between words, such as similarity and analogy.

Another influential model in the field of static word embeddings is GloVe (Global Vectors for Word Representation) [11], proposed by Pennington et al. in 2014. GloVe learns word embeddings by factorizing the word co-occurrence matrix, which captures the statistical information of word co-occurrences in a large corpus. The resulting word vectors capture both semantic and syntactic relationships between words.

2.4.2 Contextual Word Embeddings

In contrast to static word embeddings, contextual word embeddings capture the meaning of a word based on its surrounding context within a sentence or document. These embeddings take into account the variability in word meaning across different contexts, providing a more nuanced representation.

One prominent example of contextual word embeddings is ELMo (Embeddings from Language Models) [12], introduced by Peters et al. in 2018. ELMo generates word embeddings by training a bidirectional language model (bi-LSTM) on a large corpus, capturing both forward and backward contextual information. Each word is represented as a concatenation of the hidden states from different layers of the language model.

Mathematically, for a contextual word embedding model like ELMo, the representation of a word w at position i in a sentence can be denoted as $E(w, i) \in \mathbb{R}^d$, where $E(w, i)$ is a d-dimensional vector representing the contextualized word embedding.

2.5 Cosine Similarity of Contextual Word Embeddings

Cosine similarity is a measure used to determine the similarity between two non-zero vectors in a multi-dimensional space. It calculates the cosine of the angle between the vectors, which represents their directional similarity:

$$cossim(\mathbf{v_1}, \mathbf{v_2}) = \frac{\mathbf{v_1}\mathbf{v_2}}{\|\mathbf{v_1}\|\|\mathbf{v_2}\|} \tag{2.2}$$

The values range from -1 to 1, where a cosine similarity of 1 indicates that the vectors point in the same direction (perfect similarity), 0 indicates they are orthogonal (no similarity), and -1 indicates they point in completely opposite directions (complete dissimilarity).

In the context of NLP and contextual word embeddings generated by models like BERT, cosine similarity is applied to determine the similarity between the embeddings of two words or phrases. These embeddings capture the contextual meaning of words in the given context. The cosine similarity between these embeddings provides insight into how similar their contextual meanings are.

2.6 Knowledge Transfer

Knowledge transfer, in a general sense, refers to the process of transferring knowledge, skills, or insights from one entity (often referred to as the source) to another entity (referred to as the target). This concept is widely used in various areas of science, from education and cognitive psychology to machine learning and artificial intelligence. In education, it involves teachers transferring knowledge to students.

2.6.1 Knowledge Transfer in Machine Learning and NLP

In machine learning, knowledge transfer refers to the practice of leveraging knowledge learned from one task or domain to improve performance on another related task or domain. The idea main idea behind knowledge transfer is that a model that has learned useful features or representations from one dataset can use those insights to perform better on a different but related task. This approach can save computational resources and training time while enhancing the model's ability to generalize across tasks.

Knowledge transfer plays a crucial role in the field of NLP due to the scarcity of labelled data for every specific task. Pre-trained transformer-based language models like BERT, GPT, and their variants have demonstrated remarkable abilities in transferring linguistic and domain knowledge. These models are trained on large text corpora and learn contextualized word embeddings, which capture semantic and syntactic information. This initial training phase enables the model to understand the underlying structures and relationships in language, making it a valuable source of knowledge for downstream tasks.

2.6.2 Techniques of Knowledge Transfer in Transformer-Based Models

After the release of the Transformer paper, there have been many different techniques emerged to address the effective and efficient knowledge transfer by the Transformer-based language models:

- **Domain Adaptation**: This technique involves transferring knowledge from a source domain to a target domain. For instance, if a model is trained on news articles (source domain) and needs to perform well on medical text (target domain), domain adaptation techniques aim to adapt the model's knowledge to better understand and generate text relevant to the medical domain. Techniques like domain adversarial training and domain-specific fine-tuning can help achieve domain adaptation.

- **Task Adaptation**: Similar to domain adaptation, task adaptation focuses on transferring knowledge across different tasks. If a model is proficient in sentiment analysis (source task) and needs to perform named entity recognition (target task), task adaptation techniques aim to adapt the model's knowledge to excel in the new task. Transfer learning approaches like using a pre-trained model as an initialization and then fine-tuning for the target task fall under this category.

- **Knowledge Distillation**: Knowledge distillation involves training a smaller, more efficient model (student) to mimic the behavior of a larger, well-trained model (teacher). The teacher model's knowledge is transferred to the student model, which learns not just to replicate the teacher's outputs but also the reasoning behind those outputs. This technique helps create models that are computationally more efficient while retaining the knowledge captured by the larger model.

- **Multilingual and Cross-lingual Transfer**: Transformer models have been used to perform well in various languages. Training a model on multiple languages can lead to shared representations that aid in transferring knowledge across languages. Cross-lingual transfer techniques involve using knowledge learned from one language to improve performance in another.

- **Zero-shot and Few-shot Learning**: In zero-shot learning, models are trained on a particular task but can perform related tasks without additional training examples. Few-shot learning extends this concept to tasks with only a few examples available. Both techniques leverage the generalized knowledge captured during the initial training to adapt to new tasks with minimal data.

- **Other Techniques**: Other techniques, such as using auxiliary tasks during training, adapting to new domains using style transfer techniques, and leveraging external knowledge sources like ontologies and knowledge graphs, also contribute to effective knowledge transfer in transformer-based models.

2.6.3 Limitations of Existing Techniques

While knowledge transfer has proven effective, it also comes with some limitations:

- **Domain Mismatch**: The pre-trained model may not capture domain-specific nuances. Fine-tuning on a limited domain-specific dataset can lead to poor performance if there's a significant domain mismatch.
- **Catastrophic Forgetting**: Fine-tuning might cause the model to forget general knowledge learned during pre-training, leading to a loss of performance on other tasks.
- **Limited Control**: pretrained models are often treated as black boxes, making it challenging to control and direct the knowledge transfer process.
- **Lack of Diversity**: Limited data diversity in pre-training can restrict successful knowledge transfer to diverse domains and languages.
- **Overfitting**: Fine-tuning large models on small datasets can result in poor generalization.
- **Annotation Cost**: Some techniques require costly labelled data for target tasks, posing resource challenges.
- **Data Efficiency**: Some downstream tasks may require specialized or scarce data, which can limit the effectiveness of knowledge transfer.
- **Computational Efficiency**: Fine-tuning large models is computationally demanding, posing efficiency concerns.

2.6.4 Knowledge Transfer via Contextual Token Embeddings

Given these limitations, there is a strong need for exploring and testing new techniques for knowledge transfer. One such approach could be the utilizing of the ability of contextual token embeddings to transfer knowledge from domain-specific models to general knowledge models.

For instance, a model trained on domain-specific corpora (e.g., medical literature) will have learned nuanced semantic representations specific to that domain. By leveraging these embeddings, it is potentially possible to transfer domain-specific knowledge to a general knowledge model without the need for fine-tuning or other computational demanding techniques.

This approach offers several potential advantages:

- **Domain Adaptation**: Contextual embeddings can be fine-tuned on a small amount of in-domain data, making them adaptable to specific tasks while still retaining their domain-specific knowledge.
- **Efficiency and Speed**: Embeddings can be quickly integrated into existing models without the need for lengthy fine-tuning procedures.
- **Minimal Risk of Catastrophic Forgetting**: Transferring embeddings is less likely to lead to catastrophic forgetting, as it doesn't involve extensive re-training of the entire model.

However, this approach also presents challenges, such as:

- understanding the extent of knowledge that can be effectively transferred through embeddings;
- dealing with potential noise in the embeddings;
- evaluating the impact of transferred knowledge on downstream tasks.

These challenges are the main focus of this thesis.

2.7 Related Work

The thesis seeks to investigate the effectiveness of domain adaptation through knowledge transfer, specifically focusing on the syntactic and semantic context information of token embeddings while using limited computational resources and access to the training data. This section examines the previous approaches from the NLP research community for approaching this goal.

2.7.1 Vocabulary Reduction

Research papers from this category focus on reducing the size of language models by minimizing the vocabulary size or the number of tokens. The researchers aim to maintain or improve model performance while significantly compressing model size.

The paper ”Extreme Language Model Compression with Optimal Subwords and Shared Projections” [13] by Zhao et al. (2019) addresses the challenge of reducing the size of large language models like BERT for mobile and edge devices. They introduce a novel dual-training mechanism that simultaneously trains teacher and student models to achieve optimal word embeddings for the student's smaller vocabulary. This approach is coupled with shared projection matrices to transfer knowledge from teacher to student layers. Remarkably, their method compresses the BERT-base model by over 60x while maintaining task metrics. Their experiments reveal superior compression efficiency and accuracy compared to other techniques.

The subsequent paper ”Extremely Small BERT Models from Mixed-Vocabulary Training” [14] by Zhao et al. (2021) focuses on the memory footprint challenge of pre-trained language models like BERT on resource-limited devices. They introduce a novel knowledge distillation approach that aligns teacher and student embeddings through mixed-vocabulary training. This technique results in highly compressed BERT-large models with significantly smaller vocabularies and hidden dimensions, outperforming other distilled models in terms of size-accuracy trade-off on language understanding benchmarks and practical dialogue tasks. Their method's unique focus on student vocabulary size makes it easily combinable with various BERT distillation methods.

The paper ”Knowledge Distillation of Russian Language Models with Reduction of Vocabulary” [15] by Kolesnikova et al. (2022) proposes innovative strategies for efficient language models through knowledge distillation. They focus on reducing student model vocabulary and introduce Match and Reduce techniques to address the vocabulary mismatch challenge. Experimental results on Russian benchmarks demonstrate impressive compression rates (17× to 49×) while maintaining quality. Their approach offers potential refinements using contrastive and metric learning.

2.7.2 Vocabulary Adaptation

In the Vocabulary Adaptation category, the papers focus on adapting language models to specific domains by introducing domain-specific embeddings or modifying tokenization strategies. These approaches enhance the model’s performance in specialized domains without retraining it from scratch.

The paper ”Vocabulary Adaptation for Domain Adaptation in Neural Machine Translation” [16] by Sato et al. (2020) deals with the challenge of adapting neural machine translation (NMT) models to distant domains with differing vocabularies. They introduce a method called ”vocabulary adaptation” to fine-tune pre-trained NMT models effectively. This involves replacing embedding layers with domain-specific embeddings projected onto a source-domain space. The proposed approach yields significant improvements in fine-tuning performance.

The paper ”Efficient Domain Adaptation of Language Models via Adaptive Tokenization” [17] by Sachidananda et al. (2021) introduces an innovative method for adapting language models like BERT and RoBERTa to new domains. They propose adaptive tokenization, which efficiently identifies domain-specific sub-word sequences based on differences in conditional token distributions between source and domain-specific corpora. This approach achieves over 97% of the performance gains from domain-specific pretraining while minimizing model size, training time, and inference time compared to other techniques.

The paper ”Towards Simple and Efficient Task-Adaptive Pretraining for Text Classification” [18] by Ladkat et al. (2022) explores efficient techniques for task-adaptive pre-training (TAPT) in text classification. They investigate the impact of training only the embedding layer during TAPT and propose an approach to efficiently adapt BERT-based models. By updating the embedding layer and freezing the encoder layers, the model adapts to the target domain’s vocabulary while maintaining linguistic features. This approach maintains/improves performance, reduces training time, and cuts parameter count by 78% during TAPT. It offers an effective strategy for task adaptation and domain-specific tasks.

2.7.3 Vocabulary Transfer

The Vocabulary Transfer category involves transferring vocabulary during fine-tuning, aiming to improve model performance in specialized domains. Techniques in this category often involve using domain-specific tokenization for fine-tuning, resulting in better alignment between the model’s vocabulary and the target domain’s language.

The paper ”AVocaDo: Strategy for Adapting Vocabulary to Downstream Domain” [19] by Hong et al. (2021) introduces a strategy for adapting vocabulary to downstream domains in the context of fine-tuning during transfer learning. While traditional fine-tuning updates model parameters while keeping the pre-trained vocabulary unchanged, the authors propose considering the vocabulary as an optimizable parameter. This enables the expansion of the vocabulary with domain-specific terms based on tokenization statistics. To prevent overfitting, the embeddings of newly added words are preserved using knowledge from a pre-trained language model, enhanced with a regularization term. The AVocaDo strategy consistently enhances performance across diverse domains, including biomedical, computer science, news, and reviews.

The paper ”Fine-Tuning Transformers: Vocabulary Transfer” [20] by Samenko et al. (2021) investigates the impact of dataset-specific tokenization on transformer-based model fine-tuning. Through experiments, they show that this approach, coupled with appropriate initialization and fine-tuning strategies for vocabulary tokens, accelerates transfer learning and boosts model performance, termed ”vocabulary transfer”. The study covers tokenization, initialization, and fine-tuning, demonstrating benefits across datasets and techniques. They introduce an effective embedding initialization method “Vocabulary Initialization with Partial Inheritance” (VIPI).

The subsequent paper ”Vocabulary Transfer for Biomedical Texts” [21] by Mosin et al. (2022) explores the concept of vocabulary transfer as a transfer learning subtask in which language models fine-tune using domain-specific tokenization instead of the default one used during pretraining. While the previous paper focuses on general applicability and introduces vocabulary transfer as a concept, the current paper specifically applies and evaluates this concept in the context of medical text processing.

2.8 Summary

This chapter introduces transformer-based language models and their impact on NLP. Static and contextual word embeddings are explained, with emphasis on their role in representing word meanings. Cosine similarity measures the similarity between contextual embeddings. The importance of domain-specific vocabulary is highlighted, showing how it varies among models and domains. The concept of knowledge transfer is introduced, which involves using insights from one task to improve another. Techniques like domain adaptation, task adaptation, and knowledge distillation are explored for effective knowledge transfer. Limitations in current knowledge transfer methods are acknowledged and an approach involving contextual token embeddings for domain adaptation is suggested. Finally, the current body of work on vocabulary reduction, adaptation and transfer is presented.

Chapter 3

Preliminary Analysis

Before we start with the definition of the experiments, let us examine the models, which we will use later, more deeply. In particular, we are interested in the details of the vocabularies and their contextual properties. After that, we will have a closer look at the properties of token embeddings.

3.1 Model Overview

First of all, we want to introduce the Transformer-based models. As indicated in the previous chapter, we will look at the properties of the following models:

- **BERT**: ’bert-base-uncased’
- **BlueBERT**: ’bionlp/bluebert_pubmed_uncased_L-12_H-768_A-12’
- **PubMedBERT-abstract**: ’microsoft/BiomedNLP-PubMedBERT-base-uncased-abstract’
- **PubMedBERT-fulltext**: ’microsoft/BiomedNLP-PubMedBERT-base-uncased-abstract-fulltext’

All 4 models can be found on the HuggingFace platform and have the same architecture and vocabulary size of 30522 tokens, except for PubMedBERT-abstract, which has a slightly smaller vocabulary size of 28895 tokens.

3.2 Dataset overview

To perform the preliminary analysis of the models above and for some experiments later, we need small portions of the training datasets on which the above models have been trained. Therefore, we prepared from the following datasets:

- **wikitext-103-v1** [Source]: Dataset contains verified Good and Featured articles on Wikipedia from HuggingFace Hub. We first filtered out short lines ($\leq$ 20 words) in order to reduce noise and then randomly picked 250k lines.

- **PubMed abstracts** [Source]: PubMed contains citations and abstracts of biomedical literature from several NLM literature resources. We first downloaded all abstracts in XML format from the PubMed website, extracted the body of the abstracts in English and stored each abstract as a single line in a dataset. Finally, we randomly picked 250k lines for our thesis.

3.3 Analysis of Model Vocabularies

In this section, we want to analyze the vocabularies of the above models.

3.3.1 Size of models with reduced vocabulary

First, we investigate the relationship between the size of the vocabulary and the size of the model. Since all models have the same number of layers and the layers have the same amount of neurons, it is sufficient to investigate the relationship in BERT as an example.

In Figure 3.1 we can see how the total size of the model changes depending on the size of the vocabulary. If we reduce the vocabulary by 25%, the total size decreases by about 5.3%, at 50% the decrease is about 10.7% and at 75% it's even 16%.

3.3.2 Portion of Common Tokens

As all models use the same type of tokenizer, we can examine the portion of common tokens in their vocabularies. The results can be seen in Table 3.1 below.

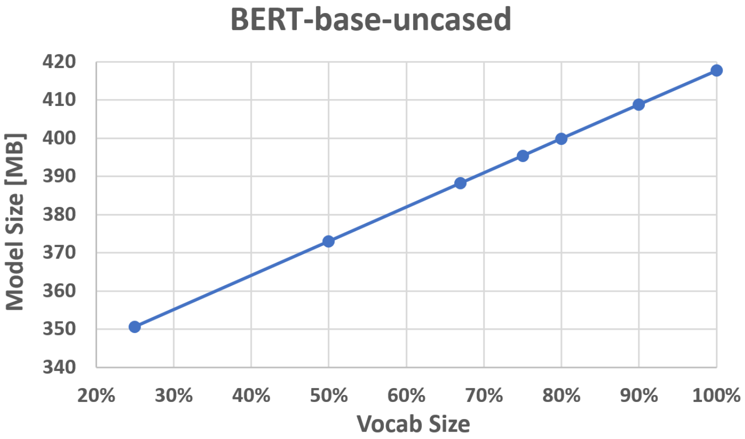
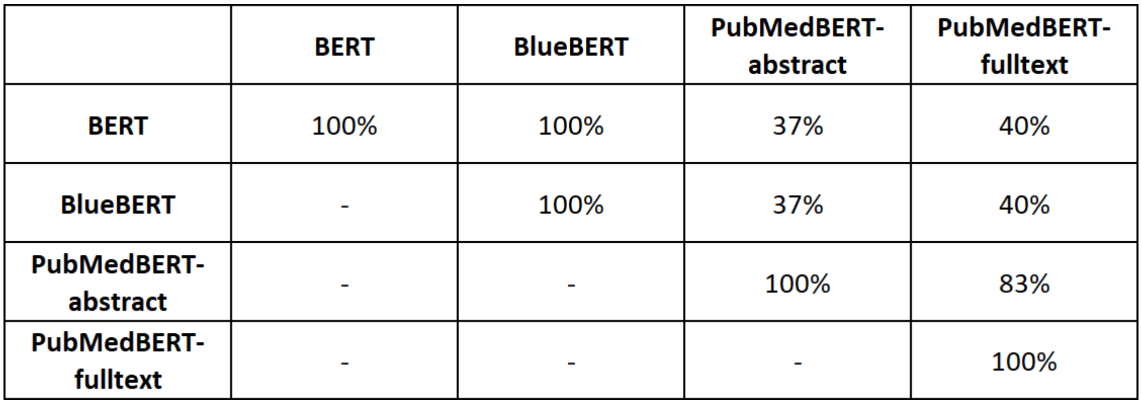


Figure 3.1: Model size vs. vocab size



	BERT	BlueBERT	PubMedBERT-abstract	PubMedBERT-fulltext
BERT	100%	100%	37%	40%
BlueBERT	-	100%	37%	40%
PubMedBERT-abstract	-	-	100%	83%
PubMedBERT-fulltext	-	-	-	100%

Table 3.1: Portion of common tokens

First, we can confirm that BERT and BlueBERT utilize identical vocabularies, and we also observe that although the PubMedBERT models were trained on the domain-specific text data only, they share approximately 40% of tokens with BERT. The two PubMedBERT models, despite being trained on the text data from the same domain, have only 83% common tokens, which may be because the abstracts provide a more dense representation of the knowledge due to the strict requirements on abstract length, therefore the publications themselves likely contain more general language, which can be confirmed by the larger portion of common tokens with BERT.

3.3.3 Length Distribution of Common Tokens

In Figure 3.2, we compare the length distribution of tokens in each vocabulary and their common tokens. For each pair of the models, we can visually split the length distribution into 3 broad categories: short tokens (1-2 characters long), mid-length tokens (3 to 11 characters long), and long tokens (12 characters and longer). Thereby, for all models, the number of short and long tokens is quite small compared to the mid-length category. The jump between lengths 2 and 3 comes from the fact that the bin with tokens of 3 characters contains not only short words or sub-words with 3 characters but also all single characters with ## at the beginning, as explained in Chapter ”Background”.

If we compare the length distribution of PubMedBERT-abstract and PubMedBERT-fulltext, we notice that the fulltext model contains a significantly larger amount of shorter tokens ($\leq$ 6 characters). In contrast, the abstract model contains longer tokens (¿ 8 characters), which might indicate that it has learned more domain-specific words.

3.3.4 Token Length Distribution

We can also plot the token distribution differently to check if there is a relationship between the position in the vocabulary and the token length. For this, we plot the ID of the token on the x-axis and the length of the respective token on the y-axis as seen in Figure 3.3.

We also added two reduction heuristics to the plots. The reduction by ID (”last”) is indicated by the vertical dotted lines and the reduction by length (”longest”) by the color of the respective token in the scatterplot, The heuristics themselves are introduced in the next chapter.

First, we notice many long tokens at the beginning of the BERT vocabulary, which are the random initialized placeholders with the form [unusedXXX] and can be replaced by domain-specific tokens if needed. Each model has a large portion of single-character tokens at the beginning of the vocabulary for protection against OOV issues but every model has its own tokens. Surprisingly, both PubMedBERT models share only half of the single-character tokens. This fact might lead to a problem during the knowledge transfer.

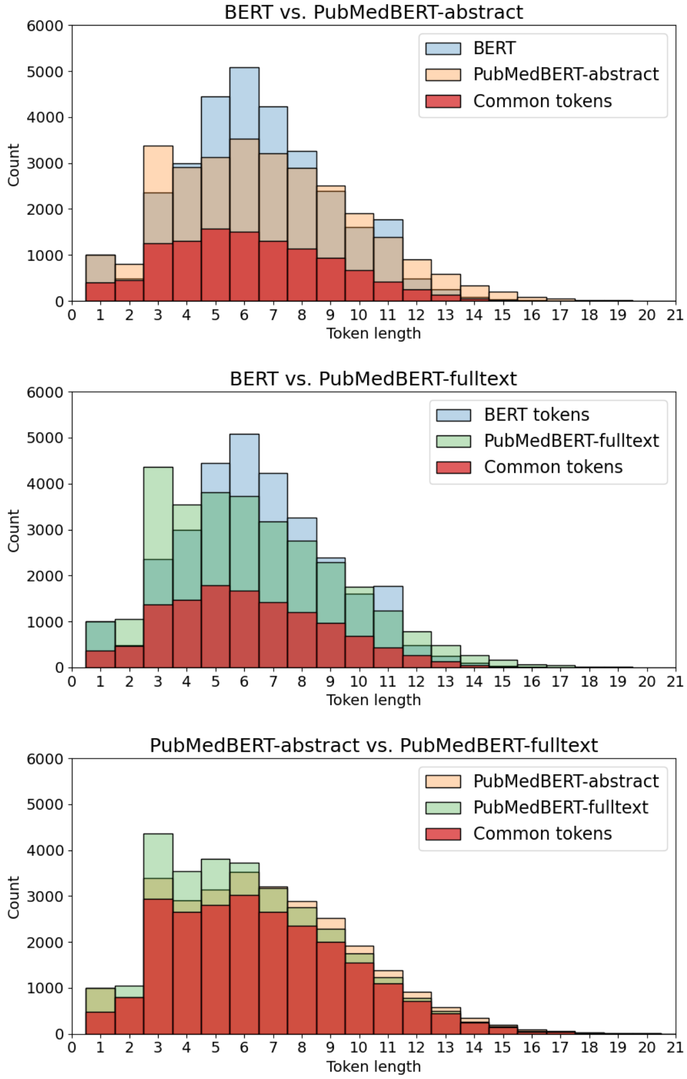


Figure 3.2: Token length distribution

In general, we observe that the length of the tokens is evenly distributed over the whole vocabulary in all models, but the PubMedBERT models have more larger tokens in the second half of the vocabulary than BERT.

3.3.5 Vocabulary in Context

Next, we want to analyze the behavior of the tokens in the context of general and domain-specific knowledge. We tokenize the prepared wikitext and PubMed abstract datasets with BERT, PubMedBERT-abstract and PubMedBERT-fulltext and plot the logarithm of the

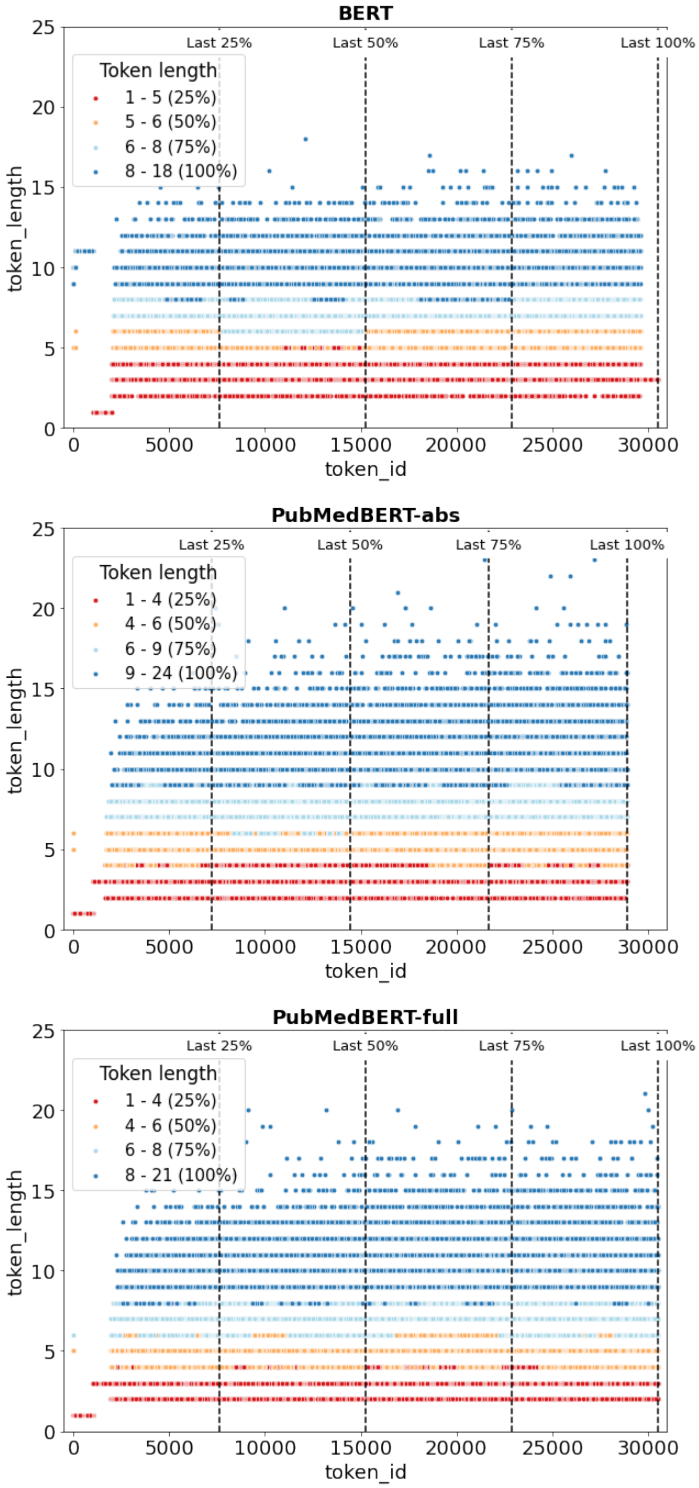


Figure 3.3: Token length vs token id

occurrence frequency of each token from their vocabularies in both tokenized datasets (see Figure 3.4). The title of each subplot also contains the total number of tokens produced by each model. Besides the mentioned components, the figures have the same elements as the plots from the previous section: ID on the x-axis, vertical lines and color coding for respective reduction heuristics.

We can clearly observe the correlation between the token's frequency and its ordinal number in vocabulary in case the domain of the training dataset and the tokenized dataset match. The frequency forms a wide band that steadily decreases as the IDs progress. It is noticeable that the band decreases more slowly and it spreads out towards the end of the vocabulary. This behavior is most prominent in the tokenization of wikitext by BERT.

When comparing the tokenization of PubMed abstracts by the two PubMedBERT models, it is also noticeable that the model that was pre-trained exclusively on the PubMed abstracts forms a much narrower frequency band than the other model, which additionally has seen a significant amount of general language from the medical publications during training.

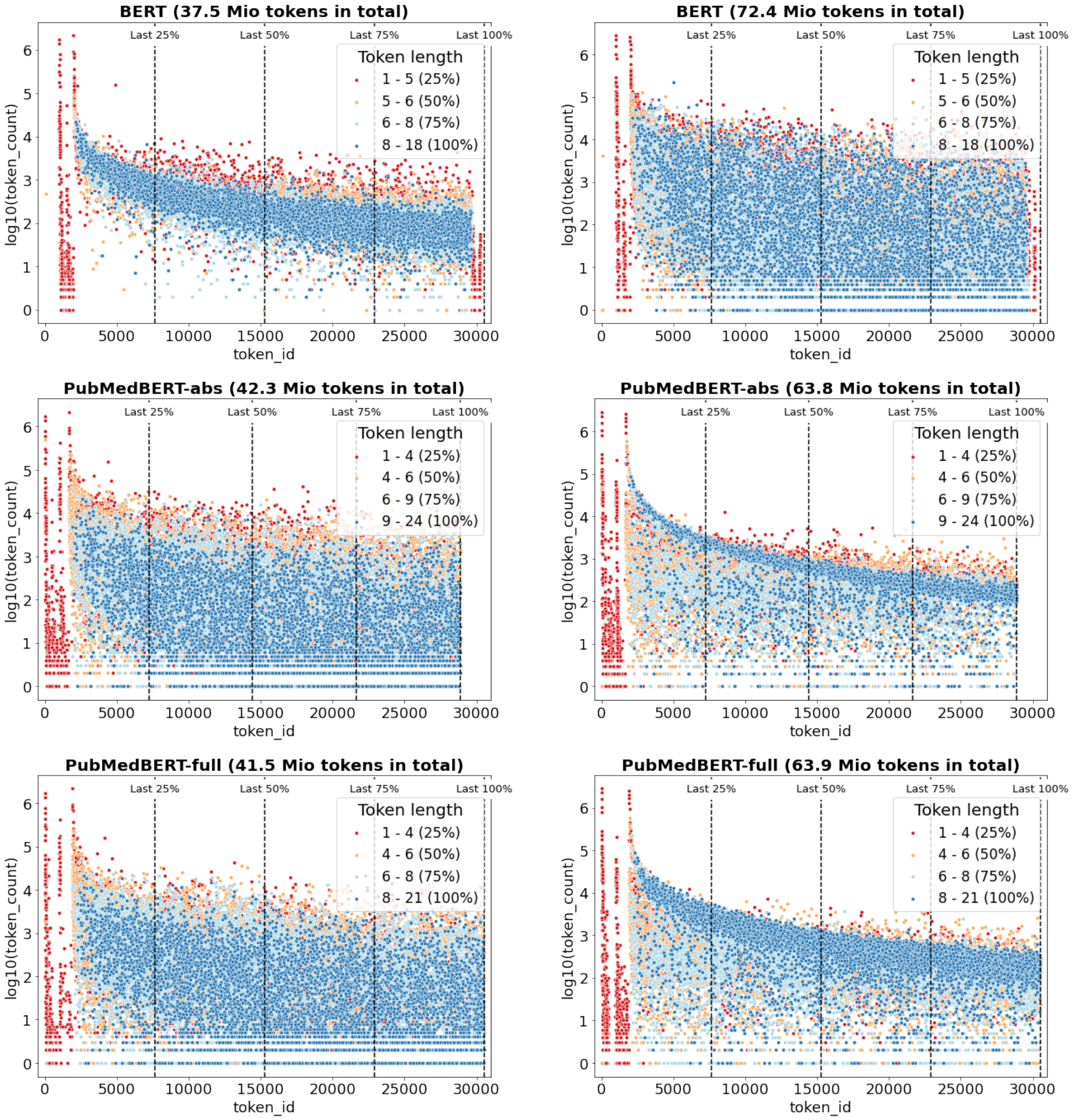
We conclude, that if the model was pre-trained on different domains than the tokenized dataset, no correlation between ID and frequency can be seen. This can be clearly observed in both cases: BERT vs. PubMed dataset as well as PubMedBERT models vs. wikitext.

The correlation is also emphasized by the number of tokens produced by the respective model. The more the model is adapted to the respective dataset, the lower the number of tokens the model outputs after tokenization. On the downside, the model may suffer from the overfitting.

3.4 Analysis of Contextual Word Embeddings

3.4.1 Visualization of Contextual Word Embeddings

To get a better understanding of high-dimensional data, it is always a good idea to visualize them in lower-dimensional space to get our minds around the data. In the case of the



(a) wikitext-103-v1

(b) PubMed abstracts

Figure 3.4: Contextual token count vs token id

contextual token embeddings of the transformer-based language models, there are two main goals:

1. Get an understanding of the token embedding distribution and how much of the context the token embeddings contain;

2. Compare different models' contextual token embeddings in terms of domain-specific knowledge after the first goal was achieved.

One of the appropriate techniques to achieve both goals is the t-SNE approach. It converts similarities between data points to joint probabilities and tries to minimize the Kullback-Leibler divergence between the joint probabilities of the low-dimensional embedding and the high-dimensional data.

We follow the tutorial by Kevin Gimpel [Source] to visualize BERT and PubMedBERT-full vocabularies and to analyze the contextual relationships between tokens within the vocabulary and between two models. For both models, we plot the first 4k tokens, which are longer than 3 tokens, are not special tokens and don't start with ##.

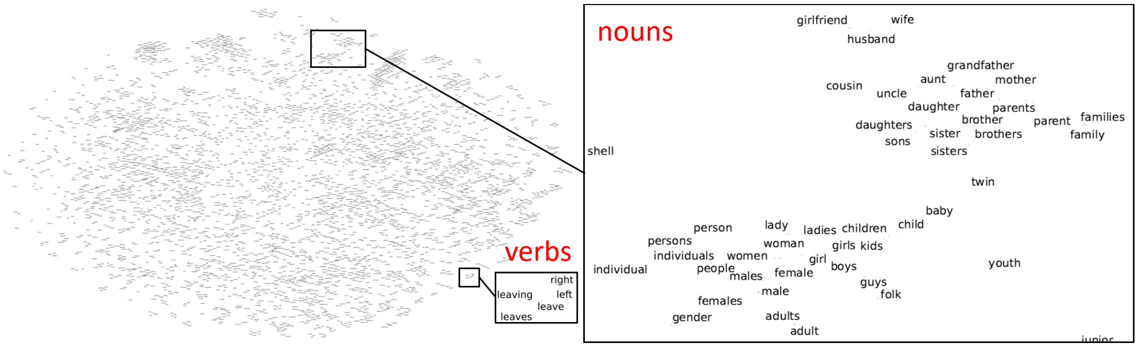


Figure 3.5: Cloud of Contextual Word Embeddings

Figure 3.5 shows the word cloud of BERT vocabulary. We can observe a large number of token clusters across the cloud, where two kinds of inner relationships immediately catch our attention: syntactic (declension for nouns and conjugation for verbs) and semantic (synonyms and antonyms). Within the cluster, the syntactically related tokens appear closer to each other, possibly indicating the higher importance of the syntactical relationship over the semantic one in the transformer-based language models.

We also see that the clusters with closer related word groups tend to appear closer to each other. In the figure, we show two clusters that are semantically very closely related and are located close to each other. On the right side of the figure, the tokens in the lower left cluster are terms used to describe different people within society in general, their sex, gender, and age. The tokens in the upper right cluster are terms for individuals within the family and their relationships to each other.

Therefore, we conclude that the token embeddings within transformer-based language models have acquired a substantial degree of contextualized knowledge about the lexical relationships through pre-training. It means that the model already has a lot of contextual knowledge about every token in a document and feeds that information into the model layers during inference time.

3.4.2 Visualization of Domain-specific Knowledge within Contextual Word Embeddings

Moving on, we want to investigate the domain-specific knowledge contextualization within general and domain-specific models. Since PubMedBERT models were pre-trained on the biomedical corpus, we focus on the contextual neighbourhood of the token ”cancer” within both vocabularies.

The upper word cloud in Figure 3.6 illustrates the contextual environment of the token in the BERT vocabulary. We notice that a cluster of medical terms evolves around the examined token, indicating that the model has learnt the semantically correct representation of the token. However, since the model has been pre-trained on a wide range of topics on the web, the medical knowledge of the model is highly limited and thus the nearest clusters are already unrelated to medicine.

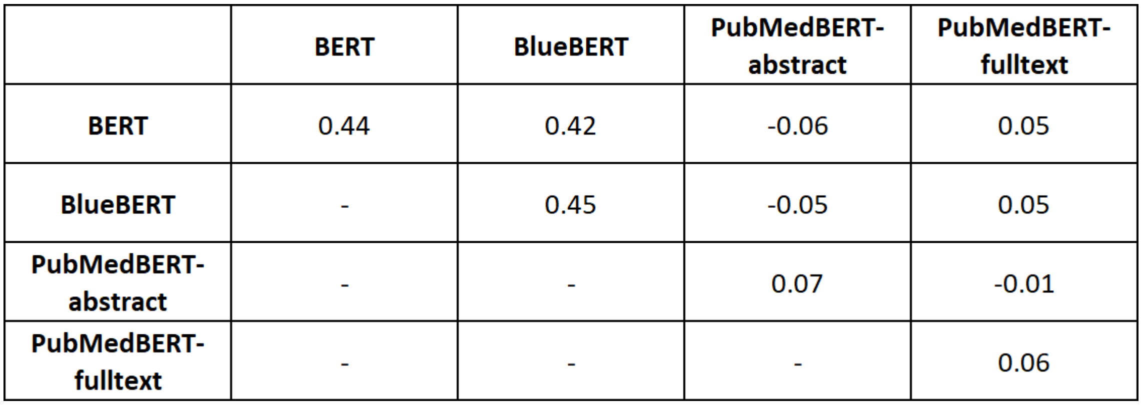
In the lower word cloud, we can see the contextual environment of the token in the PubMedBERT-fulltext vocabulary. There we observe that domain-specific pre-training has significantly expanded the medical vocabulary of the model. This not only allows medical terms to cluster together but even leads to the formation of clusters for individual branches of medicine and the token ”cancer” is correctly placed in the cluster related to oncology.

Based on the findings above, we conclude that both models have acquired semantic knowledge through their pre-training. But in contrast to domain-specific pre-trained PubMedBERT-fulltext, the biomedical vocabulary of BERT is highly limited by the wide range of topics in the pre-training corpus, leading to a sub-optimal performance on biomedical tasks.

3.4.3 Cosine Similarity of Contextual Word Embeddings

In this section, we investigate the assumption that cosine similarity captures the contextual meaning of words in the given context. Therefore, we compare the cosine similarity of token pairs with different lexical relationships among different models.

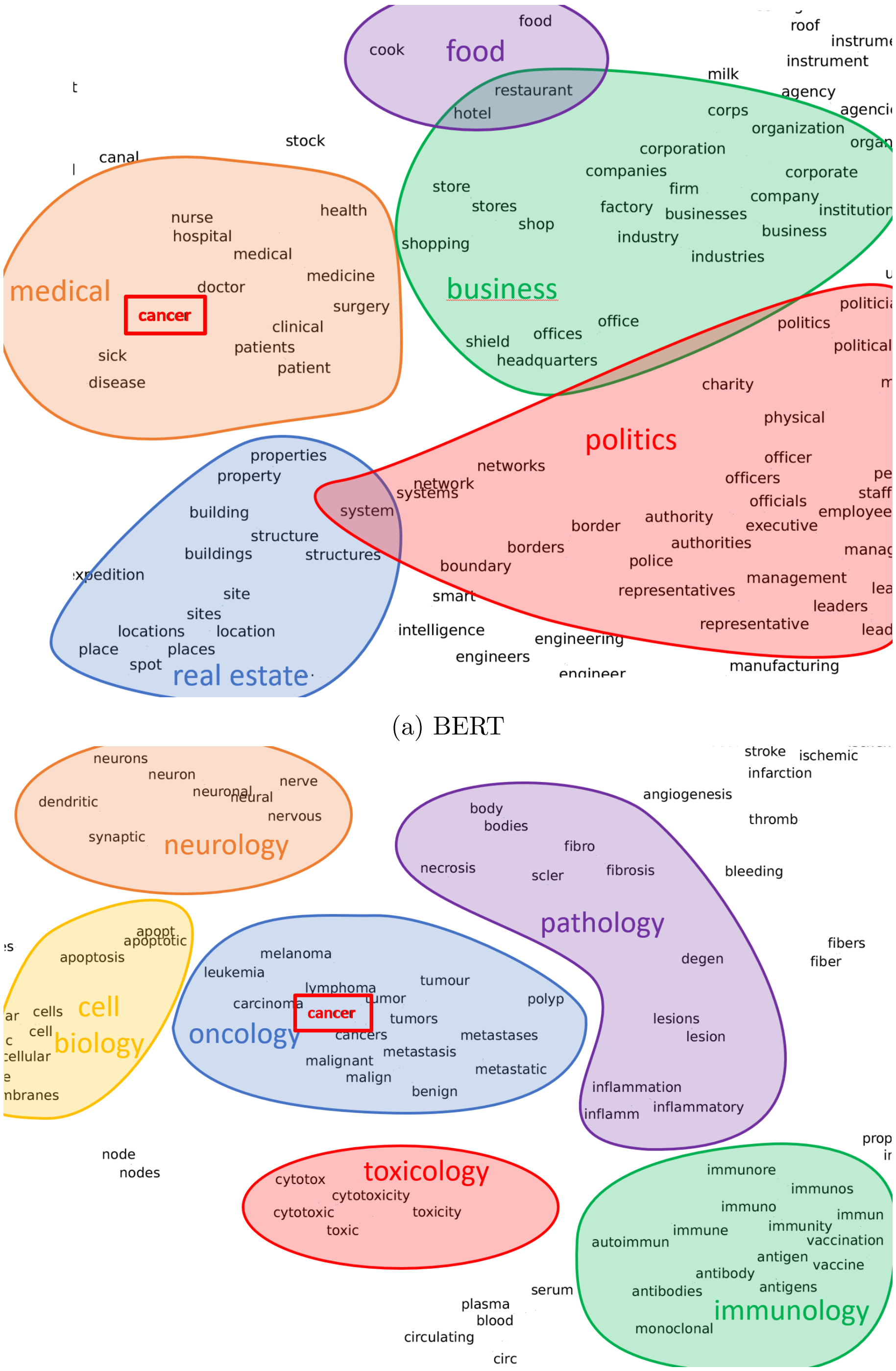
First, however, for every model, we have to determine the average pairwise cosine similarity value for all tokens within a vocabulary (see diagonal values in Table 3.2) and subtract the resulting value from the calculated similarity values of the individual token pairs to compare contextual proximity for lexical relationships among the models.



	BERT	**BlueBERT**	**PubMedBERT-abstract**	**PubMedBERT-fulltext**
BERT	0.44	0.42	-0.06	0.05
BlueBERT	-	0.45	-0.05	0.05
PubMedBERT-abstract	-	-	0.07	-0.01
PubMedBERT-fulltext	-	-	-	0.06

Table 3.2: Average pairwise cosine similarity of all token embeddings

From Table 3.2, we also note that the average intrinsic cosine similarity of BERT and BlueBERT is much higher than the similarity of PubMedBERT models. On the flip side, this implies that in PubMedBERT vocabularies, the token clusters with dissimilar contexts are much further apart in latent space than in BERT, and therefore easier to separate.



(a) BERT

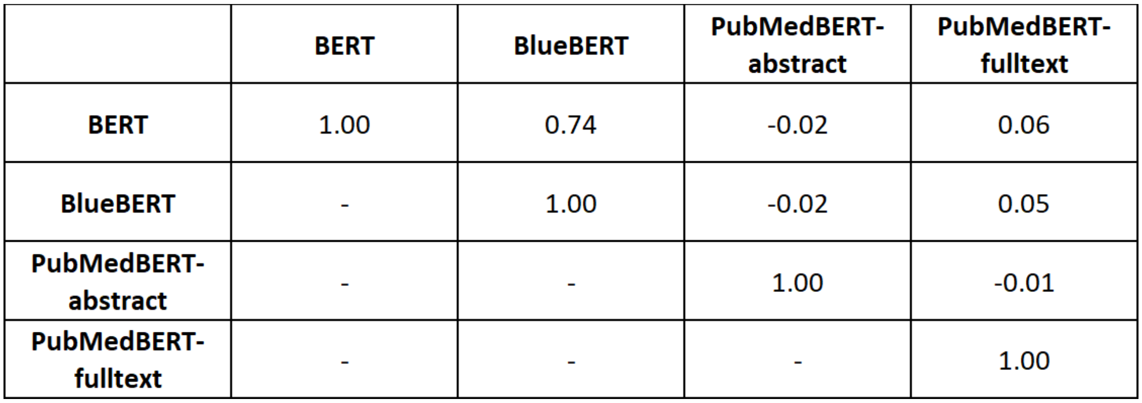
(b) PubMedBERT

Figure 3.6: Contextual neighborhood of token 'cancer'

We also calculate the average cosine similarity of all embeddings between two different models. We observe that the similarity between BERT and BlueBERT is almost identical to their intrinsic similarity, suggesting that, on average, the embeddings didn't change much during further training on biomedical data. However, the learned context of embeddings of both PubMedBERT models is entirely different from each other and also from the other models.

If we now compare the pairwise similarity of the common tokens shown in Table 3.3, we observe a significant increase in similarity between BERT and BlueBERT models, but no change in values between the other models. This may indicate that the BERT embeddings can be directly replaced by the BlueBERT embeddings and BERT could utilize them without further fine-tuning.

On the other hand, it would try to replace the BERT embeddings with the embeddings from PubMedBERT, it would lead to a catastrophic drop in performance since the embeddings are very much dissimilar.

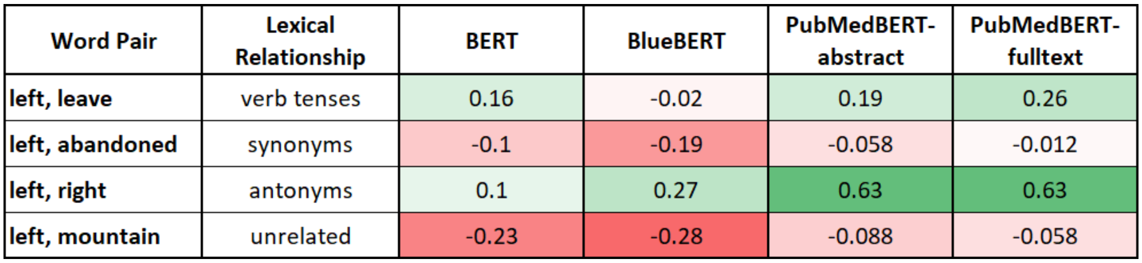


	BERT	BlueBERT	PubMedBERT-abstract	PubMedBERT-fulltext
BERT	1.00	0.74	-0.02	0.06
BlueBERT	-	1.00	-0.02	0.05
PubMedBERT-abstract	-	-	1.00	-0.01
PubMedBERT-fulltext	-	-	-	1.00

Table 3.3: Average cosine similarity of common token embeddings

Having determined the average similarity, we may now compute the contextual similarity of the word pairs. As an example, we examine four words with different lexical relationships to the word ”left”: verb tenses, synonyms, antonyms and unrelated words.

Table 3.4 shows that all four models have much higher values for verb tenses and antonyms than for synonyms and unrelated word pairs. This can be explained by the fact that the antonyms and tenses of the verb often have very similar neighbourhoods in the sentences and therefore this lexical rule is the easiest to learn.



Word Pair	Lexical Relationship	BERT	BlueBERT	PubMedBERT-abstract	PubMedBERT-fulltext
left, leave	verb tenses	0.16	-0.02	0.19	0.26
left, abandoned	synonyms	-0.1	-0.19	-0.058	-0.012
left, right	antonyms	0.1	0.27	0.63	0.63
left, mountain	unrelated	-0.23	-0.28	-0.088	-0.058

Table 3.4: Cosine similarity of lexical relationships

In the case of synonyms, it is probably more complicated by the fact that the words have different frequencies of occurrence in the training corpora, which makes this lexical rule more challenging to learn. As expected, the unrelated pair of words has the lowest contextual similarity.

We also note that the similarity scores of the antonyms are much higher in the case of PubMedBERT than for the BERT models. This could possibly help the PubMedBERT models achieve better performance in the classification tasks, as the task requires the separation of distinct classes.

3.5 Summary

To summarize the chapter, we presented a detailed examination of the transformer-based models to be used in the upcoming experiments, focusing on their vocabularies and contextual properties. We explored the size of reduced vocabularies, the proportion of common tokens, the length distributions of common tokens, and the behaviour of tokens within the models' vocabularies. We also examined the contextual word embeddings, visualizing their distribution and domain-specific knowledge. The chapter concludes with an exploration of cosine similarity among embeddings and how it captures contextual meaning. These insights lay the foundation for the upcoming experiments and shed light on the models' adaptability and knowledge representation.

Chapter 4

Methodology

In this chapter, we present the methodology for evaluating the effectiveness of token embeddings in transferring domain-specific knowledge between two pre-trained language models. The main research question is whether token embeddings can successfully transfer knowledge and thus enhance the performance of models on tasks from the target domain.

4.1 Overall Research Approach

To evaluate the main research question, we will measure the performance of all models designed during the experimental phase on tasks from the target domain. If transferred token embeddings lead to improved performance across various tasks, we consider that as an indication of successful knowledge transfer.

To be more specific, we want to work through the following steps one by one, while assessing the performance of the intermediate models on the target domain-specific benchmark:

1. **Preliminary steps:**

    - identify the **least** important tokens of the **source** model by iteratively reducing its vocabulary size;

    - identify the **most** important tokens of the **target** model by iteratively reducing its vocabulary size;
    - identify **common** tokens of both vocabularies;

2. **Vocabulary Distillation**: reduce the source model vocabulary by deleting the **least** important tokens with corresponding embeddings;

3. **Vocabulary Augmentation**: refill the vocabulary with the **most** important tokens from the target vocabulary to its initial size;

4. **Embedding Initialization**:

    - initialize source tokens with their original embeddings;
    - if augmented tokens are in the subset of common tokens, initialize their embeddings with the original embeddings as well;
    - initialize the embeddings of newly added target tokens with one of the knowledge transfer strategies.

5. **Final Evaluation**: assess the final model's performance on the domain-specific benchmark.

In the next sections of this chapter, we elaborate on each step of our research approach more deeply. We describe the techniques and algorithms we apply for vocabulary distillation, augmentation, and initialization. Additionally, we provide insights into the evaluation metrics utilized to measure the performance of the models and their transfer capabilities.

Our methodology considers the nuances of tokenization, vocabulary alignment, and statistical properties of the domain-specific corpora. By providing a detailed description of the approach, we aim to present a clear explanation of our research methods and the reasons for performing each step to assess the effectiveness of knowledge transfer through token embeddings.

4.2 Hypotheses

Based on the information about previous work on knowledge transfer in the field of transformer-based language models presented in Chapter 2 and the preliminary analysis of the vocabularies of the chosen models from Chapter 3, we propose the following set of hypotheses:

1. The performance of models on domain-specific tasks is proportional to the extent of adaptation of the model to the domain.
2. Different vocabulary reduction heuristics influence task performance differently based on the task type.
3. Knowledge transfer techniques can improve performance when transferring tokens and corresponding embeddings between models.
4. Combining vocabulary reduction and knowledge transfer approaches can lead to enhanced knowledge transfer capabilities and task performance.

To either confirm or reject the hypotheses, we propose experiments in the next sections. Later, we will validate the hypotheses, based on the results of individual experiments.

4.3 Baseline Models

As a baseline for evaluation of the knowledge transfer ability, we propose the following models:

1. **bert-base-uncased**: The model has a certain amount of biomedical domain knowledge from its pre-training and therefore, we can compare its performance to all designed models.
2. **naïve replacement**: The BERT tokens can be directly replaced with the vocabulary of PubMedBERT and the corresponding embeddings, without adapting them to the

BERT model in any way. This would allow us to examine the compatibility of the two models and provide the first estimation of the performance.

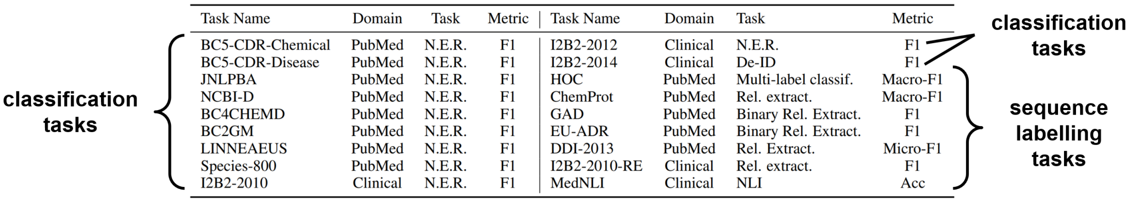
4.4 Experimental Setup

In this section, we outline the comprehensive experimental setup designed to evaluate the knowledge transfer capabilities of contextual token embeddings. We present experiments to be conducted in great detail, including the model configuration and the justification for each approach.

4.4.1 Performance Evaluation: Bio-LM benchmark

To assess the successful transfer of knowledge between two models, we need to compare the performance of the models resulting from different approaches. For this purpose, we have chosen the Bio-LM benchmark.

Lewis et al. introduced the biomedical benchmark ”Bio-LM” in their research paper ”Pretrained Language Models for Biomedical and Clinical Tasks: Understanding and Extending the State-of-the-Art” [22] which consists of 18 established biomedical (13) and clinical (5) NLP tasks. At the same time, the benchmark consists of 11 sequence labelling tasks and 7 classification tasks. We present the overview in Table 4.1 below.


classification
tasks
classification
tasks
sequence
labelling
tasks

Task Name	Domain	Task	Metric	Task Name	Domain	Task	Metric
BC5-CDR-Chemical	PubMed	N.E.R.	F1	I2B2-2012	Clinical	N.E.R.	F1
BC5-CDR-Disease	PubMed	N.E.R.	F1	I2B2-2014	Clinical	De-ID	F1
JNLPBA	PubMed	N.E.R.	F1	HOC	PubMed	Multi-label classif.	Macro-F1
NCBI-D	PubMed	N.E.R.	F1	ChemProt	PubMed	Rel. extract.	Macro-F1
BC4CHEMD	PubMed	N.E.R.	F1	GAD	PubMed	Binary Rel. Extract.	F1
BC2GM	PubMed	N.E.R.	F1	EU-ADR	PubMed	Binary Rel. Extract.	F1
LINNEAEUS	PubMed	N.E.R.	F1	DDI-2013	PubMed	Rel. Extract.	Micro-F1
Species-800	PubMed	N.E.R.	F1	I2B2-2010-RE	Clinical	Rel. extract.	F1
I2B2-2010	Clinical	N.E.R.	F1	MedNLI	Clinical	NLI	Acc

Table 4.1: Tasks from Bio-LM benchmark

They selected a broad range of datasets to cover both scientific and clinical textual domains, and common modelling tasks to optimize overlap with previous work in the space, drawing tasks from the BLUE benchmark (Peng et al., 2019), BioBERT (Lee et al., 2019), SciBERT (Beltagy et al., 2019) and ClinicalBERT (Alsentzer et al., 2019).

4.4.2 Off-the-shelf transformer-based language models

As the first experiment, we compared the performance of four off-the-shelf models on the Bio-LM benchmark: bert-base-uncased, BlueBERT-pubmed-uncased-L-12-H-768-A-12, PubMedBERT-abstract and PubMedBERT-fulltext. We aim to assess how the performance depends on the adaptation grade of vocabulary and token embeddings to the domain-specific language.

In all further experiments, we employ only two of the tested models as a foundation for all further designed models: bert-base-uncased (a.k.a. “BERT”) and PubMedBERT-base-uncased-abstract-fulltext (a.k.a. “PubMedBERT-fulltext”) since they both have the same vocabulary length and are located on the opposite ends of the domain adaptation spectrum.

4.4.3 Re-initialization of token embeddings

Next, we want to examine to what extent the model’s performance depends on the weights of the pretrained token embeddings and which type of distribution is more feasible for re-initialization once the token embeddings have been reset.

To achieve a higher degree of generality, we run the experiment with both BERT and PubMedBERT models and examine patterns in the performance changes. Based on that, we then try to derive insightful conclusions.

4.4.4 Vocabulary Reduction

To perform the first three steps of the designed approach for knowledge transfer we need to identify the least important tokens from the BERT model and the most important tokens from the PubMedBERT model. For this, we designed the next experiment to apply different reduction heuristics, gradually reducing the vocabulary sizes and evaluating the performance of the models.

The following vocabulary reduction heuristics were assessed:

1. **“last”**: delete X% of the tokens added to the vocabulary last, assuming that the

last added tokens are the least important for the pre-trained dataset, thus we expect the smallest drop in performance.

2. **"longest"**: delete X% of the longest tokens, assuming that the longest tokens can be easily represented by a combination of shorter tokens, thus a moderate drop in performance is to be expected.

3. **"freq"**: tokenize domain-specific corpus first and delete X% of the most infrequent tokens from the vocabulary. For this heuristic, We utilized 250k random abstracts from the PubMed abstracts dataset. In case the benchmark tasks have different statistical word distributions, it results in a significant drop in performance.

4. **"random"**: delete X% of random tokens, assuming that it would lead to the highest drop in performance since some important tokens would be deleted from the vocabulary as well.

To assess the performance drop for each of the reduction heuristics, the vocabulary was reduced gradually (reduction portions: 25%, 50%, 75%).

Note: all special tokens ([PAD], [UNK], ...) and tokens with lengths of fewer than 4 characters were always kept to avoid OOV issues.

4.4.5 Knowledge Transfer

To cover the next step of the knowledge transfer approach, we have to assess the knowledge transfer capabilities through the re-initialization of transferred tokens. Therefore, we designed four different transfer techniques that were evaluated on the Bio-LM benchmark:

1. **Naïve replacement (baseline model)**: Replace the vocab and token embeddings of the source model with the target's vocab and token embeddings, without re-initializing the embeddings at all.

2. **Replacement with random re-initialization**: Replace the source model vocab with the target's vocab and reset the token embeddings with random normal distribution.

3. **Syntactic knowledge transfer**:

    - For all common tokens use the original source token embeddings
    - For all the other target tokens, average the sub-token embeddings

    For example, since the token ‘nation’ is part of both vocabs, BERT and PubMedBERT, apply the token embedding from BERT:

    $E_{SYNTACTIC}('nation') = E_{BERT}('nation')$

    The token ’lymphoma’ is part of PubMedBERT’s vocab only, therefore new token embedding would be calculated by averaging the sub-token embeddings after tokenization by the BERT tokenizer, which contain 4 sub-tokens:
    $\{'l', '\#\#ym', '\#\#ph', '\#\#oma'\}$:

    $E_{SYNTACTIC}('lymphoma') = \frac{1}{4} \times (E_{BERT}('l') + E_{BERT}('\#\#ym') + E_{BERT}('\#\#ph') + E_{BERT}('\#\#oma'))$

4. **Semantic knowledge transfer**:

    - For all common tokens use the source token embeddings
    - For all the other target tokens, average the token embeddings of m semantical similar tokens (calculate cosine similarity) from the set of common tokens

    Again, since the token ‘nation’ is part of both vocabs, BERT and PubMedBERT, we just take the token embedding from BERT:

    $E_{SEMANTIC}('nation') = E_{BERT}('nation')$

    The token ’lymphoma’ is again part of PubMedBERT’s vocab only, therefore new token embedding would be calculated by averaging the token embeddings of m most semantically similar tokens from the set of common tokens:

$$cossim_{common}('lymphoma') = \{('nhl', 0.43), ('cancer', 0.34), ('tumor', 0.32)\}$$

$$E_{SEMANTIC}('lymphoma') = \frac{1}{3} \times (E_{BERT}('nhl') + E_{BERT}('cancer') + E_{BERT}('tumor'))$$

4.4.6 Combined Models

In the last experiment, we combine the best vocabulary reduction heuristic for the source and target models with the best knowledge transfer technique to refill the vocabulary to its original size with the most important tokens from the target vocabulary and re-initialize their token embeddings according to the best knowledge transfer strategy. After that, we evaluate the results on Bio-LM benchmark as usual.

4.5 Summary

The "Methodology" chapter outlines how token embeddings' effectiveness in transferring domain-specific knowledge between pre-trained language models is evaluated. It aims to answer whether this knowledge transfer improves task performance in a target domain. We introduce the overall approach for evaluation of knowledge transfer and present hypotheses covering adaptation level, vocabulary reduction's impact, knowledge transfer techniques, re-initialization effect, and combined approaches. To systematically validate the hypotheses, we designed a series of experiments, covering model selection, vocabulary reduction, and knowledge transfer strategies.

Chapter 5

Implementation

In this chapter, we reveal some important details about the implementation and execution of the experiments.

5.1 Experimental Environment

Implementation of our experiments involved utilizing PyTorch (Paszke et al., 2019 [23]) in combination with pre-trained models from the Huggingface Hub (Wolf et al., 2020 [24]).

All experiments were conducted on the computing cluster owned by the DATEXIS research group at the University of Applied Sciences of Berlin, in alignment with their ongoing research efforts. The stages of the experiments were implemented using Python along with associated frameworks and libraries, and Kubernetes jobs were employed for deployment on the cluster. To enable model execution on the Kubernetes cluster, we generated Docker images and stored them in the DATEXIS registry. Additionally, we formulated Kubernetes Jobs configuration files to allocate necessary resources and trigger container deployment. Throughout the experiments, NVIDIA A100 and V100 GPUs were employed based on availability, with the lowest priority for these resources.

5.2 Evaluation Pipeline

To evaluate all experiments, the following pipeline was defined, implemented and executed:

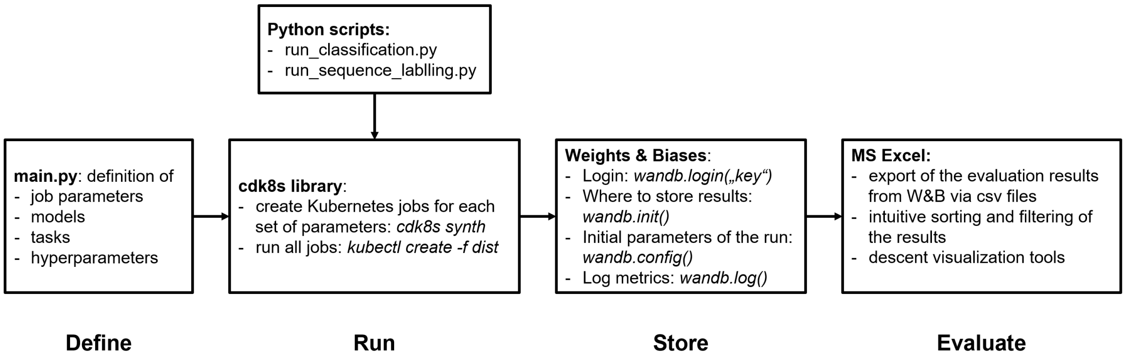

Python scripts:
- run_classification.py
- run_sequence_lablling.py
main.py: definition of
- job parameters
- models
- tasks
- hyperparameters
cdk8s library:
- create Kubernetes jobs for each set of parameters: cdk8s synth
- run all jobs: kubectl create -f dist
Weights & Biases:
- Login: wandb.login(„key“)
- Where to store results: wandb.init()
- Initial parameters of the run: wandb.config()
- Log metrics: wandb.log()
MS Excel:
- export of the evaluation results from W&B via csv files
- intuitive sorting and filtering of the results
- descent visualization tools
Define
Run
Store
Evaluate

Figure 5.1: Evaluation pipeline

5.3 Hyperparameter Optimization (HPO)

The choice of the suitable hyperparameter set is crucial for getting consistent performance on the evaluation benchmark for all examined models. Especially, evaluation tasks with a small amount of data, like HOC, suffer a lot from an inappropriate choice of hyperparameters.

Therefore, before evaluating all different model modifications, we perform a hyperparameter optimization step on the off-the-shelf models to identify a hyperparameter set for the most stable performance. The following hyperparameter options were examined:

Hyperparameter	Search Space
seed	{142, 193, 389, 793, 922}
batch size	{1, 2, 4, 8, 16}
learning rate	{1e-3, 1e-4, 1e-5, 2e-5, 3e-5, 4e-5, 5e-5, 6e-5, 7e-5, 8e-5, 9e-5, 1e-6}
max sequence length	{128, 256, 512}
train epochs	{3, 5, 10, 15, 20, 30}

Table 5.1: HPO search space

Chapter 6

Evaluation and Discussion

In this chapter, we present the results of all conducted experiments on knowledge transfer, analyze them quantitatively as well as qualitatively, and finally discuss the implications, limitations, and potential drawbacks of our findings.

6.1 Set of Hyperparameters

During the course of the HPO, we noticed that some of the hyperparameters had a significant impact on the performance of the models. Especially the tasks ChemProt, DDI and HOC were affected.

In Figure 6.1 we show an example of the impact of seed on F1-score as a function of the number of training epochs. We observe that the performance fluctuations differ a lot from model to model. PubMedBERT-fulltext reaches its maximum performance after only 5 training epochs, while other models need at least 10 epochs. It sometimes even happens that the random seed is so unfortunately set that the model is unable to achieve its full capability at all, as can be seen in the case of the BERT model.

Another parameter critical to the performance of the models is the learning rate, which in many of the tasks had led to complete model breakdown if the value was too small or too large. Especially the BERT and the syntactic transfer augmented BERT suffered from this. Therefore, we examined this parameter very carefully and finely granulated to find a

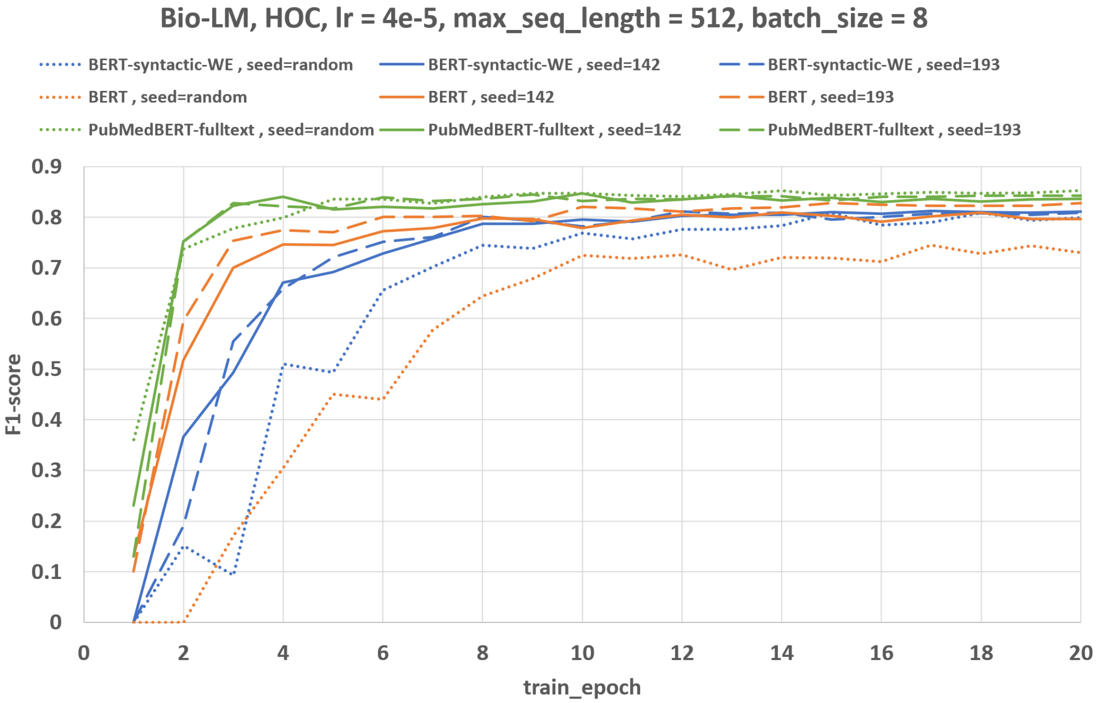


Figure 6.1: Seed evaluation on HOC

value at which all models could reach their maximum performance for a fair comparison.

Through evaluating the performance of all off-the-shelf models and some promising custom models with each possible hyperparameter combination via grid search, the following hyperparameter set was selected for further assessments:

Hyperparameter	**Final Value**
seed	142
batch size	8
learning rate	2e-5
max sequence length	512
train epochs	10

Table 6.1: Final set of hyperparameters

6.2 Experimental Results and Quantitive Error Analysis

In this section, we provide the results of all experiments, compare the performance of the custom models with the original ones (BERT and PubMedBERT-fulltext) and investigate the reason for the obtained performance.

Note: To enhance the readability, the best-performing model results in each task are highlighted in **bold** (as common in scientific publications), and the results of the second best-performing model are underlined.

6.2.1 Off-the-shelf transformer-based language models

Table 6.2 shows the results of the first experiment. All metrics in the table have absolute values and are calculated according to each task (see Table 4.1).

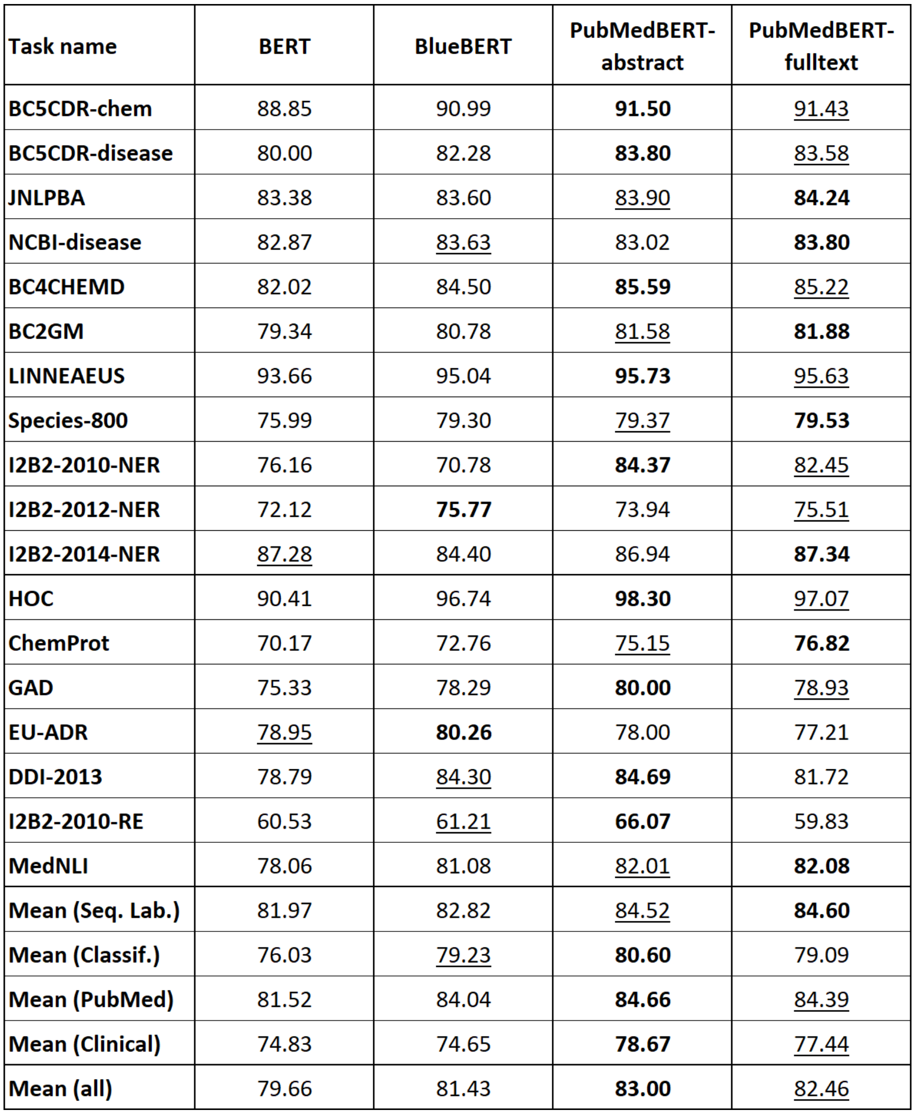
From the results, we can first observe that BlueBERT, which utilizes the same vocabulary as BERT and employs BERT as the starting model but was then further trained on the biomedical text data, performs significantly better on all Bio-LM tasks (except for two medical NER tasks I2B2-2010-NER and I2B2-2014-NER).

Comparing the performance of BlueBERT and the two PubMedBERT models, we observe that replacing the general BERT vocabulary with the domain-specific vocabulary brings another significant improvement in performance on most tasks (except for one classification task EU-ADR).

Comparing the performance between the two PubMedBERT models, they are on par. The only exceptions are two classification tasks (DDI-2013 and I2B2-2010-RE), where the model trained on PubMed abstracts only performs significantly better than the model trained additionally on PubMed publications.

Therefore, based on the results of the first experiment, we confirm the first hypothesis that the performance measured on the domain-specific benchmark is proportional to the degree of adaptation of the model to this domain.

Please note that based on this experiment, we cannot make any quantitative statement about the kind of proportionality of the performance to the degree of adaptation. For this, further investigations are needed, the results of which will be presented and evaluated in the following sections.



Task name	BERT	BlueBERT	PubMedBERT-abstract	PubMedBERT-fulltext
BC5CDR-chem	88.85	90.99	**91.50**	91.43
BC5CDR-disease	80.00	82.28	**83.80**	83.58
JNLPBA	83.38	83.60	83.90	**84.24**
NCBI-disease	82.87	83.63	83.02	**83.80**
BC4CHEMD	82.02	84.50	**85.59**	85.22
BC2GM	79.34	80.78	81.58	**81.88**
LINNEAEUS	93.66	95.04	**95.73**	95.63
Species-800	75.99	79.30	79.37	**79.53**
I2B2-2010-NER	76.16	70.78	**84.37**	82.45
I2B2-2012-NER	72.12	**75.77**	73.94	75.51
I2B2-2014-NER	87.28	84.40	86.94	**87.34**
HOC	90.41	96.74	**98.30**	97.07
ChemProt	70.17	72.76	75.15	**76.82**
GAD	75.33	78.29	**80.00**	78.93
EU-ADR	78.95	**80.26**	78.00	77.21
DDI-2013	78.79	84.30	**84.69**	81.72
I2B2-2010-RE	60.53	61.21	**66.07**	59.83
MedNLI	78.06	81.08	82.01	**82.08**
Mean (Seq. Lab.)	81.97	82.82	84.52	**84.60**
Mean (Classif.)	76.03	79.23	**80.60**	79.09
Mean (PubMed)	81.52	84.04	**84.66**	84.39
Mean (Clinical)	74.83	74.65	**78.67**	77.44
Mean (all)	79.66	81.43	**83.00**	82.46

Table 6.2: Evaluation results, off-the-shelf transformer-based language models

6.2.2 Pre-initialization of Token Embeddings

In the next experiment, we want to determine the extent to which the performance of the model depends on the weights of the pre-trained token embeddings and what kind of distribution is more suitable for their re-initialization.

To do so, we evaluate the relative change in performance with respect to the base model.

The change in performance is also highlighted in color to provide a better comparison between the tasks, with a relative drop in a performance highlighted in red and a relative increase highlighted in green. The brightness of the colors indicates the level of deviation from the performance of the base model in the respective task. The cells with the largest deviations in the entire experiment have the highest levels of intensity.

Looking at the results in Table 6.3, it is immediately noticeable that resetting the values of the token embedding weights negatively effects the results on all tasks except EU-ADR, where it helps all models improve their performance, except for the re-initialization of the weights with a normal distribution of BERT model, where it causes a slight decrease in performance.

Task name	BERT (source)	source vocab, random WE, uniform distr.	source vocab, random WE, normal distr.	PubMedBERT-fulltext (source)	source vocab, random WE, uniform distr.	source vocab, random WE, normal distr.
BC5CDR-chem	88.85	-29.72	-10.07	91.43	-31.62	-18.28
BC5CDR-disease	80.00	-29.08	-12.98	83.58	-32.14	-25.99
JNLPBA	83.38	-40.64	-2.62	84.24	-39.54	-4.33
NCBI-disease	82.87	-38.57	-10.24	83.80	-39.71	-11.27
BC4CHEMD	82.02	-37.62	-6.31	85.22	-39.52	-10.92
BC2GM	79.34	-39.14	-8.42	81.88	-41.13	-13.34
LINNEAEUS	93.66	-41.86	-24.40	95.63	-43.19	-32.40
Species-800	75.99	-52.07	-11.43	79.53	-59.55	-20.74
I2B2-2010-NER	76.16	-42.96	-13.88	82.45	-48.25	-29.52
I2B2-2012-NER	72.12	-34.46	-8.85	75.51	-41.96	-22.13
I2B2-2014-NER	87.28	-27.74	-6.30	87.34	-27.52	-5.14
HOC	90.41	-43.65	-57.65	97.07	-64.57	-57.71
ChemProt	70.17	-49.77	-29.15	76.82	-69.66	-56.10
GAD	75.33	-11.40	-7.20	78.93	-12.19	-14.92
EU-ADR	78.95	**0.82**	-1.12	77.21	**1.73**	1.42
DDI-2013	78.79	-70.62	-21.01	81.72	-53.04	-57.46
I2B2-2010-RE	60.53	-23.53	-14.69	59.83	-42.14	-33.26
MedNLI	78.06	-25.38	-15.05	82.08	-33.12	-15.20
Mean (Seq. Lab.)	81.97	-37.63	-10.50	84.60	-40.38	-17.64
Mean (Classif.)	76.03	-31.93	-20.84	79.09	-39.00	-33.32
Mean (PubMed)	81.52	-37.18	-15.59	84.39	-40.32	-24.77
Mean (Clinical)	74.83	-30.81	-11.76	77.44	-38.60	-21.05
Mean (all)	79.66	-35.41	-14.52	82.46	-39.84	-23.74

Table 6.3: Evaluation results, Pre-initialization of token embeddings

When we compare the performance drop between the two models at the same distribution type, it is noticeable that the domain-specific PubMedBERT model performs worse overall than the more general BERT model after resetting the weights. At this point, we hypothesize that the more flexible BERT model can adapt faster to the text data in the respective tasks when fine-tuned than the highly specialized PubMedBERT model.

If we consider the change in performance by re-initializing the embeddings with different distributions, it becomes quite clear that the normal distribution achieves a much better overall result (except for a few tasks) on the two base models, suggesting that normal distribution is an overall better strategy when re-initializing the embeddings.

6.2.3 Vocabulary Reduction

In this experiment, we evaluated different vocabulary reduction heuristics for both models, BERT and PubMedBERT, comparing the relative performance change of the models with the base models.

Considering the results of vocabulary reduction applied to BERT in Table 6.4, we notice that there are many tasks that don't suffer any performance drop as a result of vocabulary reduction, or rather, some tasks even benefit from it! For example, for the medical sequence labelling tasks I2B2-2010 and I2B2-2012, performance improved significantly when removing 50% of both the most recently added tokens (”last”) or the longest tokens (”longest”).

In general, we could argue that a reduction of the vocabulary with ”last” heuristics has a positive effect on solving classification tasks, except for HOC, where any change in vocabulary results in a significant degradation of performance.

We could draw the same conclusions when evaluating the results of downsizing the PubMedBERT vocabulary in Table 6.5. In addition, the vocabulary can be compressed very well with the heuristic ”freq” as well. We could explain this by the fact that the PubMedBERT model was also pre-trained on biomedical text data, on which the BioLM benchmark is also based, and thus the tokens added first to the vocabulary also occur very

reduction heuristic -->		**last**			**longest**			**freq**			**random**		
reduction portion -->		**25%**	**50%**	**75%**	**25%**	**50%**	**75%**	**25%**	**50%**	**75%**	**25%**	**50%**	**75%**
Task name	**BERT**												
BC5CDR-chem	**88.85**	-0.21	-1.16	-1.72	-0.54	-1.13	-2.40	-0.30	-0.23	-0.35	-0.76	-2.18	-3.07
BC5CDR-disease	80.00	-0.25	-1.70	-3.60	-0.90	-1.68	-4.46	0.12	**0.80**	0.07	-1.53	-2.34	-6.10
JNLPBA	**83.38**	-0.47	-0.14	-0.34	-0.30	-0.35	-0.68	-0.20	-0.24	-0.13	-0.28	-0.64	-0.81
NCBI-disease	82.87	-0.10	-0.76	-2.06	-0.32	-0.40	-1.75	0.71	-0.02	**1.36**	-0.94	-3.40	-3.08
BC4CHEMD	**82.02**	-0.34	-0.72	-1.35	-0.50	-0.95	-1.60	-0.05	-0.12	-0.25	-0.35	-1.24	-2.22
BC2GM	79.34	-0.56	-0.64	-1.90	-0.84	-1.51	-1.77	-0.08	**0.06**	-0.09	-1.35	-2.13	-3.75
LINNEAEUS	93.66	0.41	-2.51	-6.07	0.81	0.19	-6.49	**1.06**	0.86	-1.85	-5.28	-2.11	-9.14
Species-800	**75.99**	-1.03	-3.01	-2.76	-1.85	-3.17	-5.92	-1.85	-3.20	-1.82	-1.93	-2.24	-7.92
I2B2-2010-NER	76.16	**3.06**	1.55	-4.37	2.91	1.15	-0.87	1.77	-4.15	-3.12	-3.19	-3.00	-3.95
I2B2-2012-NER	72.12	-1.74	**4.64**	0.41	2.02	3.76	-2.10	0.14	0.56	1.54	0.67	-1.19	-5.21
I2B2-2014-NER	87.28	-2.96	-2.75	-2.81	-1.47	-1.93	-3.61	-1.39	-0.08	**1.54**	-1.06	-3.80	-1.15
HOC	**90.41**	-9.57	-0.09	-8.39	-7.53	-12.46	-29.96	-18.62	-7.51	-9.92	-10.41	-7.08	-24.52
ChemProt	70.17	**0.59**	-3.08	-4.31	-2.89	-9.52	-12.38	-2.37	-5.29	-6.90	-2.50	-8.90	-15.19
GAD	75.33	**0.83**	0.68	-3.24	-0.49	-3.80	-5.22	0.37	0.02	0.10	-3.04	-3.95	-3.27
EU-ADR	78.95	0.04	0.79	-0.71	0.43	**1.49**	-0.99	0.85	0.92	-0.09	-0.49	-1.25	-2.36
DDI-2013	78.79	**1.03**	0.73	-2.26	-3.78	-5.68	-6.24	-3.06	-2.26	-0.85	-5.50	-5.32	-11.92
I2B2-2010-RE	60.53	1.97	0.61	**2.20**	-5.74	-2.63	-4.59	-1.40	-0.62	-0.80	-3.13	-0.80	-5.50
MedNLI	78.06	-0.36	**0.14**	-1.79	-3.08	-1.65	-2.51	-0.86	-1.65	-2.29	-2.29	-3.08	-3.87
Mean (Seq. Lab.)	**81.97**	-0.38	-0.66	-2.42	-0.09	-0.55	-2.88	-0.01	-0.52	-0.28	-1.45	-2.21	-4.22
Mean (Classif.)	**76.03**	-0.78	-0.03	-2.64	-3.30	-4.89	-8.84	-3.58	-2.34	-2.96	-3.91	-4.34	-9.52
Mean (PubMed)	**81.52**	-0.74	-0.89	-2.98	-1.44	-3.00	-6.14	-1.80	-1.25	-1.59	-2.64	-3.29	-7.18
Mean (Clinical)	74.83	0.00	**0.84**	-1.27	-1.07	-0.26	-2.74	-0.35	-1.19	-0.63	-1.80	-2.37	-3.94
Mean (all)	**79.66**	-0.54	-0.41	-2.50	-1.34	-2.24	-5.20	-1.40	-1.23	-1.33	-2.41	-3.04	-6.28

Table 6.4: Evaluation results, Reduced models, BERT

frequently both in the training data set and in the benchmark.

Overall, since a 50% reduction of the vocabulary with "last" heuristic has a positive effect on solving classification tasks and medical tasks in both models and causes hardly any degradation in the other tasks, we will use it as the best reduction heuristic for the ongoing experiments.

6.2.4 Knowledge Transfer

In this experiment, we evaluate the capabilities of knowledge transfer from one model to another using tokens and the corresponding embeddings.

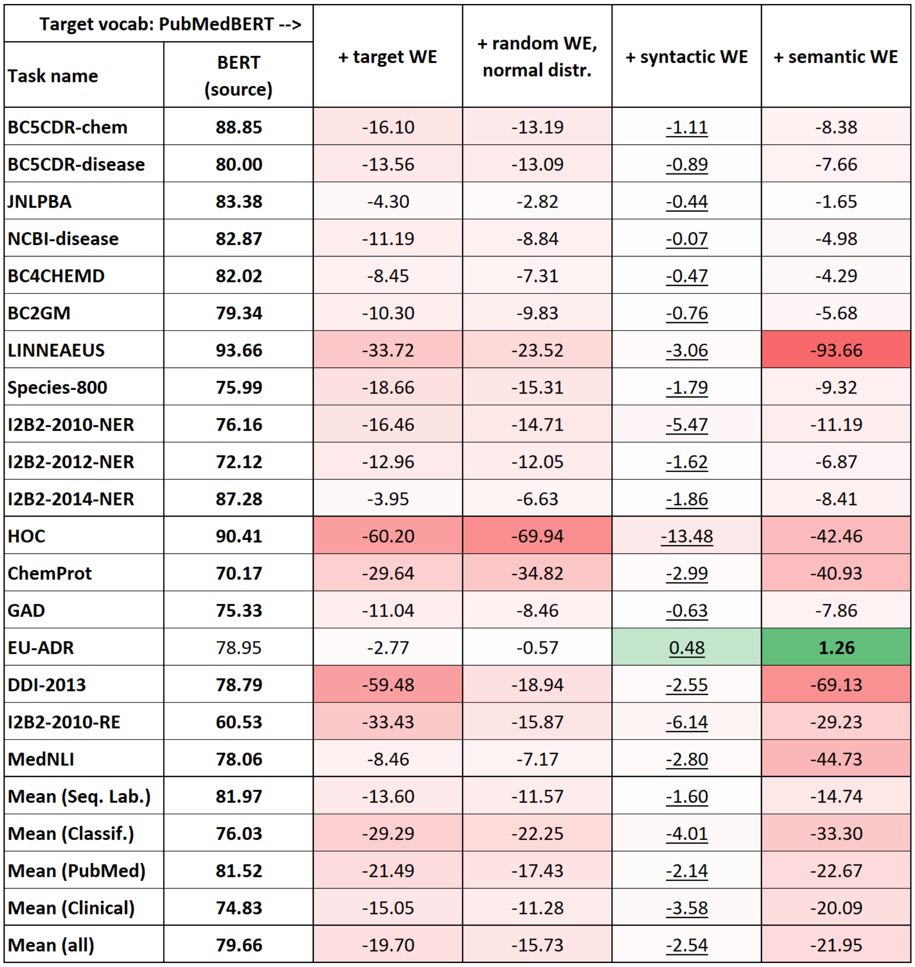
We observe that the naive replacement of the vocabulary of BERT with the tokens

reduction heuristic -->		last			longest			freq			random		
reduction portion -->		25%	50%	75%	25%	50%	75%	25%	50%	75%	25%	50%	75%
Task name	**PubMedBERT-fulltext**												
BC5CDR-chem	**91.43**	-0.52	-1.05	-3.44	-0.69	-2.35	-5.42	-0.24	-0.43	-1.12	-1.73	-2.59	-6.15
BC5CDR-disease	**83.58**	-1.01	-1.39	-7.39	-0.96	-4.88	-10.31	-0.28	-0.22	-1.83	-3.31	-6.70	-11.54
JNLPBA	84.24	-0.03	-0.19	-0.51	-0.26	-0.68	-1.45	-0.11	**0.06**	-0.14	-0.59	-0.83	-1.58
NCBI-disease	83.80	-0.44	-1.25	-2.21	-1.55	-2.79	-5.92	-0.81	-0.49	**0.39**	-1.52	-3.39	-5.57
BC4CHEMD	85.22	-0.27	-0.49	-2.11	-0.29	-1.47	-3.92	**0.10**	-0.29	-0.95	-0.55	-1.86	-4.08
BC2GM	**81.88**	-0.22	-1.37	-3.48	-0.99	-1.94	-5.55	-0.02	-0.14	-1.23	-1.38	-2.75	-5.39
LINNEAEUS	95.63	-0.40	-3.70	-9.02	**0.09**	-5.02	-14.59	-0.93	-0.62	-3.62	-4.24	-11.60	-14.90
Species-800	79.53	-0.44	-0.93	-5.25	**0.02**	-3.25	-7.19	-0.57	-1.62	-3.01	-1.20	-7.19	-10.74
I2B2-2010-NER	82.45	0.46	-0.68	-8.37	**1.69**	-6.16	-10.50	-6.31	-6.38	-6.49	-2.23	-4.37	-6.24
I2B2-2012-NER	75.51	-2.40	**2.32**	-1.79	0.39	-1.93	-6.27	-0.53	0.74	-0.31	-1.22	-3.02	-4.01
I2B2-2014-NER	87.34	-1.91	-0.75	-3.75	**0.71**	-1.55	-2.17	-1.09	-2.85	-1.17	0.21	-0.13	0.34
HOC	97.07	0.23	**1.71**	-1.86	0.17	-1.72	-59.04	0.01	1.02	-0.31	-0.01	-1.48	-20.01
ChemProt	**76.82**	-1.70	-0.81	-5.23	-3.03	-5.06	-21.02	-1.84	-1.47	-2.82	-3.70	-6.47	-16.76
GAD	78.93	0.34	0.19	-2.22	-0.76	-1.94	-5.47	**0.66**	-0.30	0.12	-1.07	-2.45	-6.72
EU-ADR	77.21	-0.08	**2.91**	1.30	-1.36	1.91	0.99	1.08	2.60	1.11	1.39	0.67	1.03
DDI-2013	81.72	**1.95**	0.94	-4.54	-0.19	-1.49	-12.90	1.03	1.44	1.07	-0.35	-2.62	-15.38
I2B2-2010-RE	59.83	1.04	2.67	-8.07	-2.09	2.24	-6.44	**4.52**	1.97	-1.85	2.10	-2.56	-4.12
MedNLI	82.08	0.29	-0.57	-3.80	-0.86	-3.37	-7.03	0.29	**0.57**	-2.44	-3.23	-3.73	-9.10
Mean (Seq. Lab.)	**84.60**	-0.65	-0.86	-4.30	-0.17	-2.91	-6.66	-0.98	-1.11	-1.77	-1.61	-4.04	-6.35
Mean (Classif.)	79.09	0.30	**1.00**	-3.49	-1.16	-1.35	-15.84	0.82	0.83	-0.73	-0.69	-2.66	-10.15
Mean (PubMed)	**84.39**	-0.20	-0.42	-3.54	-0.75	-2.36	-11.68	-0.15	-0.03	-0.95	-1.40	-3.79	-9.06
Mean (Clinical)	77.44	-0.50	**0.60**	-5.16	-0.03	-2.15	-6.48	-0.63	-1.19	-2.45	-0.87	-2.76	-4.63
Mean (all)	**82.46**	-0.28	-0.14	-3.99	-0.55	-2.30	-10.23	-0.28	-0.35	-1.37	-1.26	-3.50	-7.83

Table 6.5: Evaluation results, Reduced models, PubMedBERT

from PubMedBERT with corresponding embeddings has a significant negative impact on the performance of the model. This could be explained by the fact that while the tokens themselves might be a better fit for solving the benchmark, they are incompatible with the model weights from BERT due to different pre-training processes. To overcome this issue, the token embeddings would have to be adapted to the BERT model.

Therefore, we tested three different approaches and compiled the results in Table 6.6.



Target vocab: PubMedBERT -->		+ target WE	+ random WE, normal distr.	+ syntactic WE	+ semantic WE
Task name	BERT (source)				
BC5CDR-chem	88.85	-16.10	-13.19	-1.11	-8.38
BC5CDR-disease	80.00	-13.56	-13.09	-0.89	-7.66
JNLPBA	83.38	-4.30	-2.82	-0.44	-1.65
NCBI-disease	82.87	-11.19	-8.84	-0.07	-4.98
BC4CHEMD	82.02	-8.45	-7.31	-0.47	-4.29
BC2GM	79.34	-10.30	-9.83	-0.76	-5.68
LINNEAEUS	93.66	-33.72	-23.52	-3.06	-93.66
Species-800	75.99	-18.66	-15.31	-1.79	-9.32
I2B2-2010-NER	76.16	-16.46	-14.71	-5.47	-11.19
I2B2-2012-NER	72.12	-12.96	-12.05	-1.62	-6.87
I2B2-2014-NER	87.28	-3.95	-6.63	-1.86	-8.41
HOC	90.41	-60.20	-69.94	-13.48	-42.46
ChemProt	70.17	-29.64	-34.82	-2.99	-40.93
GAD	75.33	-11.04	-8.46	-0.63	-7.86
EU-ADR	78.95	-2.77	-0.57	0.48	**1.26**
DDI-2013	78.79	-59.48	-18.94	-2.55	-69.13
I2B2-2010-RE	60.53	-33.43	-15.87	-6.14	-29.23
MedNLI	78.06	-8.46	-7.17	-2.80	-44.73
Mean (Seq. Lab.)	81.97	-13.60	-11.57	-1.60	-14.74
Mean (Classif.)	76.03	-29.29	-22.25	-4.01	-33.30
Mean (PubMed)	81.52	-21.49	-17.43	-2.14	-22.67
Mean (Clinical)	74.83	-15.05	-11.28	-3.58	-20.09
Mean (all)	79.66	-19.70	-15.73	-2.54	-21.95

Table 6.6: Evaluation results, Knowledge transfer

First, we reset the token embeddings and replaced them with random weights with a normal distribution. For many tasks, this improves performance compared to simple replacement, but it remains significantly below the performance level of the initial model.

The other two strategies utilize averaging from the pre-trained embeddings, so practically they are leveraging knowledge from prior training of the models and thus theoretically should perform better than a model with the randomly pre-initialized embeddings.

This hypothesis is totally valid for syntactic transfer. Re-initialization of the token embeddings with the weights resulting from the averaging of the sub-tokens helps the model to correctly parse and process the new tokens. However, it is not enough to beat the base model for any BioLM task, except for the EU-ADR classification task.

However, the semantic transfer didn't perform well at all. While the model did better on the Sequence Labelling Tasks than the first two simpler transfer strategies, it ran very unstable, so no result could be obtained on LINNEAEUS. The model also performed significantly worse on the classification tasks than all other models in this experiment.

Thus, the model with syntactically transferred token embeddings is the best model from all evaluated models in this experiment regarding its performance on BioLM and we will compare it with the other best models from other experiments.

6.2.5 Combined models

In our final experiment, we want to evaluate whether the two approaches from the previous experiments can be combined and thus possibly reveal an observable knowledge transfer. In this experiment, both BERT and PubMedBERT are again examined.

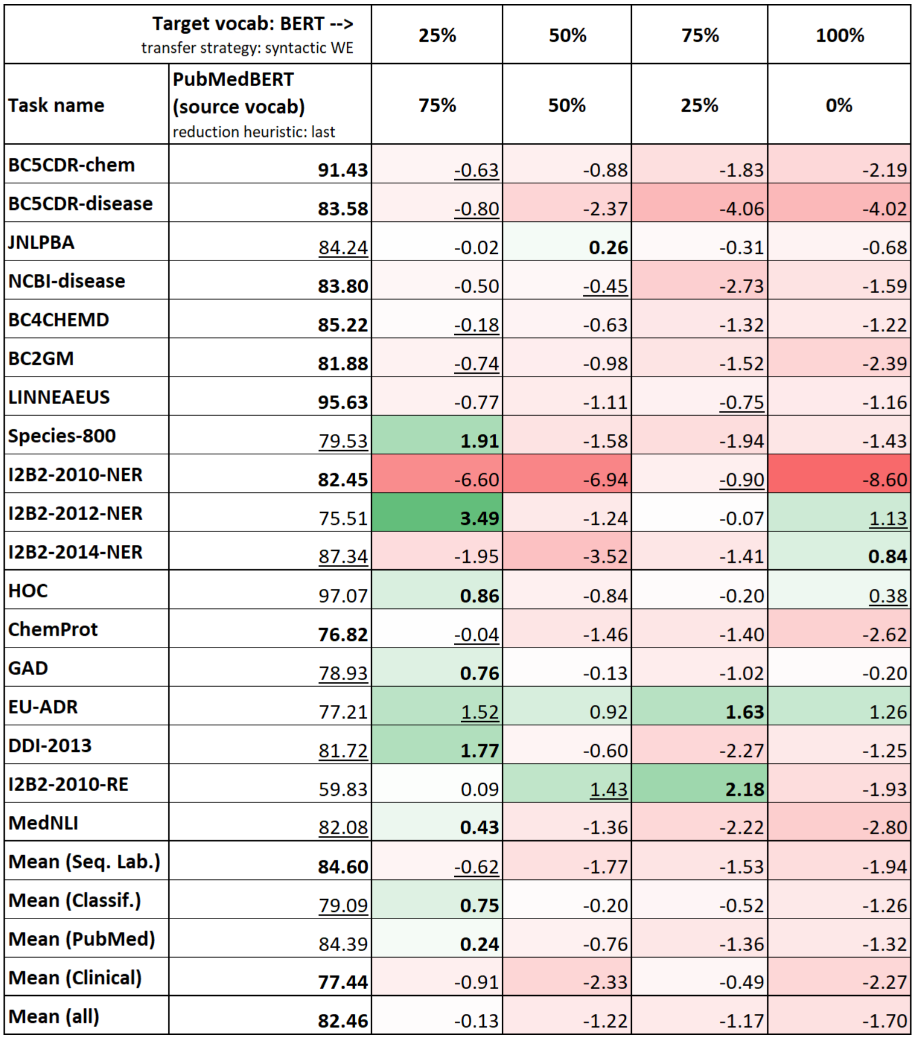
For this, the vocabulary of a source model is first reduced with the "last" heuristic, since it had shown to be the best of all the reduction heuristics. Afterwards, the vocabulary is augmented with the first tokens from the target model to its initial size. There are two restrictions for augmenting: firstly, no duplication of the tokens should occur and secondly, the tokens should be selected from the beginning of the target vocabulary. The embeddings of the newly added tokens are pre-initialized with the syntactic knowledge transfer approach, which means that each token is first split into sub-tokens by the tokenizer of the source model and then their embeddings are averaged.

Looking at the results on BERT first (see Table 6.7), we observe an improvement in performance on some of the tasks. Most notably, the medical classification I2B2 tasks seem to benefit from the approach, as well as NCBI-disease and HOC.

Target vocab: PubMedBERT --> transfer strategy: syntactic WE		**25%**	**50%**	**75%**	**100%**
Task name	**BERT (source vocab)** reduction heuristic: last	**75%**	**50%**	**25%**	**0%**
BC5CDR-chem	**88.85**	-1.25	-1.25	-1.02	-1.11
BC5CDR-disease	**80.00**	-1.02	-1.24	-1.53	-0.89
JNLPBA	**83.38**	-0.23	-0.53	-0.26	-0.44
NCBI-disease	82.87	-0.10	**1.15**	0.53	-0.07
BC4CHEMD	**82.02**	-0.18	-0.26	-0.31	-0.47
BC2GM	**79.34**	-1.12	-0.66	-0.46	-0.76
LINNEAEUS	**93.66**	-0.52	-0.02	-1.60	-3.06
Species-800	**75.99**	-2.94	-3.89	-4.38	-1.79
I2B2-2010-NER	76.16	-1.86	0.84	**4.07**	-5.47
I2B2-2012-NER	72.12	-0.12	**1.02**	-0.66	-1.62
I2B2-2014-NER	87.28	**0.60**	-0.18	-1.03	-1.86
HOC	90.41	0.90	**1.34**	0.89	-13.48
ChemProt	**70.17**	-1.91	-1.11	-0.87	-2.99
GAD	**75.33**	-0.01	-0.08	-0.68	-0.63
EU-ADR	78.95	-0.09	-0.21	-0.95	**0.48**
DDI-2013	78.79	-0.29	**0.35**	-0.31	-2.55
I2B2-2010-RE	**60.53**	-3.14	-4.82	-2.88	-6.14
MedNLI	78.06	**0.14**	-1.51	0.00	-2.80
Mean (Seq. Lab.)	**81.97**	-0.79	-0.46	-0.60	-1.60
Mean (Classif.)	**76.03**	-0.88	-1.23	-0.95	-2.44
Mean (PubMed)	**81.52**	-0.67	-0.49	-0.84	-2.14
Mean (Clinical)	**74.83**	-0.87	-0.93	-0.10	-3.58
Mean (all)	**79.66**	-0.83	-0.73	-0.73	-1.89

Table 6.7: Evaluation results, Combined models, BERT

Surprisingly, the reverse augmentation also yielded an improvement (see Table 6.8). Replacing the last added tokens from PubMedBERT vocabulary with the first added tokens from BERT vocabulary noticeably improved the performance of the model on many tasks. Most prominently, the classification tasks benefited from replacing the 25% last tokens of PubMedBERT with the first tokens from the BERT vocabulary. Also, the further replacement of the domain-specific tokens by the more common ones resulted in a further improvement of the performance on some classification tasks.



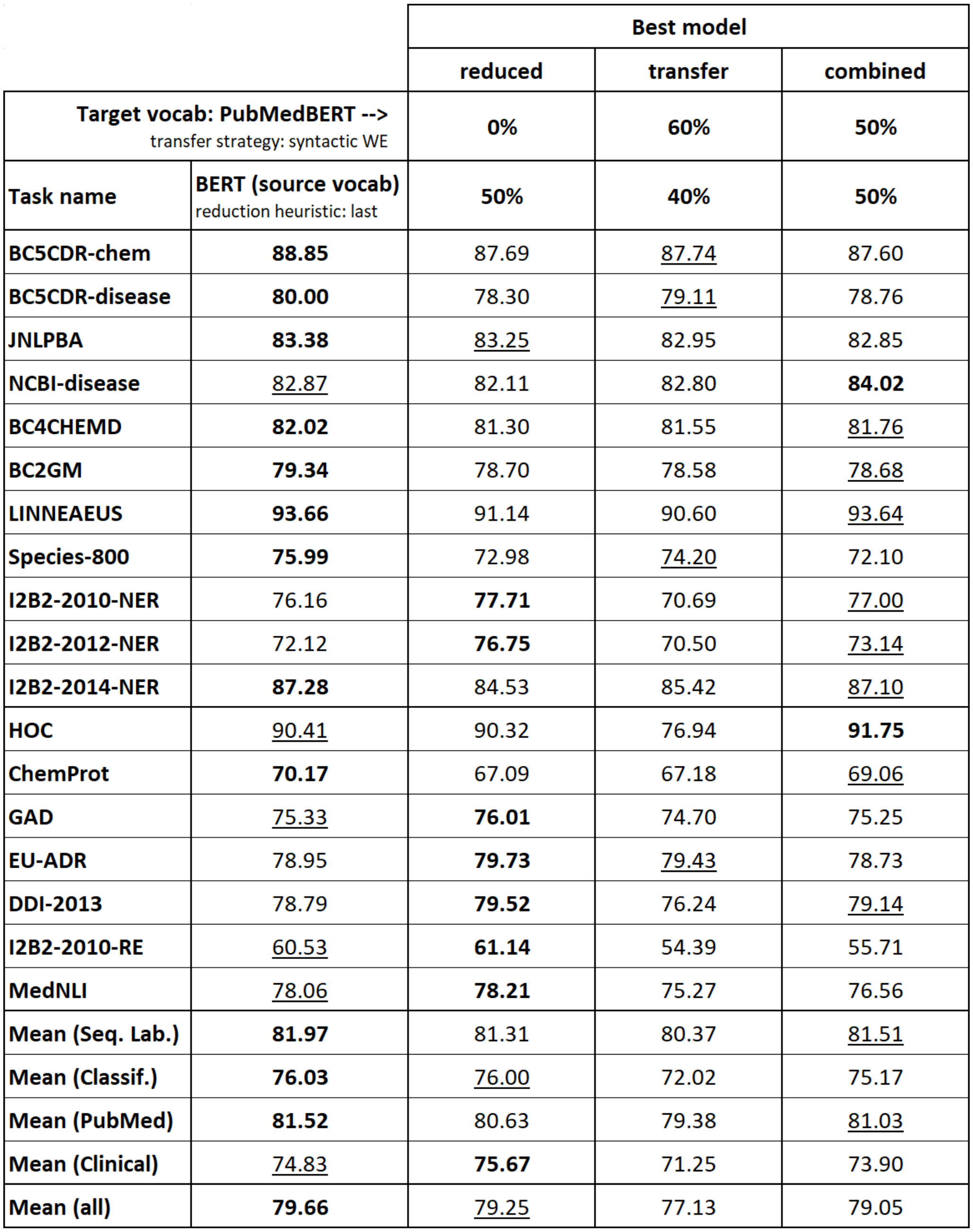
Target vocab: BERT --> transfer strategy: syntactic WE		**25%**	**50%**	**75%**	**100%**
Task name	**PubMedBERT (source vocab)** reduction heuristic: last	**75%**	**50%**	**25%**	**0%**
BC5CDR-chem	**91.43**	-0.63	-0.88	-1.83	-2.19
BC5CDR-disease	**83.58**	-0.80	-2.37	-4.06	-4.02
JNLPBA	84.24	-0.02	**0.26**	-0.31	-0.68
NCBI-disease	**83.80**	-0.50	-0.45	-2.73	-1.59
BC4CHEMD	**85.22**	-0.18	-0.63	-1.32	-1.22
BC2GM	**81.88**	-0.74	-0.98	-1.52	-2.39
LINNEAEUS	**95.63**	-0.77	-1.11	-0.75	-1.16
Species-800	79.53	**1.91**	-1.58	-1.94	-1.43
I2B2-2010-NER	**82.45**	-6.60	-6.94	-0.90	-8.60
I2B2-2012-NER	75.51	**3.49**	-1.24	-0.07	1.13
I2B2-2014-NER	87.34	-1.95	-3.52	-1.41	**0.84**
HOC	97.07	**0.86**	-0.84	-0.20	0.38
ChemProt	**76.82**	-0.04	-1.46	-1.40	-2.62
GAD	78.93	**0.76**	-0.13	-1.02	-0.20
EU-ADR	77.21	1.52	0.92	**1.63**	1.26
DDI-2013	81.72	**1.77**	-0.60	-2.27	-1.25
I2B2-2010-RE	59.83	0.09	1.43	**2.18**	-1.93
MedNLI	82.08	**0.43**	-1.36	-2.22	-2.80
Mean (Seq. Lab.)	**84.60**	-0.62	-1.77	-1.53	-1.94
Mean (Classif.)	79.09	**0.75**	-0.20	-0.52	-1.26
Mean (PubMed)	84.39	**0.24**	-0.76	-1.36	-1.32
Mean (Clinical)	**77.44**	-0.91	-2.33	-0.49	-2.27
Mean (all)	**82.46**	-0.13	-1.22	-1.17	-1.70

Table 6.8: Evaluation results, Combined models, PubMedBERT

Unfortunately, no correlation between the augmented volume of the vocabulary and the performance of the models on the BioLM benchmark can be identified in this experiment.

6.2.6 Summary

If we now compare all the best models from the experiments with BERT as a source model and PubMedBERT as a target model (see Table 6.9), we conclude that no approach was able to transfer any significant amount of knowledge from a domain-specific model to the general model. However, a special reducing heuristic of the general model vocabulary resulted in an improvement of many of the classification tasks, which was however counterbalanced by the performance drop on the other tasks.



		Best model		
		reduced	**transfer**	**combined**
Target vocab: PubMedBERT --> transfer strategy: syntactic WE		**0%**	**60%**	**50%**
Task name	**BERT (source vocab)** reduction heuristic: last	**50%**	**40%**	**50%**
BC5CDR-chem	**88.85**	87.69	87.74	87.60
BC5CDR-disease	**80.00**	78.30	79.11	78.76
JNLPBA	**83.38**	83.25	82.95	82.85
NCBI-disease	82.87	82.11	82.80	**84.02**
BC4CHEMD	**82.02**	81.30	81.55	81.76
BC2GM	**79.34**	78.70	78.58	78.68
LINNEAEUS	**93.66**	91.14	90.60	93.64
Species-800	**75.99**	72.98	74.20	72.10
I2B2-2010-NER	76.16	**77.71**	70.69	77.00
I2B2-2012-NER	72.12	**76.75**	70.50	73.14
I2B2-2014-NER	**87.28**	84.53	85.42	87.10
HOC	90.41	90.32	76.94	**91.75**
ChemProt	**70.17**	67.09	67.18	69.06
GAD	75.33	**76.01**	74.70	75.25
EU-ADR	78.95	**79.73**	79.43	78.73
DDI-2013	78.79	**79.52**	76.24	79.14
I2B2-2010-RE	60.53	**61.14**	54.39	55.71
MedNLI	78.06	**78.21**	75.27	76.56
Mean (Seq. Lab.)	**81.97**	81.31	80.37	81.51
Mean (Classif.)	**76.03**	76.00	72.02	75.17
Mean (PubMed)	**81.52**	80.63	79.38	81.03
Mean (Clinical)	74.83	**75.67**	71.25	73.90
Mean (all)	**79.66**	79.25	77.13	79.05

Table 6.9: Evaluation results, Summary

6.3 Qualitative Error Analysis

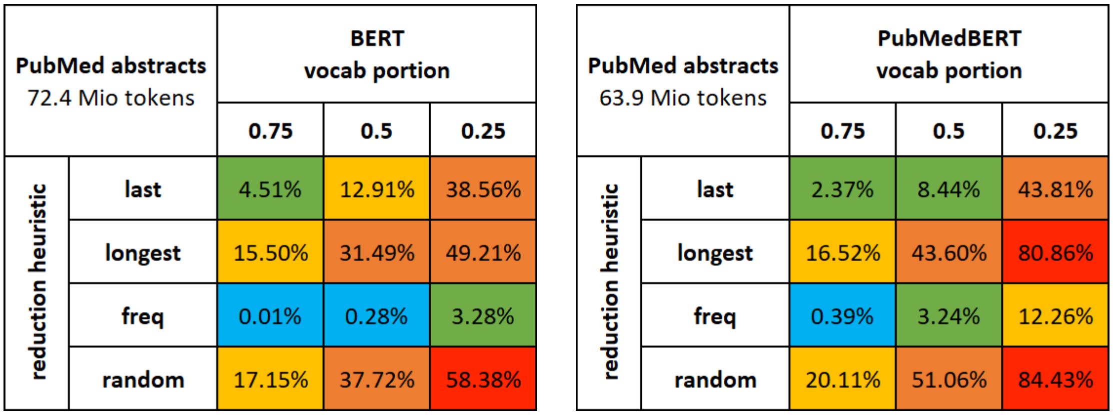
In this section, we will discuss the reasons for the insufficient performance of the custom models.

6.3.1 Reduced Models

For the reduced models, it was expected that performance would not improve significantly, since by simply removing the tokens, no new knowledge is injected into the model. However, by smartly reducing the vocabulary, noise at the input of the model could be eliminated to some extent, resulting in improved performance for some tasks. In particular, removing the last tokens from the vocabulary led to an improvement in performance on clinical tasks because these tokens occurred less frequently in the training set of the models (as can be seen in Figure 3.4) from Chapter ”Preliminary Analysis” and therefore, their context could not be learned well enough. When these tokens appeared in the tasks, the models couldn’t process them properly and as a result, the performance of the models decreased. When the rare tokens were removed, the models were able to replace these tokens by combining the shorter and more frequent tokens at the bottom levels, which barely decreased the overall performance and in some cases even led to an improvement.

Removing the longest tokens drops performance significantly because, as can be seen in Figure 3.3 from Chapter ”Preliminary Analysis”, a substantial portion of the long tokens is located at the beginning of the vocabulary, suggesting that these tokens were seen very frequently during training and their context was well understood by the models. When we remove these tokens and replace them with the shorter but less frequent tokens inside the model, we harm the performance.

In Table 6.10, we can see the change in the token count on 250k random abstracts extracted from the PubMed dataset, tokenized with the variants of reduced models based on BERT and PubMedBERT-fulltext. In the upper left corner of each sub-table, we can see the total amount of tokens resulting after tokenization by each of the base models. The tables themselves contain the relative increase in token count after tokenizing the



PubMed abstracts 72.4 Mio tokens		BERT vocab portion		
		0.75	0.5	0.25
reduction heuristic	last	4.51%	12.91%	38.56%
	longest	15.50%	31.49%	49.21%
	freq	0.01%	0.28%	3.28%
	random	17.15%	37.72%	58.38%

(a) BERT

PubMed abstracts 63.9 Mio tokens		PubMedBERT vocab portion		
		0.75	0.5	0.25
reduction heuristic	last	2.37%	8.44%	43.81%
	longest	16.52%	43.60%	80.86%
	freq	0.39%	3.24%	12.26%
	random	20.11%	51.06%	84.43%

(b) PubMedBERT

Table 6.10: Increase of context length due to vocabulary reduction

PubMed abstracts by each of the reduced models.

We first notice that tokenization by the BERT model leads to a higher token count compared to PubMedBERT, which was totally expected since PubMedBERT was trained on the text date from the same domain. In total, tokenization with BERT resulted in ca. 13.3% more tokens.

Comparing the reduced models based on BERT, it is immediately noticeable that the reduction of the vocabulary with the heuristic ”freq” increases the number of tokens the least. This means that most of the tokens in the vocabulary are not useful for tokenization at all since they probably originate from a domain other than biomedical. Surprisingly, removing the least used tokens didn't lead to better performance, as you can see in Table 6.4.

Furthermore, we observe that all other models, except the one with deleted 25% last tokens, lead to significantly higher token count, which leads to larger sequence length and might add additional noise to the model's input and result in the performance drop.

Regarding the PubMedBERT model, we observe that at first, the ”last” heuristic hardly increases the total number of tokens. Only when we remove 75% of all last tokens, does the number of tokens increase noticeably. While the ”freq” heuristic leads to an even lower increase in token count, the performance of these models didn't surpass that of the ”last” heuristic (see Table 6.5. For both other heuristics, the steep increase in token count happens much earlier. This significant increase in sequence length adds additional noise to the model's input and leads to the performance drop.

6.3.2 Knowledge Transfer

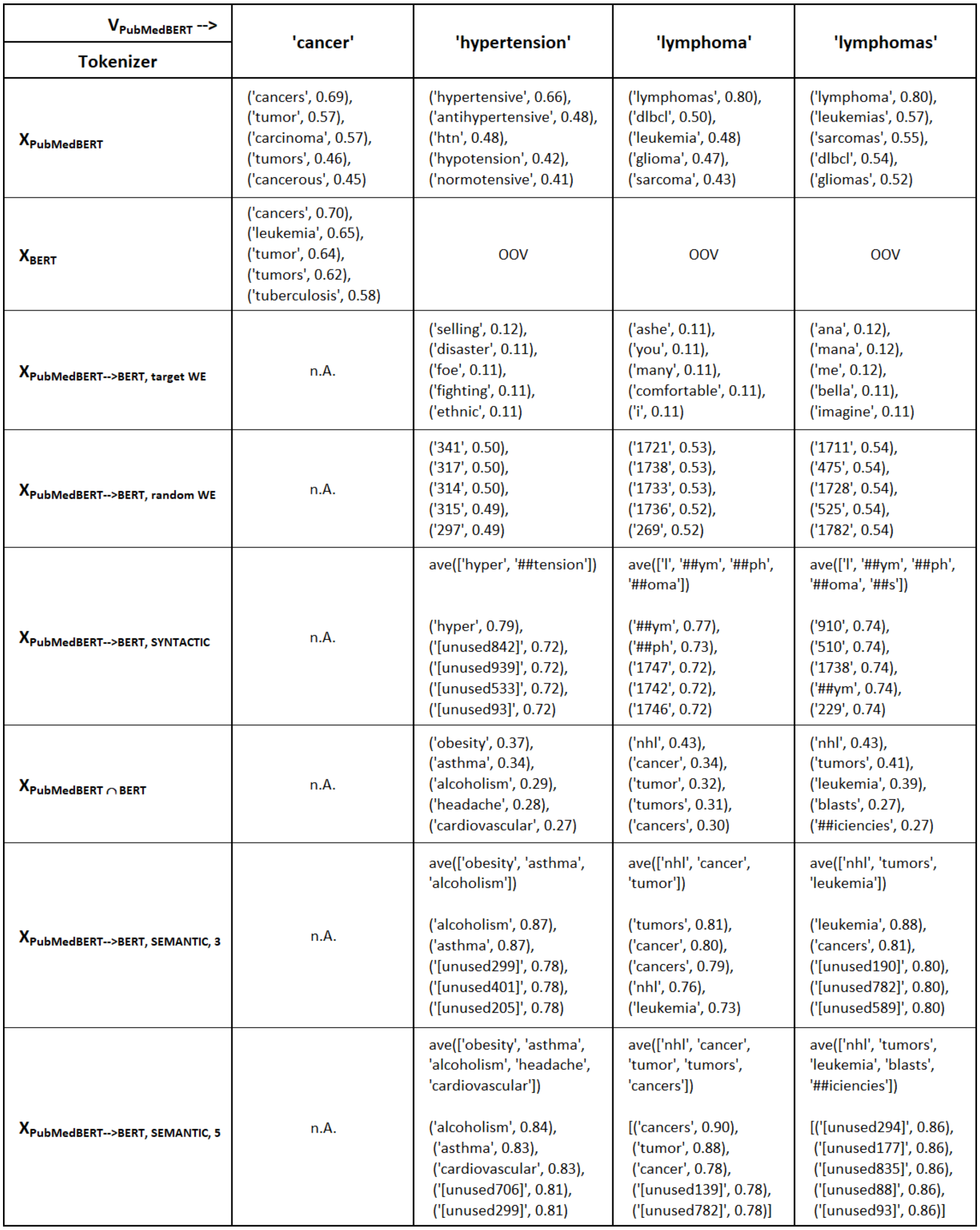
For the transferred knowledge models, we expected performance to improve by replacing the least important tokens in the vocabulary of the source model with the most important tokens from the vocabulary of the target model.

The research question here is which of the approaches better initializes the new tokens such as the source model can best make use of them.

Thereby, we analyze the knowledge transfer from PubMedBERT to BERT with different approaches. We aim to match the contextual neighbourhood in the resulting vocabulary as closely as possible to the target vocabulary by calculating the cosine similarity of a token and the remaining vocabulary.

We selected four example words from the medical domain: "cancer", "hypertension", "lymphoma" and its plural "lymphomas". The first word is included in both vocabularies and we want to verify that both models can contextually classify the word correctly. All other words are only included in PubMedBERT vocabulary and are therefore split by BERT, but with different numbers of sub-tokens. Our intention was to test whether the number of splits affects the syntactic knowledge transfer in any way.

The results are compiled in Table 6.11.



$V_{PubMedBERT}$ --> Tokenizer	'cancer'	'hypertension'	'lymphoma'	'lymphomas'
$X_{PubMedBERT}$	('cancers', 0.69), ('tumor', 0.57), ('carcinoma', 0.57), ('tumors', 0.46), ('cancerous', 0.45)	('hypertensive', 0.66), ('antihypertensive', 0.48), ('htn', 0.48), ('hypotension', 0.42), ('normotensive', 0.41)	('lymphomas', 0.80), ('dlbcl', 0.50), ('leukemia', 0.48) ('glioma', 0.47), ('sarcoma', 0.43)	('lymphoma', 0.80), ('leukemias', 0.57), ('sarcomas', 0.55), ('dlbcl', 0.54), ('gliomas', 0.52)
X_{BERT}	('cancers', 0.70), ('leukemia', 0.65), ('tumor', 0.64), ('tumors', 0.62), ('tuberculosis', 0.58)	OOV	OOV	OOV
$X_{PubMedBERT\text{-->}BERT,\ target\ WE}$	n.A.	('selling', 0.12), ('disaster', 0.11), ('foe', 0.11), ('fighting', 0.11), ('ethnic', 0.11)	('ashe', 0.11), ('you', 0.11), ('many', 0.11), ('comfortable', 0.11), ('i', 0.11)	('ana', 0.12), ('mana', 0.12), ('me', 0.12), ('bella', 0.11), ('imagine', 0.11)
$X_{PubMedBERT\text{-->}BERT,\ random\ WE}$	n.A.	('341', 0.50), ('317', 0.50), ('314', 0.50), ('315', 0.49), ('297', 0.49)	('1721', 0.53), ('1738', 0.53), ('1733', 0.53), ('1736', 0.52), ('269', 0.52)	('1711', 0.54), ('475', 0.54), ('1728', 0.54), ('525', 0.54), ('1782', 0.54)
$X_{PubMedBERT\text{-->}BERT,\ SYNTACTIC}$	n.A.	ave(['hyper', '##tension']) ('hyper', 0.79), ('[unused842]', 0.72), ('[unused939]', 0.72), ('[unused533]', 0.72), ('[unused93]', 0.72)	ave(['l', '##ym', '##ph', '##oma']) ('##ym', 0.77), ('##ph', 0.73), ('1747', 0.72), ('1742', 0.72), ('1746', 0.72)	ave(['l', '##ym', '##ph', '##oma', '##s']) ('910', 0.74), ('510', 0.74), ('1738', 0.74), ('##ym', 0.74), ('229', 0.74)
$X_{PubMedBERT \cap BERT}$	n.A.	('obesity', 0.37), ('asthma', 0.34), ('alcoholism', 0.29), ('headache', 0.28), ('cardiovascular', 0.27)	('nhl', 0.43), ('cancer', 0.34), ('tumor', 0.32), ('tumors', 0.31), ('cancers', 0.30)	('nhl', 0.43), ('tumors', 0.41), ('leukemia', 0.39), ('blasts', 0.27), ('##iciencies', 0.27)
$X_{PubMedBERT\text{-->}BERT,\ SEMANTIC,\ 3}$	n.A.	ave(['obesity', 'asthma', 'alcoholism']) ('alcoholism', 0.87), ('asthma', 0.87), ('[unused299]', 0.78), ('[unused401]', 0.78), ('[unused205]', 0.78)	ave(['nhl', 'cancer', 'tumor']) ('tumors', 0.81), ('cancer', 0.80), ('cancers', 0.79), ('nhl', 0.76), ('leukemia', 0.73)	ave(['nhl', 'tumors', 'leukemia']) ('leukemia', 0.88), ('cancers', 0.81), ('[unused190]', 0.80), ('[unused782]', 0.80), ('[unused589]', 0.80)
$X_{PubMedBERT\text{-->}BERT,\ SEMANTIC,\ 5}$	n.A.	ave(['obesity', 'asthma', 'alcoholism', 'headache', 'cardiovascular']) ('alcoholism', 0.84), ('asthma', 0.83), ('cardiovascular', 0.83), ('[unused706]', 0.81), ('[unused299]', 0.81)	ave(['nhl', 'cancer', 'tumor', 'tumors', 'cancers']) [('cancers', 0.90), ('tumor', 0.88), ('cancer', 0.78), ('[unused139]', 0.78), ('[unused782]', 0.78)]	ave(['nhl', 'tumors', 'leukemia', 'blasts', '##iciencies']) [('[unused294]', 0.86), ('[unused177]', 0.86), ('[unused835]', 0.86), ('[unused88]', 0.86), ('[unused93]', 0.86)]

Table 6.11: Contextual neighborhood before and after knowledge transfer

We first look at the token ”cancer”. BERT classifies the token contextually correctly, but the model suffers from a lack of biomedical terms since already the 5th token is contextually different from the other tokens in the list.

All the other two tokens only occur in the vocabulary of PubMedBERT, therefore the BERT vocabulary has to be augmented by these tokens and their embeddings must be re-initialized with one of the strategies.

- **Initialization with target embeddings**: The naïve replacement of the embeddings together with the respective tokens misses the contextual meaning completely, as the newly added tokens have very low cosine similarity values with some random tokens. The high importance of contextual similarity is also underlined by the worst evaluation performance in the whole experiment.
- **Random initialization**: The random initialization also misses the context but puts the new tokens in the range of numerals, which probably could not have been taught enough context to escape this range after the random initialization before training.
- **Syntactic transfer**: The syntactic initialization, which calculates the mean of the embeddings of sub-tokens, also pushes the tokens into the contextual area of contextually irrelevant tokens. This probably happens by mutual cancellation of the embeddings of sub-tokens.
- **Semantic transfer**: For semantic initialization, we first need to find the contextual neighbors of the token within the common tokens of the two vocabularies. While semantically related to the token, they have much smaller cosine similarity values than the top-5 most similar tokens in the target vocabulary. Thus, less context can be transferred but at least some transfer is possible due to a relatively large overlap of the vocabularies (about 40% of the tokens). We then compute an average of the embeddings of the contextually related neighbors, the number of which is a hyperparameter in itself. We exemplarily examine the 3 and 5 nearest neighbors.

  Since tokens from the intersection of both vocabularies that are contextually most similar to the token ”hypertension” describe the causes and symptoms of high blood pressure, semantic transfer performs well, resulting in contextual proximity to the tokens with high similarity from the list of common tokens. We also notice significantly higher similarity scores with the neighbors than for all other methods.

For the token ”lymphoma”, the 3 neighbors perform even better than for ”hypertension” and the resulting embedding falls within the region of oncological terms. For ”lymphomas”, however, this method reaches its limits, as the token was seen less frequently during PubMedBERT training and has lower similarity values with the contextually related tokens. Averaging then puts it closer to the randomly identified placeholders [unusedXXX].

This behavior becomes even more prominent when we increase the number of averaged contextual neighbors to 5. Thereby, the similarity values increase even further, as well as the number of randomly identified tokens in the direct surroundings. In the case of ”lymphomas” it leads to a situation where there are no contextual-related tokens left in the close neighbourhood and the embedding becomes practically unusable.

In terms of metrics, all knowledge transfer approaches resulted in performance below the source models (see Table 6.6). Only the syntactic transfer could somewhat keep up with the best custom models from other experiments.

6.3.3 Combined Models

For the combined models, we aimed to combine the best approaches from the previous experiments and as a result improve the performance of the models. First, we kept the first half of the source vocabulary, including embeddings, unchanged, as this resulted in the best performance by reducing noise. Second, we added the most important domain-specific tokens from the target vocabulary and assigned syntactic initialization to their embeddings. The second step also added back to the vocabulary the common tokens, some of which were removed by the first step and which, after defining the syntactic approach, received the embeddings of the source vocabulary again. The remaining new tokens were initialized with the mean of the embeddings of sub-tokens.

Unfortunately, this approach did not improve the performance on Bio-LM, since the domain-specific tokens hardly differed from the randomly initialized tokens of the source model due to the averaging of the embeddings as we saw in the previous section. We hypothesize that the model did not have enough computing time to adapt to the biomedical domain due to the relatively short fine-tuning process through the training sets of the individual tasks.

6.4 Discussion

In this section, we want to discuss the implications, limitations, and potential drawbacks of our experimental setup and findings.

6.4.1 Interpretation of Results

The results of the experiments provide valuable insights into the performance of various model modifications on the Bio-LM benchmark. While the findings shed light on the impact of different strategies, there are several aspects to consider when interpreting the results.

One of the key aspects of our study was the exploration of vocabulary reduction strategies and knowledge transfer approaches. While our experiments showed some improvements in specific scenarios, it's important to acknowledge the limitations of these techniques. Vocabulary reduction, particularly the "last" heuristic, yielded benefits in terms of noise reduction and improved performance in certain tasks. However, this strategy doesn't provide an opportunity for the model to gain new knowledge. Additionally, the reduction of tokens might lead to information loss, especially when dealing with infrequent but relevant terms.

Similarly, the knowledge transfer approaches, while conceptually promising, faced challenges in terms of effective embedding initialization and contextual understanding. The use of syntactic transfer showed better results than semantic transfer, indicating that structural information might be more readily transferable than purely semantic context.

However, even with syntactic transfer, the models failed to consistently outperform the source models. This could be attributed to the complex interplay between token embeddings and their corresponding context, which is hard to capture solely through averaging sub-token embeddings.

Furthermore, the proposed custom models were primarily designed to address the challenges of domain adaptation and limited data availability. However, some tasks within the Bio-LM benchmark might require a more nuanced understanding of the biomedical domain. For instance, tasks involving complex medical relationships or domain-specific terminology might need more advanced techniques beyond the scope of this study.

6.4.2 Generalization of Findings

Our experiments were conducted within the context of biomedical text data, and the effectiveness of the proposed strategies could vary when applied to other domains. While the underlying principles of adaptation and transfer might hold true across domains, the specific characteristics of different domains might require tailored approaches.

To assess the generalization of the findings, similar experiments could be conducted on diverse text datasets spanning various domains. Comparing the results across domains would provide insights into whether adaptation strategies can be transferred to other domains and show whether certain approaches consistently outperform others.

6.4.3 Lack of Contextual Pre-training

Our experiments focused on the adaptation of pre-trained models to biomedical tasks. However, the lack of further contextual training on biomedical text might have negatively impacted the models' performance. Model weights of models like PubMedBERT have been designed to capture domain-specific information during pre-training, which our approach doesn't fully leverage. Incorporating further domain-specific training with the MLM approach could potentially enhance the models' ability to transfer knowledge and adapt to biomedical tasks.

6.4.4 Token Embedding Modifications

The experiments involving modifications to token embeddings reveal that certain strategies, such as reducing vocabulary or transferring knowledge, do not always lead to consistent improvements. The challenges associated with initializing embeddings for domain-specific tokens highlight the complexities of transferring knowledge between models. While certain embedding initialization methods showed promise, the limitations in capturing semantic meaning across different vocabularies remain.

6.4.5 Hyperparameter Sensitivity

Hyperparameter optimization is a critical aspect of model training and evaluation. In our experiments, we identified hyperparameters that led to stable and optimal performance. However, the choice of hyperparameters can greatly affect the results, and the optimal set might differ for different models and tasks. Our choice of hyperparameters might not be universally applicable and could require further tuning in different scenarios.

6.4.6 Task-Specific Strategies

It is important to recognize that the effectiveness of model modifications can vary across different tasks. Some tasks might inherently require more sophisticated adaptations or domain-specific strategies that were not explored in this study. A task-by-task analysis of the proposed modifications could provide a more granular understanding of their impact.

6.4.7 Model Complexity and Generalization

The experiments mainly focus on variations of transformer-based models. More complex architectures or novel approaches might yield different results. Additionally, the generalization capability of the models to handle out-of-domain data like different tokenization techniques or tasks not covered by the benchmark remains an important limitation.

6.5 Summary

This chapter discusses experimental results on knowledge transfer and adaptation in the biomedical domain. It explores strategies like vocabulary reduction, token embedding modifications, and knowledge transfer between models. Results show mixed outcomes for vocabulary reduction, with some heuristics improving specific tasks and others causing information loss. Knowledge transfer experiments indicate that syntactic transfer performs better than semantic transfer, but none of the approaches consistently outperforms source models. The combined model approach did not yield significant improvements, possibly due to limited contextual pre-training on domain-specific text. The findings emphasize the need for tailored strategies and domain-specific training for effective adaptation.

Chapter 7

Conclusion and Future Work

7.1 Assessment of Hypotheses

Based on the evaluation results, we provide the following assessment of the hypotheses:

1. **Performance Proportionality to Domain Adaptation**: The results support this hypothesis. BlueBERT outperformed BERT on most tasks, and PubMedBERT models exhibited improved performance, confirming that domain adaptation enhances performance on domain-specific tasks.

2. **Vocabulary Reduction Heuristics and Task Performance**: Vocabulary reduction using the "last" heuristic positively impacted classification tasks without significant degradation in other tasks, confirming the hypothesis to some extent but we could not derive a clear correlation between any reduction heuristic and its reduction portion and the model's performance.

3. **Knowledge Transfer from One Model to Another**: Syntactic knowledge transfer improved performance on some tasks, but the semantic transfer did not perform well. Probably, the method needs further training to develop its full potential, which would be possible with MLM training of PubMed abstracts for some epochs.

4. **Combined Approaches for Enhanced Knowledge Transfer**: Combining vo-

cabulary reduction with syntactic knowledge transfer showed improved performance on some tasks, but the overall knowledge transfer remained limited. Again, we would need to investigate whether further training on biomedical data would lead to improvement in results.

7.2 Summary

In this study, we conducted a comprehensive exploration of various vocabulary modifications and adaptation strategies to enhance the performance of transformer-based language models on biomedical language understanding tasks. Through a series of experiments, we investigated vocabulary reduction, token embedding pre-initialization, knowledge transfer, and combined approaches to assess their impact on task performance. Our findings emphasize the complexity of domain adaptation and knowledge transfer in the context of biomedical text analysis.

The results of our experiments revealed nuanced interactions between model modifications and task performance. While some strategies demonstrated improvements in specific scenarios, no single approach consistently outperformed the source models across all tasks. The performance of domain-specific models like PubMedBERT emphasizes the value of domain adaptation, yet challenges in generalization were evident.

The experiments involving token embedding modifications highlighted the difficulties in initializing embeddings for domain-specific tokens. While certain strategies showed promise, the ability to capture semantic meaning across different vocabularies remained a significant challenge.

It is important to note that the experiment results of our proposed methodology were domain-dependent, emphasizing the need for tailored approaches to adapt models to specific domains. The generalisation of our findings across different domains remains an open question for further investigation.

In conclusion, this study contributes to our understanding of domain adaptation and knowledge transfer in the context of transformer-based language models. While our findings provide valuable insights, they also raise new questions and challenges. In the end, effective knowledge transfer and domain adaptation is an important goal to improve the capabilities of language models in specific domains.

7.3 Future Work

While this study offers valuable insights into domain adaptation and knowledge transfer for biomedical language understanding, there are many open questions for further exploration and improvement. The following directions can guide future research to expand upon the findings presented in this work:

- **Advanced Knowledge Transfer Approaches**: The limitations observed in the semantic and syntactic knowledge transfer experiments point towards the need for more sophisticated approaches. Future research could lead to advanced methods that better capture the nuances of transferring knowledge between models. Techniques such as further MLM training of data from the target domain, fine-tuning with a mix of domain-specific and general-domain data or leveraging external resources like ontologies and medical databases could be explored to improve knowledge transfer effectiveness.

- **Hybrid Model Architectures**: Given the challenges in achieving consistent improvements with individual strategies, hybrid model architectures could offer a promising direction. Combining multiple strategies, such as vocabulary reduction, pre-initialization, and knowledge transfer, in a single model could lead to synergistic effects. Investigating the optimal combination of these approaches and designing architectures that adapt dynamically based on task characteristics could yield more robust and adaptable models.

- **Cross-Domain Evaluation**: Expanding the evaluation beyond the biomedical domain could provide a broader perspective on the effectiveness of the proposed strategies. Applying the same set of experiments to different domains, such as legal or financial text, could reveal patterns in adaptation and transferability. This cross-domain evaluation would help determine whether certain strategies consistently outperform others and generalize across different domains.

- **Task-Specific Adaptation**: While this study considered a range of tasks within the Bio-LM benchmark, some tasks might require task-specific adaptation strategies. Future work could focus on understanding the characteristics of challenging tasks and developing targeted approaches to improve model performance on those tasks. These approaches could involve leveraging task-specific data augmentation techniques, designing task-specific pre-training objectives, or incorporating external domain-specific resources.
- **Expanding Beyond Transformer Models**: While transformer-based models have revolutionized NLP, exploring alternative architectures could yield new insights into domain adaptation and knowledge transfer. Investigating models that incorporate explicit domain-specific knowledge, memory-augmented networks, or architectures with attention mechanisms that adapt dynamically to domain shifts could lead to novel strategies for improving model performance in specialized domains.

7.4 Acknowledgments

We gratefully acknowledge the support and resources provided by the DATEXIS research group at the University of Applied Sciences of Berlin.

Bibliography

[1] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, *Bert: Pre-training of deep bidirectional transformers for language understanding*, 2019. arXiv: 1810.04805 [cs.CL].

[2] Y. Gu, R. Tinn, H. Cheng, *et al.*, "Domain-specific language model pretraining for biomedical natural language processing," *ACM Transactions on Computing for Healthcare (HEALTH)*, vol. 3, pp. 1–23, 2020.

[3] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, *Attention is all you need*, 2017. arXiv: 1706.03762 [cs.CL].

[4] A. Radford and K. Narasimhan, "Improving language understanding by generative pre-training," 2018. [Online]. Available: https://api.semanticscholar.org/CorpusID:49313245.

[5] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:160025533.

[6] T. B. Brown, B. Mann, N. Ryder, *et al.*, "Language models are few-shot learners," *ArXiv*, vol. abs/2005.14165, 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:218971783.

[7] OpenAI, "Gpt-4 technical report," *ArXiv*, vol. abs/2303.08774, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:257532815.

[8] Y. Peng, S. Yan, and Z. Lu, "Transfer learning in biomedical natural language processing: An evaluation of bert and elmo on ten benchmarking datasets," in *Proceedings of the 2019 Workshop on Biomedical Natural Language Processing (BioNLP 2019)*, 2019, pp. 58–65.

[9] T. Mikolov, K. Chen, G. S. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *International Conference on Learning Representations*, 2013.

[10] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," *ArXiv*, vol. abs/1310.4546, 2013.

[11] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Conference on Empirical Methods in Natural Language Processing*, 2014.

[12] M. E. Peters, M. Neumann, M. Iyyer, *et al.*, "Deep contextualized word representations," in *North American Chapter of the Association for Computational Linguistics*, 2018.

[13] S. Zhao, R. Gupta, Y. Song, and D. Zhou, "Extreme language model compression with optimal subwords and shared projections," *ArXiv*, vol. abs/1909.11687, 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:202888875.

[14] S. Zhao, R. Gupta, Y. Song, and D. Zhou, "Extremely small bert models from mixed-vocabulary training," in *Conference of the European Chapter of the Association for Computational Linguistics*, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:231855076.

[15] A. Kolesnikova, Y. Kuratov, V. Konovalov, and M. S. Burtsev, "Knowledge distillation of russian language models with reduction of vocabulary," *ArXiv*, vol. abs/2205.02340, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:248524841.

[16] S. Sato, J. Sakuma, N. Yoshinaga, M. Toyoda, and M. Kitsuregawa, "Vocabulary adaptation for domain adaptation in neural machine translation," in *Findings*, 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:226283659.

[17] V. Sachidananda, J. S. Kessler, and Y.-A. Lai, "Efficient domain adaptation of language models via adaptive tokenization," *ArXiv*, vol. abs/2109.07460, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:237513469.

[18] A. Ladkat, A. Miyajiwala, S. Jagadale, R. Kulkarni, and R. Joshi, "Towards simple and efficient task-adaptive pre-training for text classification," in *AACL*, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:252544759.

[19] J. Hong, T. Kim, H. Lim, and J. Choo, "Avocado: Strategy for adapting vocabulary to downstream domain," in *Conference on Empirical Methods in Natural Language Processing*, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:239885682.

[20] I. Samenko, A. Tikhonov, B. M. Kozlovskii, and I. P. Yamshchikov, "Fine-tuning transformers: Vocabulary transfer," *ArXiv*, vol. abs/2112.14569, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:245537907.

[21] V. D. Mosin and I. P. Yamshchikov, "Vocabulary transfer for medical texts," *ArXiv*, vol. abs/2208.02554, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:251320581.

[22] P. Lewis, M. Ott, J. Du, and V. Stoyanov, "Pretrained language models for biomedical and clinical tasks: Understanding and extending the state-of-the-art," in *Clinical Natural Language Processing Workshop*, 2020.

[23] A. Paszke, S. Gross, F. Massa, *et al.*, *Pytorch: An imperative style, high-performance deep learning library*, 2019. arXiv: 1912.01703 [cs.LG].

[24] T. Wolf, L. Debut, V. Sanh, *et al.*, "Transformers: State-of-the-art natural language processing," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, Online: Association for Computational Linguistics, Oct. 2020, pp. 38–45. DOI: 10.18653/v1/2020.emnlp-demos.6. [Online]. Available: https://aclanthology.org/2020.emnlp-demos.6.