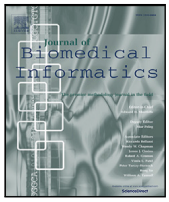
Journal of Biomedical Informatics 146 (2023) 104486

Contents lists available at ScienceDirect

Journal of Biomedical Informatics

journal homepage: www.elsevier.com/locate/yjbin



A self-supervised language model selection strategy for biomedical question answering

Negar Arabzadeh [a,*], Ebrahim Bagheri [b]

[a] University of Waterloo, Waterloo, ON, Canada
[b] Toronto Metropolitan University, Toronto, ON, Canada

ARTICLE INFO

Keywords:
Biomedical question answering
Domain-specific language model
General-purpose language model
Self-supervised learning

ABSTRACT

Large neural-based Pre-trained Language Models (PLM) have recently gained much attention due to their noteworthy performance in many downstream Information Retrieval (IR) and Natural Language Processing (NLP) tasks. PLMs can be categorized as either *general-purpose*, which are trained on resources such as large-scale Web corpora, and *domain-specific* which are trained on in-domain or mixed-domain corpora. While domain-specific PLMs have shown promising performance on domain-specific tasks, they are significantly more computationally expensive compared to general-purpose PLMs as they have to be either retrained or trained from scratch. The objective of our work in this paper is to explore whether it would be possible to leverage general-purpose PLMs to show competitive performance to domain-specific PLMs without the need for expensive retraining of the PLMs for domain-specific tasks. By focusing specifically on the recent BioASQ Biomedical Question Answering task, we show how different general-purpose PLMs show synergistic behaviour in terms of performance, which can lead to overall notable performance improvement when used in tandem with each other. More concretely, given a set of general-purpose PLMs, we propose a *self-supervised* method for training a classifier that systematically selects the PLM that is most likely to answer the question correctly on a per-input basis. We show that through such a selection strategy, the performance of general-purpose PLMs can become competitive with domain-specific PLMs while remaining computationally light since there is no need to retrain the large language model itself. We run experiments on the BioASQ dataset, which is a large-scale biomedical question-answering benchmark. We show that utilizing our proposed selection strategy can show statistically significant performance improvements on general-purpose language models with an average of 16.7% when using only lighter models such as `DistilBERT` and `DistilRoBERTa`, as well as 14.2% improvement when using relatively larger models such as `BERT` and `RoBERTa` and so, their performance become competitive with domain-specific large language models such as `PubMedBERT`.

1. Introduction

Large Pre-trained Language Models (PLM) have significantly influenced the performance of many Information Retrieval (IR) and Natural Language Processing (NLP) downstream tasks [1–3]. BERT has shown to be a notable example of such PLMs especially because its pre-training process is performed in a self-supervised manner over an unlabelled corpus [4]. General-purpose PLMs are often pre-trained on general domain corpora such as the Web, book corpora, Wikipedia, and news corpora, among others. More recently, researchers have began looking into developing domain-specific PLMs that would capture the detailed semantics of specific domains. Authors such as Gu et al. [5] argue that the development of domain-specific PLMs is essential since using general-purpose PLMs is only warranted when (1) the target and source domains are highly homogeneous and comparable; and, (2) training data for the specific domain under consideration is in short supply [6–11]. Therefore, they argue that for domains such as biomedicine, which enjoys abundant publicly available unlabelled corpora, continuing pre-training from a general-purpose PLM, i.e., mixed domain pre-training, can be a questionable endeavour and in some cases such a transfer learning strategy could even be harmful [5]. As such, they hypothesize that once there is a sufficient amount of in-domain training data, training domain-specific PLMs from scratch can lead to significant boosts on domain-specific tasks compared to continuing pre-training of a general-purpose PLM.

Empirical studies show that performance improvements are in fact observed quite significantly when training domain-specific PLMs from

* Corresponding author.
E-mail address: narabzad@uwaterloo.ca (N. Arabzadeh).
[1] https://pubmed.ncbi.nlm.nih.gov/

https://doi.org/10.1016/j.jbi.2023.104486
Received 24 January 2023; Received in revised form 14 August 2023; Accepted 1 September 2023
Available online 16 September 2023
1532-0464/© 2023 Elsevier Inc. All rights reserved.

scratch [5,12,13]. However, in this paper, we argue that the performance improvements shown by domain-specific PLMs come at *noticeable costs*. The first of these costs relates to the fact that training a domain-specific PLM from scratch is computationally expensive. For instance, PubMedBERT [5], which is a biomedical PLM that was trained from scratch on PubMed[1] abstracts, has been trained on 16 v100 GPUs on a DGX-2 machine, which costed over $400K USD to complete. Such powerful computational resources are hard to access for many researchers and institutions. The second cost is that training a domain-specific PLM is not only computationally expensive but it is also extremely time-consuming. For instance, the domain-specific PLM, namely PubMedBERT, was trained for five days on the above-mentioned computational resources. Last but not least, while domains such as biomedicine have a sufficient amount of publicly available corpora for training domain-specific PLMs, their availability of resources across the subdomains of biomedicine is not evenly distributed and some areas face significant privacy considerations to use their biomedical data to train PLMs.

On this basis and in this paper, we are interested in exploring whether and how one could leverage possible synergies between various general-purpose PLMs in order to make their collective performance on domain-specific downstream tasks competitive to domain-specific PLMs. We are specifically seeking to study whether it would be possible to (1) enhance the performance of domain-specific tasks when utilizing general-purpose PLMs; and, (2) design a strategy for using different general-purpose PLMs in order to obtain competitive performance in comparison to domain-specific PLMs on domain-specific tasks, specifically on the biomedical question task.

In summary, we argue that training a domain-specific language model is not always feasible, and even so, it would be very expensive to train domain-specific language models. Therefore, it is necessary to be able to employ general-purpose PLMs to improve their performance on domain-specific tasks. As such, we propose a simple yet effective strategy to benefit from the strengths and the complementarity behaviour of general-purpose language models. We note that while in highly critical domains such as healthcare and finance, performance improvements could be considered to be more valuable than computational cost, abundant data for training may not be available for some other domains such as those related to the legal or enterprise applications, mostly due to privacy issues. Here, we target the biomedical domain as a specific domain and we show that it is possible to achieve better performance without utilizing domain-specific PLMs. We leave further exploration on other domains for the future studies.

In this paper, we run experiments on the BioASQ7b and BioASQ8b datasets, which are large-scale biomedical question-answering benchmarks. We show that general-purpose PLMs can complement each other leading to better overall performance on these datasets. More specifically, we demonstrate that utilizing our proposed strategy can boost the performance of general-purpose language models on biomedical question-answering tasks in a statistically significant way with an average of 16.7% when using only lighter models such as DistilBERT and DistilRoBERTa, as well as 14.2% improvement when relatively larger models such as BERT and RoBERTa arre used. Additionally, we show that the performance obtained through this process is competitive with domain-specific PLMs, such as PubMedBERT yet does not require as many resources to be trained.

The rest of this paper is structured as follows. In Section 2, we review the literature on domain-specific language models, specifically in the biomedical field. Next, in Section 3, we define the problem of general–purpose language model selection for domain-specific tasks. Further, in Section 4, we study the complementary behaviour of general-purpose PLMs in addressing biomedical QA tasks. We propose a self-supervised selection strategy between general-purpose language models in Section 5 and describe our methodology in this section. We discuss the details of the experimental setup and show the results of our proposed approach in Section 6. Finally, we conclude the work and point out potential future works in the last section.

2. Related work

2.1. Domain-specific language models

Previous research have demonstrated that when addressing a domain-specific task, pre-training with in-domain data can usually lead to better performance [5,12,14]. Such research shows that while continuing pre-training a general-purpose language model for domain-specific tasks sounds reasonable, it may suffer from problems such as lack of domain-specific vocabulary. Therefore, for domains that enjoy abundantly available data, such as biomedicine and specifically biomedical question answering [15–17], researchers tend to favour training domain-specific language models on in-domain data such as those available through PubMed [18–20]. As expected and similar to other domains, work on PLMs for the biomedical domain has shown that having domain-specific language models can actually lead to improved performance on downstream biomedical NLP tasks [12,13,18,21–23].

Various biomedical PLMs have been recently proposed, which include BioBERT [13], BlueBERT [21] and BioClinicalBERT [24] to name a few. These methods are also known as *Mixed-domain Language models* and are initialized from a general purpose language model such as BERT and are then continued pre-training on domain-specific data. For instance, BioBERT is based on the continued pre-training of BERT on PubMed abstracts and PubMed Central (PMC) full-text articles. BlueBERT is based on BERT with continued pre-training on PubMed as well as de-identified clinical notes from MIMIC-III [25]. In addition, BioClinicalBERT is another recently proposed domain-specific language model that has been continued pre-training on MIMIC dataset [24,25]. From an empirical perspective, Gu et al. [5] have studied the impact of utilizing domain-specific vs general-purpose language models for tackling domain-specific tasks with a focus on the biomedical area. They investigated whether in-domain pre-training, e.g., pre-training from scratch on domain-specific corpora such as PubMed, can surpass pre-training on general-purpose or mixed-domain corpora. Due to the diversity of NLP tasks and available datasets, conducting a fair comparison between different PLMs and their core impact on various tasks can be quite challenging. Thus, to make such a comparison, it is required to have a valid benchmark with predefined tasks and gold standards. Therefore, inspired by BLUE [21], Gu et al. presented a comprehensive Biomedical Language Understanding & Reasoning Benchmark (BLURB)[2] with a focus on PubMed-based biomedical applications. BLURB constitutes an exhaustive set of biomedical NLP tasks, including named entity recognition (NER), evidence-based medical information extraction (PICO), relation extraction, sentence similarity, document classification, and question answering. Based on BLURB, these researchers showed that training a domain-specific PLM from scratch outperforms other general-purpose or mixed-domain PLMs when fine-tuned on each task.

One of the tasks in BLURB which shows a great performance gap when using a general-purpose language model compared to when a domain-specific PLM is used is the Biomedical Question Answering (QA) task [5,26]. Empirical work has shown that biomedical QA significantly benefits from domain-specific language models by a high margin due to challenges such as sub-word vocabulary set. However, we note that training a domain-specific language model is quite expensive and also not feasible for all domains. Therefore, in this work, we focus on minimizing this gap between the performance of general-purpose language models and domain-specific biomedical language models by increasing the performance of fine-tuned general-purpose language models through a self-supervised PLM selection strategy for the Biomedical QA tasks.

[2] https://microsoft.github.io/BLURB/

2.2. Model integration

To the best of our knowledge, the potential synergy between general-purpose language models has not been explored in past literature. Therefore, our work is among the first to investigate the role of general-purpose PLM selection in the context of biomedical question answering. However, the idea of interpolating a set of methods to achieve a goal has been previously explored in other tasks, particularly in the information retrieval and natural language processing communities. In natural language processing, model interpolation involves combining the output of multiple models to improve performance on a given task [27–29]. This technique has been used in various NLP tasks, including machine translation, text classification, and information retrieval [30–32]. The basic idea is to assign a weight to each model's output and then combine them to produce a final prediction [33,34]. The weights can be determined by optimizing a loss function, such as cross-entropy, on a validation set. Other techniques, such as ensembling methods (e.g., bagging, boosting, or stacking), have been used to improve the robustness and accuracy of NLP models [35,36]. Overall, model interpolation is a powerful technique for improving NLP model performance, especially in cases where a single model may not perform well on its own [37]. It allows the combination of complementary strengths from multiple models, resulting in more accurate and robust predictions.

Interpolation of different methods and investigating the synergy between them to tackle a specific task has also been widely used in Information Retrieval problems. In information retrieval, interpolation between rankers has been shown to boost performance over a single retriever because each ranker can potentially address a different subset of queries [38–43]. Researchers have shown effectiveness at both training and inference levels [39,40]. For example, Wang et al. [38] demonstrated that interpolating traditional bag-of-word-based sparse retrievers, such as BM25, is necessary for neural-based dense retrievers to perform effectively, and the gains provided by the interpolation are significant. Interpolation has also been explored for web search ranking. The authors in [44] showed that model interpolation, although simple, not only obtained significant boosts in performance but also increased method generalizability.

Our work in this paper is inspired by the idea of using multiple retrievers in the information retrieval community. Several researchers have shown that when answering a query, different retrievers may show dissimilar performances, whereby some retrievers are more effective on a certain subset of queries. Therefore, these researchers suggest that a *query routing* strategy could be used, which would determine on a case-by-case basis which retriever would be used to address the incoming query; hence, collectively maximizing retrieval effectiveness [43,45,46]. Similarly, in this paper, we leverage the idea of query routing by implementing a self-supervised PLM selection strategy that would decide which general-purpose language model would be best for answering a certain incoming question in the biomedical question-answering task.

3. Problem definition

The focus of our work in this paper is on the Biomedical Question Answering (QA) task. The objective of this task is to predict the exact answer (yes/no) to a given input question, given a relevant snippet.

More formally, given a set of questions $Q = \{q_1, q_2, q_3, \dots, q_m\}$, aligned with their relevant snippets set $R = \{r_{q_1}, r_{q_2}, r_{q_3}, \dots, r_{q_m}\}$ and their ideal answers $A = \{a_{q_1}, a_{q_2}, a_{q_3}, \dots, a_{q_m}\}$ where $a_{q_i} \in \{1, 0\}$ is a binary value for class 1 as yes and class 0 as no, we find the answer (yes/no) for each question $q_i \in Q$ by employing a fine-tuned language model L to encode each question $q_i \in Q$ and its relevant snippets $r_{q_i} \in R$. As such, we obtain the predicted answer to q_i and r_{q_i} as $f_L(q_i, r_{q_i}|L)$ where $f_L(q_i, r_{q_i}|L) \in \{1, 0\}$.

One of the most well-known standard datasets for biomedical question answering is based on the BioASQ challenge[3] [26,47,48]. BioASQ is a large-scale question-answering challenge, addressing a wide range of a question-answering tasks for yes/no, factoid, list and summary questions. In this work, similar to [5] and without loss of generality, we focus on yes/no questions and treat it as a classification task. We adopt BioASQ7b and BioASQ8b datasets, which are specifically released for the yes/no question-answering challenge. Table 1 shows a few examples of questions aligned with their relevant snippets and ideal answers from the BioASQ7b dataset.

In the context of the biomedical question answering task, we first study the potential of leveraging general-purpose language models for addressing domain-specific tasks by exploring to what extent different general-purpose PLMs overlap with each other in terms of success in answering biomedical questions. We investigate whether there is a significant number of non-overlapping questions that can be answered by different general-purpose PLMs. Our work shows that general-purpose PLMs do in fact have complementarity when addressing different questions. As such, it might be possible to maximize the potential of employing various general-purpose PLMs in order to address a domain-specific task.

Given general-purpose PLMs perform differently depending on the question, we focus on developing a *self-supervised* PLM selection strategy that would effectively decide which PLM each input question would need to be routed to for maximal efficiency. We find that such a self-supervised selection strategy can be quite effective in selecting the most appropriate general-purpose PLM per input question and can lead to improved performance on the biomedical questions answering task.

An effective supervised selection strategy over general-purpose PLMs would reduce the need for domain-specific PLMs, which can be expensive in terms of time and computation needed to train them. More importantly, this approach would be beneficial in domains where there is not enough domain-specific data available for training a large language model. For these reasons, our proposed strategy can increase the performance of general-purpose PLMs on domain-specific tasks. We perform experiments on four different general-purpose PLMs, namely `BERT`, `RoBERTa` and their distilled versions `DistilBERT` and `DistilRoBERTa`.

We note that for replicability purposes, we made our code publicly available at https://github.com/Narabzad/Language-model-selection-strategy

4. Synergy between general-purpose PLMs

The core idea behind our work in this paper is to investigate the potential of utilizing general-purpose PLMs for addressing the domain-specific biomedical question answering task. More specifically, we investigate whether general-purpose PLMs have complementary characteristics that would allow us to systematically choose between them to increase their individual performances. In other words, we are interested in systematically deciding which general-purpose PLMs would be the best for answering each question on a case-by-case basis in order to improve overall performance. To explore whether such complementarity exists between different general-purpose PLMs, we assume that there exists an *Oracle* that is able to determine the best general-purpose PLM on per input question basis. Such an Oracle would allow us to route each incoming question to the best general-purpose PLM and hence will provide the best possible overall performance.

More formally, given a set of general-purpose PLMs $L = \{L_1, L_2..., L_n\}$, where L_i is a general purpose language model, we define an *oracle score* S for question set $Q = \{q_1, q_2, \dots, q_m\}$ accompanied with their relevant snippet set $R = \{r_1, r_2, \dots, r_m\}$ and PLM set L as follows:

$$S(Q, R, L) = \frac{1}{m} \sum_{j=1}^{m} max\{ACC(q_j, r_j, L_i)|i \in \{1, 2, \dots, n\}\} \quad (1)$$

[3] http://www.bioasq.org/

Table 1
Examples of questions from BioASQ7 aligned with their relevant snippets and the ideal answer.

Questions	Is Pim-1 a protein phosphatase?	Is selenocysteine an aminoacid?
Relevant Snippet	Pim-1 proto-oncogene, serine/threonine kinase (PIM-1) phosphorylates a series of substrates to exert its oncogenic function in numerous malignancies. The Pim1 serine/threonine kinase is associated with multiple cellular functions including proliferation, survival, differentiation, apoptosis, tumorigenesis, immune regulation and inflammation in vertebrate	Selenocysteine (Sec), a rare genetically encoded amino acid with unusual chemical properties, is of great interest for protein engineering. Selenocysteine (SeC) is a naturally available Se-containing amino acid that displays splendid anticancer activities against several human tumors.
Answer	No	Yes

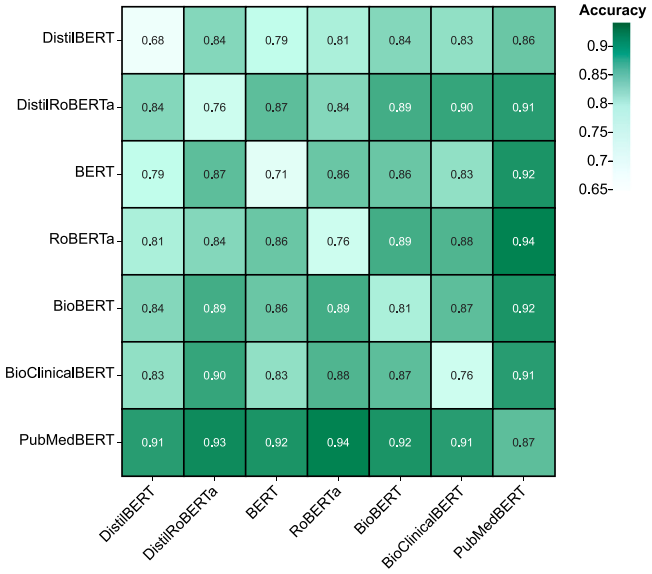


Fig. 1. Oracle performance with different sets of general-purpose PLMs complementing each other in terms of accuracy when addressing the biomedical question-answering task. The diagonal cells indicate single general-purpose PLM $|L| = 1$ and the rest show the performance when $|L| = 2$.

where:

$$ACC(q_j, r_j, L_i) = \begin{cases} 1 & a_{q_j} = f_L(q_j, r_{q_j}|L_i) \\ 0 & o.w \end{cases} \quad (2)$$

As mentioned earlier, a_{q_j} and $f_L(q_j, r_{q_j}|L_i)$ represent the true answer and predicted answer to question q_j with regards to the relevant snippet r_j using a PLM L_i. Here, $ACC(q_j, r_j, L_i)$ is a proxy for the *accuracy* of PLM L_i in addressing question q_j and it is 1 when there exists a PLM that is able to answer the question correctly. Otherwise, when there is no PLM in L that can answer question q_j properly, $Acc(q_j, r_j, L_i)$ would be 0. Simply put, oracle score S indicates the percentage of questions that can be answered correctly given at least one PLM from set L and exhibits the potential of PLM set L for answering the question set Q correctly.

Fig. 1 presents the Oracle score for a set of PLMs when experimenting on the BioASQ7b task. The diagonal cells indicate the performance of the PLM when deployed on a standalone basis, and the other bars display the Oracle when integrating the PLM with another PLM through the Oracle. Each cell on the diagonal shows the performance of the general-purpose PLM on the QA task. The subsequent cells show the performance of the Oracle, which is the result of systematically selecting from among two PLMs from the same row and column, i.e., $|L| = 2$. The difference between the diagonal cell and the rest in the column/row is the maximum amount of performance improvement that can be obtained through an Oracle when considering two general-purpose PLMs. For instance, when only employing BERT to encode the question and its relevant snippet, we obtain an accuracy of 0.71 (rounded by two decimal points for clearer visualization); however, when the Oracle considers both BERT and DistilBERT, the accuracy increases by 11.2% reaching 0.79 . Similarly, when BERT and DistilRoBERTa are considered by the Oracle, we observe an improvement of 23.2% with an accuracy of 0.87.

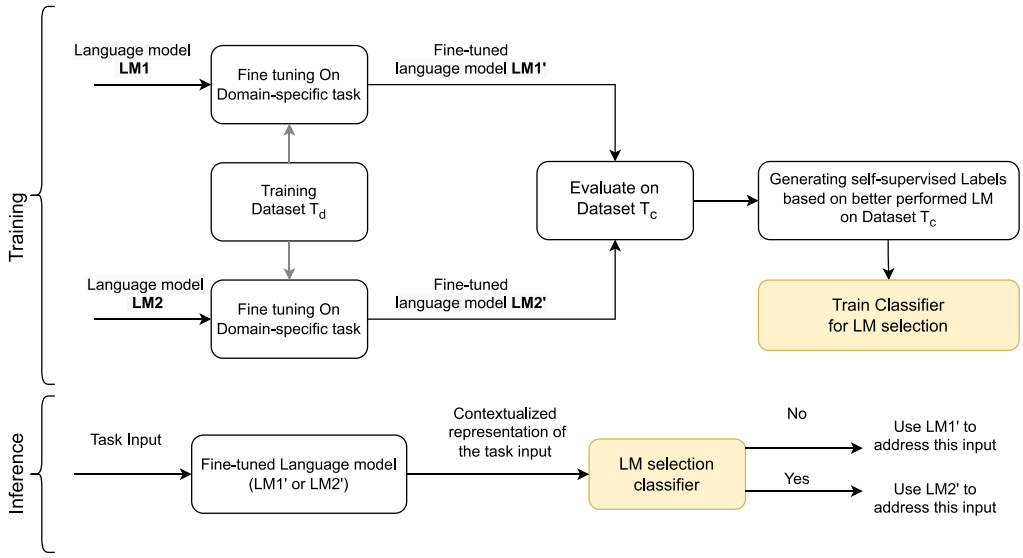
Furthermore, we would like to draw attention to the performance of the domain-specific PLM PubMedBERT on the QA task reported in Fig. 1. As shown in this Figure, the accuracy of PubMedBERT is 0.871. We observe that when this domain-specific PLM is considered by the Oracle in collaboration with other general-purpose PLMs, its performance increases. For instance, when PubMedBERT and RoBERTa are integrated through the Oracle, we can observe an increase of 8.2% over PubMedBERT showing an overall accuracy of 0.94.

In addition to a suite of general-purpose language models (DistilBERT, BERT, DistilRoBERTa, and RoBERTa), our repertoire includes two mixed-domain language models. These models were initially based on the BERT architecture and underwent further pre-training using domain-specific data from other sources including PubMed and the MIMIC dataset [25]. As depicted in Fig. 1, the oracle demonstrates the potential for significantly enhancing the performance of mixed-domain pre-trained language models through a systematic selection of the optimal model for each instance. For example, while BioBERT achieves a standalone accuracy of 0.81, we observe that by selectively choosing the appropriate pre-trained language model between BioBERT and DistilRoBERTa on a per-question basis, the performance can be improved up to 0.89.

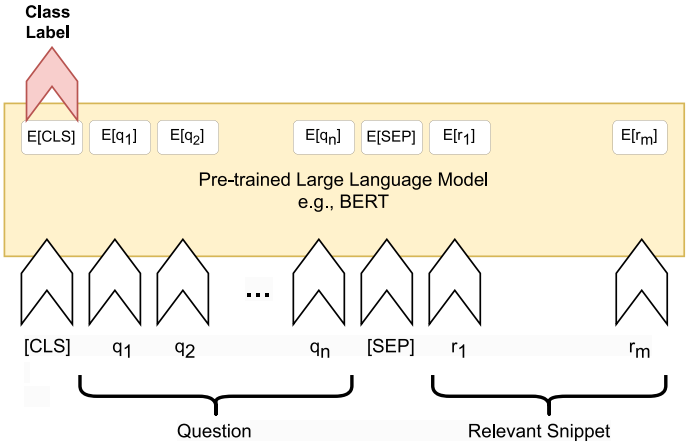
Overall, we make the following observations:

1. The systematic selection between different general-purpose PLMs through an Oracle has the potential to improve the performance of each individual general-purpose PLM and show competitive performance to domain-specific PLMs;
2. The systematic selection between different mixed-domain and general-purpose PLMs through an Oracle has the potential to improve the performance of each individual mixed-domain PLM and has the potential to show superior performance to pure domain-specific PLMs;
3. The integration of domain-specific PLMs with general-purpose ones will lead to increased performance over that of the domain-specific PLMs leading to state-of-the-art performance.

Based on the above two observations, we hypothesize that while single general-purpose language models are not capable of showing competitive performance to domain-specific ones, there is potential room for improving the performance of general-purpose language models on domain-specific tasks by leveraging the synergy between them. In the following, we propose a systematic methodology to leverage this complementary behaviour between general-purpose language models to leverage the best out of them and be able to increase their performance on domain-specific tasks. Such an approach would reduce the need for domain-specific language models.



(a) The overview of our proposed methodology for selecting appropriate general-purpose PLMs.



(b) Overview of fine-tuning pre-trained large language models for exact biomedical question answering task where q_i and r_i represent the i_th token of question q and relevant snippet r respectively.

Fig. 2. Overview of the proposed integration process and model architecture.

5. Self-supervised PLM selection

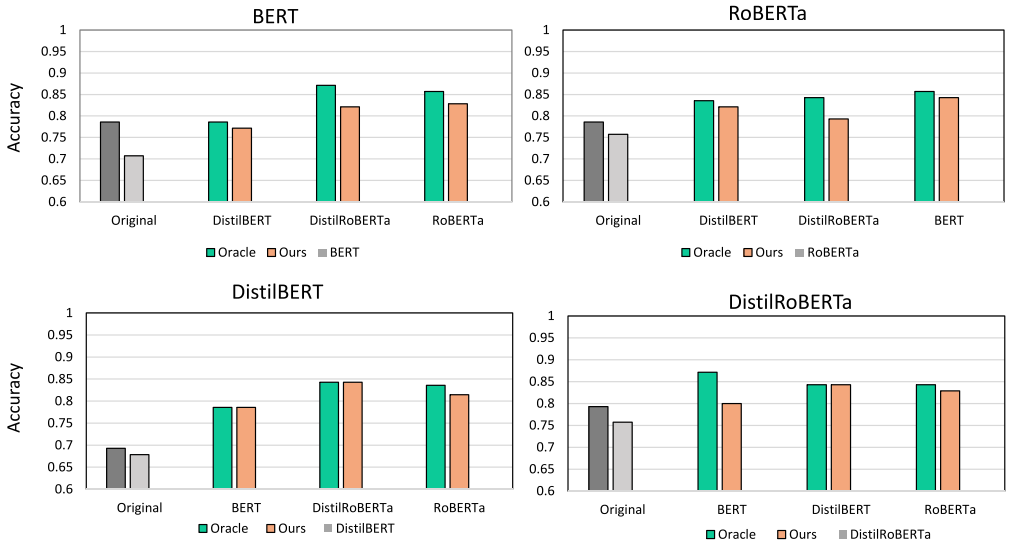
Our observations show that an ideal selection between two general-purpose PLMs through an Oracle has the potential to improve the performance of each PLM and the selection between a general purpose and a domain-specific PLM could potentially outperform the current state of the art on biomedical QA task. Here, our objective is to show whether it would be possible to estimate the *Oracle*. We refer to the task of estimating the Oracle as *PLM Selection* and define it as follows:

Given a question q and its a relevant context r, its exact answer $a \in \{yes = 1, no = 1\}$, and a set of PLMs $S = \{L_1, L_2, \ldots, L_n\}$, the PLM Selection task aims to select $L_{Optimal} \in S$, which would lead to the highest performance when answering q, as:

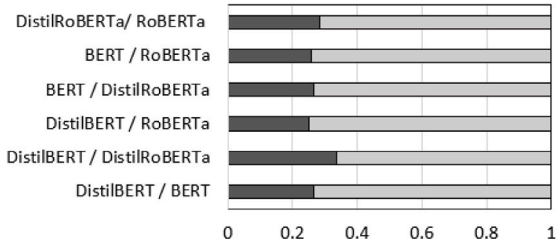
$$f_L(q, r|L_{Optimal}) = \min_{\forall L_i \in S} \{|a - f_L(q, r|L_i)|\} \quad (3)$$

where $f_L(q, r|L)$ is the predicted answer for question q and its relevant snippet r when encoded by PLM L.

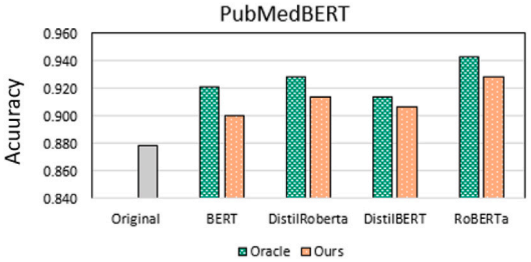
In order to select $L_{Optimal}$, the PLM that is most likely to answer the question correctly, from a set of language models L, we estimate the likelihood of a PLM being more successful on any given subset of questions. To this end, we aim to build a classifier which predicts the suitable language model on per input basis. To train such a classifier that routes each sample to a language model that can address the question, we first split our training set into two portions, T_d and T_c. Given $|L| = 2$ i.e., $L = \{L_1, L_2\}$ and without loss of generality, we fine-tune PLMs L_1 and L_2 on our domain-specific task on the portion of our train set, namely dataset T_d, to obtain fine-tuned L'_1 and L'_2, respectively. Then, based on L'_1 and L'_2, we assess questions from the other unseen portion of the train data (Dataset T_c which has been separated for training the classifier) where $T_d \cap T_c = \emptyset$. We evaluate L'_1 and L'_2 on dataset T_c to detect which questions can be answered correctly with each of the fine-tuned language models. While evaluating, we generate self-supervised labels on which PLM succeeded in answering the question correctly. Particularly, given L'_1 and L'_2, a set of questions Q, their relevant snippets R and answers A, we generate



(a) Result of our proposed approach in terms of Accuracy on BioASQ7b test set.



(b) Percentage of predicted classes in the trained classifier. In each row, the darker bar represents the first language model (on the left) and the light grey bar presents the second language model.



(c) Cooperation of PubMedBERT as domain-specific LM with general purpose LMs using our proposed strategy. The grey bar shows the accuracy of the PubMedBERT standalone. The green bars show the performance based on the proposed selection strategy between PubMedBERT and other general-purpose language models. The orange bars show the performance of the proposed selection strategy between PubMedBERT and the general-purpose language models.

Fig. 3. Overview of the results from the experiments.

labels for each task input $\{q, r, a\}$ as follows:

$$LM_{oracle}(q_i, r_i|L) = \begin{cases} 0 & f_L(q_i, r_i|L'_1) = a_i \\ 1 & f_L(q_i, r_i|L'_2) = a_i \end{cases} \quad (4)$$

Here, essentially based on training samples in T_c, we determine which PLM is most suited for answering each question in T_c; hence generating *self-supervised labels*. In case both PLMs answer a question correctly or both fail to answer it correctly, we randomly select one of the PLMs when curating the self-supervised labels.

Based on these self-supervised labels, we train a classifier using the contextualized representation of q and r encoded by the selected language model from L to select the PLM that is more likely to answer each question correctly. More information on the structure of the classifier can be found in Section 6.1. The base PLM for the classifier can be either L'_1 or L'_2. Fig. 2(a) shows the overview of our proposed approach. Once we train the PLM Selection classifier, given a question and its relevant snippet (q, r), we first decide which PLM would be the most relevant for the question to be routed to and answer the question based on the selected PLM.

6. Experiments and results

6.1. Experimental setup

Since the focus of this paper is to investigate the impact of general-purpose PLMs on the biomedical question answering task with different

PLMs, we use the Transformer-based architecture, which is the most common approach and has shown promising results on many downstream tasks [49–51]. Similar to [5], we fine-tune PLMs by first processing the input sequence and performing task-specific transformations such as appending a special instance marker (e.g., $[CLS]$). The transformed input is then tokenized using the PLM vocabulary, and fed into the PLM to generate the contextualized representations of the input. On top of the embedded vector of the input sequence, a task-specific prediction model generates the final output. Task-specific prediction layer parameters are jointly fine-tuned along with the underlying neural language model. More specifically, for fine-tuning PLMs to address the biomedical QA task, the input sequence, i.e., the question Q, as well as the accompanying snippet for each question R, are concatenated with each other through the $[SEP]$ token. Further, a $[CLS]$ token is prepended to the input sequence. As such, the Transformer input takes the form of $[CLS]Q[SEP]R[SEP]$. The $[CLS]$ is then used for the final classification. Finally, we include a linear classification layer with cross-entropy loss to reduce dimensionality. The overview of the described architecture is shown in Fig. 2(b).

We selected the learning rate from $\{1e-5, 3e-5, 5e-5\}$, set batch size to 16 and epoch number from $\{1, 2, \ldots, 20\}$. In practice and similar to what was reported in [5], we observe that the development performance is not sensitive to the hyper-parameter selection.

Further, we consider BioASQ8b train set as our T_c dataset to train the classifier. Therefore, we evaluate the performance of PLMs on the BioASQ8b train set and train the second step of our proposed approach, i.e., the classifier. For each pair of PLMs, we use the lighter model as the trained classifier to avoid extra computational overhead. We note that the number of parameters for the models under experiments of this paper is as follows: BERT-base-uncased[4] (110M), DistilBERT-base-uncased[5] (66M), Roberta-base[6] (125M), DistilRobBERTa-base[7] (82M), Bio_ClinicalBERT[8] (110M), BioBERT[9] (110M) and PubMedBERT[10] (110M). In the case of deciding the LLM between BERT and PubMedBERT, we employ BERT as the classifier since it is a general-purpose language model and it is not dependent on the domain. We leave further explorations on the choice of appropriate PLM for the classifier for future works.

BioASQ7b and BioASQ8b have 670 and 884 yes/no questions in their train set, respectively. We report the end-to-end performance of our proposed approach as well as the baselines on BioASQ7b test set, which includes 140 questions. Finally, we use the Hugging Face public releases[11] of `BERT` [4], `RoBERTa` [52], `DistilBERT` and `DistilRoBERTa` [53] to construct our set of general-purpose PLMs.

6.2. Results and findings

Fig. 3(a) shows the results of our experiments, where the title of each sub-figure represents the PLM that was used in the second step of the pipeline. For example, considering the `BERT` sub-figure in Fig. 3(a), the grey bars represent the performance of `BERT` under a standalone condition, i.e., $L = \{\texttt{BERT}\}$. The three green bars in this sub-figure show the performance of the Oracle when selecting the PLM from $L = \{\texttt{BERT}, \texttt{DistilBERT}\}$, $L = \{\texttt{BERT}, \texttt{DistilRoBERTa}\}$ and $L = \{\texttt{BERT}, \texttt{RoBERTa}\}$, respectively. Furthermore, the orange bars illustrate our proposed approach when selecting PLMs between `BERT` and each of the mentioned PLMs on a per-question basis. A similar explanation can be adapted to other sub-figures in Fig. 3(a).

As shown in each of the sub-figures, selecting PLMs from the set of two general-purpose PLMs instead of using one general-purpose PLM to encode all the questions and relevant snippets leads to performance improvements over the initial general-purpose PLM. One of the most significant boosts happens with lighter models such as `DistilBERT`. While `DistilBERT` shows a relatively lower performance when used in isolation, it shows great complementary behaviour with other PLMs and leads to notable performance improvement when used within the context of our proposed selection strategy. This means that while `DistilBERT` fails to show a good performance on all of the questions, there are specific subsets of questions that can be answered with `DistilBERT` very well. Hence, as shown in Table 2, when choosing from two of the lightest PLMs i.e., `DistilBERT` and `DistilRoBERTa`, we observe one of the largest performance improvements, i.e., an increase of 23.16% from 0.679 to 0.836.

Table 2 shows the performance of the Oracle in contrast to our proposed approach in terms of accuracy and percentage of improvement compared to the base PLM. As shown in Table 2, while all the PLMs showed statistically significant improvements measured based on a paired t-test (p-value < 0.05), `DistilBERT` enjoyed the largest performance improvement when incorporated through our selection strategy with other PLMs, namely 15.79%, 23.16% and 20.0% improvement when used in conjunction with `BERT`, `DistilRoBERTa` and `RoBERTa`, respectively.

Overall, the results in Table 2 and Fig. 3(a), shed light on the research questions we raised and show that it is possible to adopt a selection strategy that jointly utilizes different general-purpose PLMs and boosts the performance of each general-purpose PLM in addressing domain-specific question answering task. In fact, while the state-of-the-art domain-specific PLM, `PubMedBERT` shows an accuracy of 0.878, the integration of `DistilRoBERTa` and `DistilBERT` through our proposed selection strategy shows the overall performance of 0.836, which is competitive to `PubMedBERT` and over 10% more than the best-performing general-purpose PLM.

Additionally, we are interested in exploring the distribution of predicted classes between the set of general-purpose language models. This would show how balanced the model's predictions are with regard to different classes of language models. To investigate the distribution of predicted classes between two PLMs used in the proposed language model selection strategy, we report the percentage of predicted classes in Fig. 3(b) between each pair of the two language models. On average, on the six pairs of general purpose language models presented in Fig. 3(b), (e.g., {`DistilRoBERTa`, `RoBERTa`}, {`BERT`,`RoBERTa`}, ...), the distribution of predicted classes is 27% (dark grey) to 73% (light grey bars) in Fig. 3(b). As shown in this Figure, while both models had contributions towards the gold labels, the distribution of model choices is rather biased towards the model represented by the lighter bar, which is often the larger model. For example, `RoBERTa` models have better performance than `BERT` models and full-fledged models usually have better performance compared to their distilled versions.

For example, when $L = \{\texttt{DistilBERT}, \texttt{DistilRoBERTa}\}$ are used in combination with each other through the selection strategy, we observe that in 34% of the cases, the correct answer is selected by `DistilBERT` and in 66% of the cases, the answers were selected by `DistilRoBERTa`. It is interesting that this high complementary behaviour between the two language models also leads to one of the highest improvements over both of the initial PLMs, i.e., more than 10% and 23% over the performance of `DistilBERT` and `DistilRoBERTa`, respectively.

In Table 3, we show two examples where `DistilBERT` and `DistilRoBERTa` were both successful in answering *only one of the two questions* correctly. For example, the correct answer to the question 'Is Pim-1 a protein phosphatase?' is 'No'. `DistilRoBERTa` was able to predict the answer correctly; whereas, `DistilBERT` incorrectly

[4] https://huggingface.co/bert-base-uncased
[5] https://huggingface.co/distilbert-base-uncased
[6] https://huggingface.co/roberta-base
[7] https://huggingface.co/distilroberta-base
[8] https://huggingface.co/emilyalsentzer/Bio_ClinicalBERT
[9] https://huggingface.co/dmis-lab/biobert-v1.1
[10] https://huggingface.co/microsoft/BiomedNLP-PubMedBERT-base-uncased-abstract-fulltext
[11] https://huggingface.co/models

Table 2
Result of our proposed approach in terms of percentage of improvement. The percentage is rounded to 2 decimal places based on the actual accuracy numbers.

BERT	Oracle	Ours	% Δ
Original	–	0.707	–
DistilBERT	0.786	0.771	9.09%
DistilRoBERTa	0.871	0.821	16.16%
RoBERTa	0.857	0.829	17.17%

RoBERTa	Oracle	Ours	% Δ
Original	–	0.757	–
DistilBERT	0.836	0.821	8.49%
DistilRoBERTa	0.843	0.793	4.72%
BERT	0.857	0.843	11.32%

DistilBERT	Oracle	Ours	% Δ
Original	–	0.679	–
BERT	0.786	0.786	15.79%
DistilRoBERTa	0.843	0.836	23.16%
RoBERTa	0.836	0.814	20.00%

DistilRoBERTa	Oracle	Ours	% Δ
Original	–	0.757	–
BERT	0.871	0.8	5.66%
DistilBERT	0.843	0.836	10.38%
RoBERTa	0.843	0.829	9.43%

Table 3
Sample Test Questions and Responses by different LLMs and Their Interpolation.

(a) Sample questions that could not be answered correctly with neither of the general-purpose language models but were successfully answered by our approach.

Question	DistilBERT answer	DistilRoBERTa answer	Our method answer	True answer
-Is Pim-1 a protein phosphatase?	Yes	No	No	No
-Is celecoxib effective for treatment of amyotrophic lateral sclerosis?	No	Yes	No	No

(b) Examples where the answer to the question is "No" but the end-to-end results of our proposed methodology of $L = \{PubMedBERT, L_g\}$ where L_g is a general purpose language model sometimes failed to predict correctly. $L_g \in$ {DistilBERT, DistilRoBERTa, BERT, RoBERTa}.

Question	Answer	Predicted answer with $L = \{PubMedBERT, L_g\}$			
		DistilBERT	DistilRoBERTa	BERT	RoBERTa
Is lithium effective for treatment of amyotrophic lateral sclerosis?	No	No	No	Yes	No
Does Groucho related gene 5 (GRG5) have a role only in late development?	No	No	Yes	No	No
Has the protein SIRT2 been associated to cervical cancer?	No	Yes	Yes	No	Yes
Is the NLM medical text indexer (MTI) still useful and relevant?	No	No	No	No	Yes
Are artificial blood cells available?	No	Yes	Yes	No	Yes
Are cardenolides inhibitors of Na+/K+ ATPase?	No	No	Yes	No	No
Is myc a tumour suppressor gene?	No	Yes	No	Yes	Yes

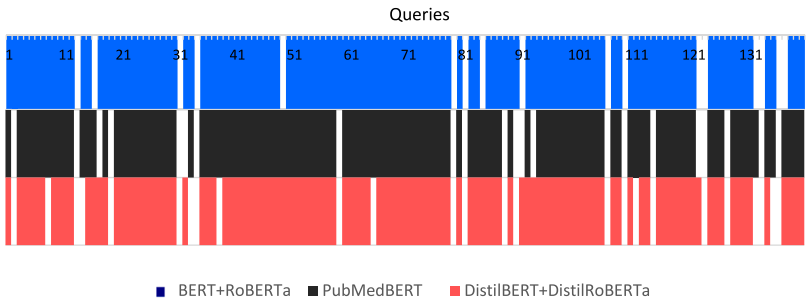
predicted the answer to this question as 'Yes'. For another question 'Is celecoxib effective for treatment of amyotrophic lateral sclerosis? ', whose correct answer is 'No', the opposite happens. In other words, DistilBERT predicted the answer correctly as 'No', while DistilRoBERTa was not successful in answering the question properly. If any of the two PLMs is used to answer both questions, we would not be able to answer both correctly. However, using our proposed methodology, we were able to select the PLM which is able to answer each question correctly, i.e., for the first question, we select DistilRoBERTa to answer the question and for the second one, we employ DistilBERT. As a result, we are able to answer both questions correctly.

We conducted a detailed analysis of the performance of the state-of-the-art domain-specific PLM and our proposed approach. Fig. 4(a) illustrates our findings, showing each question in our test set individually. The filled bars indicate that the question was answered correctly using the proposed approach, while the non-filled bars indicate that the question was answered mistakenly by that method. We present two of our computationally least expensive combinations of PLMs, i.e., DistilBERT and DistilRoBERTa, as well as the computationally most expensive pairs under the experiment, i.e., BERT and RoBERTa. As shown in the Figure, even when using distilled and light PLMs, our proposed approach can accurately answer the majority of the questions that PubMEDBERT answered correctly. In other words, the general-purpose models can successfully capture critical domain-specific features.

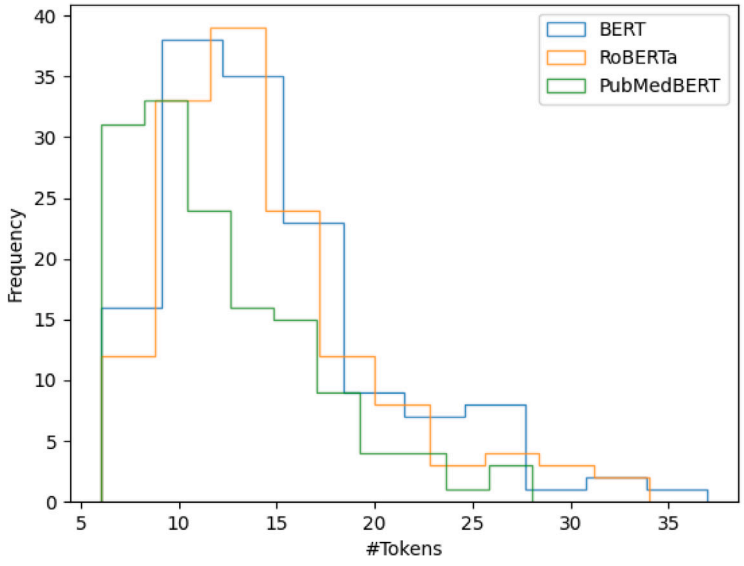
Finally, we investigate the potential benefits of our proposed selection strategy for both mixed-domain and domain-specific PLMs. We conduct similar experiments, selecting between the state-of-the-art biomedical domain-specific PLM, PubMedBERT, and mixed-domain PLMs represented by BioBERT and BioClinicalBERT, utilizing general-purpose PLMs including BERT, RoBERTa, DistilBERT, and DistilRoBERTa. The results are presented in Table 4. As shown, both mixed-domain PLMs, BioBERT and BioClinicalBERT, exhibit performance improvements when combined with general-purpose language models. However, as anticipated, the percentage of improvement compared to the original model is slightly lower than when solely employing the PLMs from general-purpose language models. In addition in Fig. 3(c), we observe that our proposed PLM selection strategy boosts the performance of PubMedBERT as the state-of-the-art domain-specific PLM. Overall, the selection between PubMedBERT with RoBERTa leads to over 5.69% of improvement by lifting up the performance from 0.879 to 0.929.

We also conducted a detailed analysis of the performance of our approach, which combines a state-of-the-art domain-specific PLM with a general-purpose PLM. Upon examining the incorrect results, we discovered that over 85% of errors occurred when the model mistakenly predicted an answer as "yes" instead of "no". In Table 3, we present examples where the correct answer to all questions is "no", but our approach failed to predict them accurately. For example, as shown in Table 3, DistilRoBERTa wrongly predict the answer as "Yes" in 4 out of the 7 examples. We hypothesize that this could be due to the unbalanced training data that the question-answering model was exposed to, which may have resulted in a bias towards predicting the "yes" label.

One reason why general-purpose language models may fail to perform as well as domain-specific ones is due to out-of-vocabulary problems. In [5], it was found that training a domain-specific language model from scratch has the advantage of having in-domain vocabulary. For example, when using the original vocabulary of BERT, some



(a) Questions addressed correctly using $L = \{\texttt{BERT}, \texttt{RoBERTa}\}$ (first row) and Distilled version of them $L = \{\texttt{DistilBERT}, \texttt{DistilRoBERTa}\}$ (last row) vs the domain-specific language model `PubMedBERT` (middle row). The filled bars indicate the questions that were answered correctly using the proposed approach, while the white bars demonstrate questions which were answered mistakenly by that method.



(b) Histogram of number of tokens for questions in the test set of experiment based on `BERT`, `RoBERTa` and `PubMedBERT`.

Fig. 4. Detailed insights on model performance.

Table 4
Results of the proposed selection strategy over domain-specific language model (`PubMedBERT`) and mixed-domain PLMs (`BioBERT` and `BioClinicalBERT` with different general-purpose PLMs $\in$ {`BERT`,`DistilRoBERTa`,`DistilBERT`,`RoBERTa`} in terms of accuracy and percentage of improvement.

	PubMedBERT			BioBERT			BioClinicalBERT		
	Oracle	Ours	% Δ	Oracle	Ours	% Δ	Oracle	Ours	% Δ
Original	–	0.879	–	–	0.814	–	–	0.757	–
BERT	0.921	0.900	2.44%	0.864	0.844	3.72%	0.829	0.821	8.49%
RoBERTa	0.943	0.929	5.69%	0.886	0.857	5.30%	0.879	0.857	13.21%
DistilBERT	0.914	0.907	3.25%	0.843	0.829	1.79%	0.829	0.814	7.55%
DistilRoberta	0.929	0.914	4.07%	0.886	0.879	7.93%	0.901	0.893	17.92%

biomedical terms such as "Naloxon" as a drug name or "Acetyltransferase" as a gene name would be broken down into na-lo-xon-e and ace-ty-lt-ran-sf-eras-e, respectively. Here, we compared the number of tokens in the BioASQ7b questions when tokenizing with BERT, RoBERTa, and PubMedBERT to see if general-purpose language models do in fact break down domain-specific terms. We hypothesize that breaking down tokens might hurt performance since the semantics of the terms would be compromised. In Fig. 4(b), we plotted the histogram of the number of tokens based on each LM tokenizer. As shown in this Figure, PubMedBERT has the lowest number of tokens, i.e., its tokenizer does not break down domain-specific terms as much as BERT and RoBERTa. We note that the mixed-domain PLMs used in this study all share their vocabulary with BERT since they have been initialized with this language model and then continued pretraining on domain-specific data. This analysis confirms the observation in [5], which suggests that the out-of-vocabulary problem could be one of the reasons why domain-specific language models perform better under a standalone setting.

We conclude that not only does the systematic selection between general purpose PLMs leads to notable performance improvement and competitive results to domain-specific PLMs, but also their systematic integration through our selection strategy with domain-specific PLMs leads to improvements over state-of-the-art results.

7. Computational analysis

In this section, we undertake a comprehensive computational analysis to explore the trade-offs and considerations associated with our proposed approach. We begin by examining the computational expenses involved in training mixed-domain or domain-specific language models and compare them against our strategy. As highlighted earlier in the introduction, training a domain-specific PLM from scratch presents significant computational challenges. For instance, the biomedical PLM PubMedBERT, trained on PubMed abstracts, necessitated the use of powerful hardware resources, including 16 v100 GPUs on a DGX-2 machine, with a total cost exceeding 400K USD. This cost acts as a notable barrier, limiting access for many researchers and institutions. Furthermore, the time required for training domain-specific PLMs is substantial. PubMedBERT, as an example, demanded five days of training on the aforementioned computational setup, further exacerbating the challenges associated with developing and deploying domain-specific PLMs.

In terms of computational analysis of our proposed approach, it may appear that training two distinct language models introduces additional computational overhead. However, we emphasize that fine-tuning for downstream tasks is notably less resource-intensive, both in terms of time and computational requirements, compared to training a language model from scratch. For instance, fine-tuning BERT for question answering in our experiments utilized a single RTX-3090 GPU, with training times averaging less than 10 min due to limited available training data for the task. Consequently, the computational demands for this process were modest, with no need for high GPU numbers or extensive memory resources.

In terms of storage considerations, our approach requires the storage of two fine-tuned language models instead of one. Despite this, the relatively modest size of the models used in this paper, all consuming less than 500MB of storage, renders the storage impact negligible. It is worth noting that while this may not pose a significant concern for smaller language models, the storage implications could become more relevant when employing larger, higher-parameter models.

Turning to inference time, the average inference time for answering questions using any of the seven fine-tuned language models in our experiments, executed on an RTX-3090 GPU with 24 GB memory, averaged 1.9 ms. Given sufficient GPU availability and time constraints, we suggest parallel execution of both fine-tuned models to minimize computational delay. Additionally, a classification step, which averaged 1.4 ms across all language models, adds minimal computational time for decoding and classification. Thus, if parallel execution is possible, no additional computational time is introduced, limited by the more resource-intensive answering time of 1.9 ms. However, in cases of resource constraints, a sequential approach involving classification followed by answering with the selected PLM would result in an additional 1.4 ms.

In summary, the decision to train additional language models depends on the specific application and computational constraints. This analysis provides insights into optimizing the use of language models, helping determine when to train new models or utilize existing ones, thus fostering efficiency and sustainability.

8. Future works and concluding remarks

Domain-specific PLMs have shown strong performance on downstream domain-specific tasks. However, we argue that training domain-specific PLMs is quite expensive in terms of time and computation. Moreover, some domains suffer from a lack of abundant publicly available corpora for training PLMs. Therefore, in this paper, we study whether it would be possible to employ general-purpose PLMs on domain-specific tasks. We show that general-purpose PLMs have the potential to complement each other and exhibit synergistic behaviour. On this basis, we propose a self-supervised approach to integrate different general-purpose PLMs. We show that while general-purpose PLMs might not be able to outperform domain-specific PLMs, they can complement each others' performance leading to better overall performance. We conclude that selecting an appropriate PLM is a lightweight strategy for using general-purpose PLMs that can show competitive performance to domain-specific PLMs, yet does not require as many resources for being trained.

To the best of our knowledge, this is the first work which investigates the synergy between different PLMs specifically in the context of the biomedical QA task. Thus, we hope that this work will open up many future avenues for IR and NLP communities to build on general-purpose PLMs, which are computationally less demanding and can be considered *greener* and more *environmentally-friendly* models compared to domain-specific PLMs that require significant resources to be trained.

In our future work, we are interested in addressing the limitations of the current work by exploring the following directions:

- While end-to-end performance is limited to the degree of complementarity of the integrated models, in the experiments in this paper, we show that all pairs of investigated language models have a promising degree of complementarity, which lead to significant improvement when leveraging our proposed strategy for this specific task. However, we believe that applying a similar methodology to other domain-specific PLMs and tasks could benefit the community by showing that our approach is generalizable.
- As a pioneering study in strategic PLM selection, this research presents early findings, leaving ample room for further exploration and generalization. Initially, we acknowledge the need to delve into the scalability of our proposed approach. Although our methodology exhibits adaptability to the inclusion of more than two PLMs simultaneously, it is conceivable that increased complexity in classification tasks could demand larger training datasets. Thus, a valuable avenue for future research lies in empirically gauging the potential of extending our approach to incorporate multiple PLMs. Additionally, we recognize that the experiments in this work were confined to binary question answering, which raises valid concerns about the applicability of our findings across broader contexts. To address this, we envision broadening the scope of our systematic PLM selection technique beyond binary question answering. This expansion could encompass a variety of tasks, including question answering, retrieval, named entity recognition, document classification, and other downstream tasks pertinent to information retrieval and natural language processing. By doing so, we aim to establish the versatility and practicality of our approach across a spectrum of real-world applications.
- Although our proposed strategy is clearly less expensive than training a PLM from scratch, further investigation into the added complexity and computational overhead would be useful to quantify this advantage;
- Systematically investigating why different PLMs complement each other and what are the most important factors affecting this complementary behaviour, would provide beneficial insights to the community for further capitalization on this synergy for other downstream applications.

CRediT authorship contribution statement

Negar Arabzadeh: Conceptualization, Formal analysis, Investigation, Software, Validation, Writing – original draft. **Ebrahim Bagheri:** Conceptualization, Investigation, Writing – review & editing, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] F. Zhu, W. Lei, C. Wang, J. Zheng, S. Poria, T.-S. Chua, Retrieving and reading: A comprehensive survey on open-domain question answering, 2021, arXiv preprint arXiv:2101.00774.

[2] H. Zamani, J.R. Trippas, J. Dalton, F. Radlinski, Conversational information seeking, 2022, arXiv preprint arXiv:2201.08808.

[3] A. Yates, R. Nogueira, J. Lin, Pretrained transformers for text ranking: BERT and beyond, in: Proceedings of the 14th ACM International Conference on Web Search and Data Mining, 2021, pp. 1154–1156.

[4] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, 2018, arXiv preprint arXiv:1810.04805.

[5] Y. Gu, R. Tinn, H. Cheng, M. Lucas, N. Usuyama, X. Liu, T. Naumann, J. Gao, H. Poon, Domain-specific language model pretraining for biomedical natural language processing, ACM Trans. Comput. Healthc. (HEALTH) 3 (1) (2021) 1–23.

[6] Y. Xu, X. Liu, Y. Shen, J. Liu, J. Gao, Multi-task learning with sample re-weighting for machine reading comprehension, 2018, arXiv preprint arXiv:1809.06963.

[7] X. Liu, J. Gao, X. He, L. Deng, K. Duh, Y.-Y. Wang, Representation learning using multi-task deep neural networks for semantic classification and information retrieval, 2015.

[8] S. Ruder, An overview of multi-task learning in deep neural networks, 2017, arXiv preprint arXiv:1706.05098.

[9] A. Axelrod, X. He, J. Gao, Domain adaptation via pseudo in-domain data selection, in: Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, 2011, pp. 355–362.

[10] K. Weiss, T.M. Khoshgoftaar, D. Wang, A survey of transfer learning, J. Big data 3 (1) (2016) 1–40.

[11] S. Niu, Y. Liu, J. Wang, H. Song, A decade survey of transfer learning (2010–2020), IEEE Trans. Artif. Intell. 1 (2) (2020) 151–166, http://dx.doi.org/10.1109/TAI.2021.3054609.

[12] I. Beltagy, K. Lo, A. Cohan, SciBERT: A pretrained language model for scientific text, in: K. Inui, J. Jiang, V. Ng, X. Wan (Eds.), Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019, Association for Computational Linguistics, 2019, pp. 3613–3618, http://dx.doi.org/10.18653/v1/D19-1371.

[13] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C.H. So, J. Kang, BioBERT: a pre-trained biomedical language representation model for biomedical text mining, Bioinformatics 36 (4) (2019) 1234–1240, http://dx.doi.org/10.1093/bioinformatics/btz682.

[14] A. Trewartha, N. Walker, H. Huo, S. Lee, K. Cruse, J. Dagdelen, A. Dunn, K.A. Persson, G. Ceder, A. Jain, Quantifying the advantage of domain-specific pre-training on named entity recognition tasks in materials science, Patterns 3 (4) (2022) 100488.

[15] D. Hristovski, D. Dinevski, A. Kastrin, T.C. Rindflesch, Biomedical question answering using semantic relations, BMC Bioinform. 16 (1) (2015) 1–14.

[16] W. Yoon, R. Jackson, A. Lagerberg, J. Kang, Sequence tagging for biomedical extractive question answering, Bioinformatics 38 (15) (2022) 3794–3801.

[17] M. Sarrouti, S.O. El Alaoui, A passage retrieval method based on probabilistic information retrieval model and UMLS concepts in biomedical question answering, J. Biomed. Inform. 68 (2017) 96–103.

[18] Y. Li, R.M. Wehbe, F.S. Ahmad, H. Wang, Y. Luo, A comparative study of pretrained language models for long clinical text, J. Am. Med. Inform. Assoc. 30 (2) (2023) 340–347.

[19] Y. Li, S. Long, Z. Yang, H. Weng, K. Zeng, Z. Huang, F.L. Wang, T. Hao, A Bi-level representation learning model for medical visual question answering, J. Biomed. Inform. 134 (2022) 104183.

[20] Q. Jin, Z. Yuan, G. Xiong, Q. Yu, C. Tan, M. Chen, S. Huang, X. Liu, S. Yu, Biomedical question answering: A comprehensive review, 2021.

[21] Y. Peng, S. Yan, Z. Lu, Transfer learning in biomedical natural language processing: An evaluation of BERT and ELMo on ten benchmarking datasets, CoRR abs/1906.05474, 2019, arXiv:1906.05474.

[22] M. Sarrouti, S.O. El Alaoui, SemBioNLQA: A semantic biomedical question answering system for retrieving exact and ideal answers to natural language questions, Artif. Intell. Med. 102 (2020) 101767.

[23] A. Wen, M.Y. Elwazir, S. Moon, J. Fan, Adapting and evaluating a deep learning language model for clinical why-question answering, JAMIA Open 3 (1) (2020) 16–20.

[24] E. Alsentzer, J.R. Murphy, W. Boag, W. Weng, D. Jin, T. Naumann, M.B.A. McDermott, Publicly available clinical BERT embeddings, CoRR abs/1904.03323, 2019, arXiv:1904.03323.

[25] A.E. Johnson, T.J. Pollard, L. Shen, L.-w.H. Lehman, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. Anthony Celi, R.G. Mark, MIMIC-III, a freely accessible critical care database, Sci. Data 3 (1) (2016) 1–9.

[26] A. Nentidis, A. Krithara, K. Bougiatiotis, M. Krallinger, C. Rodriguez-Penagos, M. Villegas, G. Paliouras, Overview of bioasq 2020: The eighth bioasq challenge on large-scale biomedical semantic indexing and question answering, in: International Conference of the Cross-Language Evaluation Forum for European Languages, Springer, 2020.

[27] M. Lapata, F. Keller, Web-based models for natural language processing, ACM Trans. Speech Lang. Process. (TSLP) 2 (1) (2005) 3–es.

[28] T. Huang, Q. She, J. Zhang, BoostingBERT: Integrating multi-class boosting into BERT for NLP tasks, 2020, arXiv preprint arXiv:2009.05959.

[29] I. Lauriola, A. Lavelli, F. Aiolli, An introduction to deep learning in natural language processing: Models, techniques, and tools, Neurocomputing 470 (2022) 443–456.

[30] A. Finch, E. Sumita, Dynamic model interpolation for statistical machine translation, in: Proceedings of the Third Workshop on Statistical Machine Translation, 2008, pp. 208–215.

[31] R. Sennrich, Perplexity Minimization for Translation Model Domain Adaptation in Statistical Machine Translation, Association For Computational Linguistics, 2012.

[32] A. Jindal, A.G. Chowdhury, A. Didolkar, D. Jin, R. Sawhney, R. Shah, Augmenting NLP models using latent feature interpolations, in: Proceedings of the 28th International Conference on Computational Linguistics, 2020, pp. 6931–6936.

[33] B. Bakker, T. Heskes, Clustering ensembles of neural network models, Neural Netw. 16 (2) (2003) 261–269.

[34] M.P. Perrone, L.N. Cooper, When networks disagree: Ensemble methods for hybrid neural networks, in: How We Learn; How We Remember: Toward an Understanding of Brain and Neural Systems: Selected Papers of Leon N Cooper, World Scientific, 1995, pp. 342–358.

[35] M. Kanakaraj, R.M.R. Guddeti, Performance analysis of ensemble methods on Twitter sentiment analysis using NLP techniques, in: Proceedings of the 2015 IEEE 9th International Conference on Semantic Computing, IEEE ICSC 2015, IEEE, 2015, pp. 169–170.

[36] W. Zhang, J. Jiang, Y. Shao, B. Cui, Snapshot boosting: a fast ensemble framework for deep neural networks, Sci. China Inf. Sci. 63 (2020) 1–12.

[37] M.A. Ganaie, M. Hu, A. Malik, M. Tanveer, P. Suganthan, Ensemble deep learning: A review, Eng. Appl. Artif. Intell. 115 (2022) 105151.

[38] S. Wang, S. Zhuang, G. Zuccon, Bert-based dense retrievers require interpolation with bm25 for effective passage retrieval, in: Proceedings of the 2021 ACM SIGIR International Conference on Theory of Information Retrieval, 2021, pp. 317–324.

[39] A. Abolghasemi, A. Askari, S. Verberne, On the interpolation of contextualized term-based ranking with bm25 for query-by-example retrieval, in: Proceedings of the 2022 ACM SIGIR International Conference on Theory of Information Retrieval, 2022, pp. 161–170.

[40] A. Askari, A. Abolghasemi, G. Pasi, W. Kraaij, S. Verberne, Injecting the BM25 score as text improves BERT-based re-rankers, in: Advances in Information Retrieval: 45th European Conference on Information Retrieval, ECIR 2023, Dublin, Ireland, April 2–6, 2023, Proceedings, Part I, Springer, 2023, pp. 66–83.

[41] Q. Wu, C.J. Burges, K.M. Svore, J. Gao, Adapting boosting for information retrieval measures, Inf. Retr. 13 (2010) 254–270.

[42] Q. Wu, C.J. Burges, K.M. Svore, J. Gao, Ranking, Boosting, and Model Adaptation, Citeseer, 2008.

[43] N. Arabzadeh, X. Yan, C.L. Clarke, Predicting efficiency/effectiveness trade-offs for dense vs. Sparse retrieval strategy selection, in: Proceedings of the 30th ACM International Conference on Information & Knowledge Management, 2021, pp. 2862–2866.

[44] J. Gao, Q. Wu, C. Burges, K. Svore, Y. Su, N. Khan, S. Shah, H. Zhou, Model adaptation via model interpolation and boosting for web search ranking, 2019, arXiv preprint arXiv:1907.09471.

[45] H. Jin, X. Ning, H. Chen, Z. Yin, Efficient query routing for information retrieval in semantic overlays, in: Proceedings of the 2006 ACM Symposium on Applied Computing, 2006, pp. 1669–1673.

[46] T. Yeferny, K. Arour, Learningpeerselection: A query routing approach for information retrieval in p2p systems, in: 2010 Fifth International Conference on Internet and Web Applications and Services, IEEE, 2010, pp. 235–241.

[47] G. Tsatsaronis, G. Balikas, P. Malakasiotis, I. Partalas, M. Zschunke, M.R. Alvers, D. Weissenborn, A. Krithara, S. Petridis, D. Polychronopoulos, et al., An overview of the BIOASQ large-scale biomedical semantic indexing and question answering competition, BMC Bioinform. 16 (1) (2015) 1–28.

[48] A. Nentidis, K. Bougiatiotis, A. Krithara, G. Paliouras, Results of the seventh edition of the BioASQ challenge, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, 2019, pp. 553–568.
[49] R. Nogueira, K. Cho, Passage re-ranking with BERT, CoRR abs/1901.04085, 2019, arXiv:1901.04085.
[50] C. Sun, X. Qiu, Y. Xu, X. Huang, How to fine-tune bert for text classification? in: China National Conference on Chinese Computational Linguistics, Springer, 2019.
[51] S. González-Carvajal, E.C. Garrido-Merchán, Comparing BERT against traditional machine learning text classification, 2020, arXiv preprint arXiv:2005.13012.
[52] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, RoBERTa: A robustly optimized BERT pretraining approach, CoRR abs/1907.11692, 2019, arXiv:1907.11692.
[53] V. Sanh, L. Debut, J. Chaumond, T. Wolf, DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter, 2019, arXiv preprint arXiv:1910.01108.