Proceedings of the 13th Conference on Language Resources and Evaluation (LREC 2022), pages 363–373
Marseille, 20-25 June 2022

KIMERA: Injecting Domain Knowledge into Vacant Transformer Heads

Benjamin Winter*, Alexei Figueroa*, Alexander Löser, Felix A. Gers, Amy Siu
Berliner Hochschule für Technik
{Benjamin.Winter, afigueroa, aloeser, gers, asiu}@bht-berlin.de
*Both authors contributed equally.

Abstract

Training transformer language models requires vast amounts of text and computational resources. This drastically limits the usage of these models in niche domains for which they are not optimized, or where domain-specific training data is scarce. We focus here on the clinical domain because of its limited access to training data in common tasks, while structured ontological data is often readily available. Recent observations in model compression of transformer models show optimization potential in improving the representation capacity of attention heads. We propose KIMERA (**K**nowledge **I**njection via **M**ask **E**nforced **R**etraining of **A**ttention) for detecting, retraining and instilling attention heads with complementary structured domain knowledge. Our novel multi-task training scheme effectively identifies and targets individual attention heads that are least useful for a given downstream task and optimizes their representation with information from structured data. KIMERA generalizes well, thereby building the basis for an efficient fine-tuning. KIMERA achieves significant performance boosts on seven datasets in the medical domain in Information Retrieval and Clinical Outcome Prediction settings. We apply KIMERA to BERT-base to evaluate the extent of the domain transfer and also improve on the already strong results of BioBERT in the clinical domain.

Keywords: Language Modelling, Neural language representation models, Statistical and Machine Learning Methods, Semi-supervised, weakly-supervised and unsupervised learning

1. Introduction

Transformer models like BERT (Devlin et al., 2019) and its derivatives outperform other models in many NLP benchmarks and have achieved widespread acceptance. Due to the general nature of pre-training data, these models often lack specific domain knowledge or vocabulary and under-perform in even broad domains like the medical one (Lee et al., 2020). One option to impart this domain knowledge is to use structured data in the form of knowledge graphs. Additionally, recent findings in model compression have shown that these large transformer models contain redundancies in their components (Michel et al., 2019; Sanh et al., 2019). We propose KIMERA, a novel re-training method for effective knowledge injection in transformer models which enhances these redundant parameters with the help of structured domain knowledge.

First, we detect the redundant attention heads in these transformer models, by using the findings of model pruning. This allows KIMERA to leave the relevant components of the model untouched while improving the more irrelevant ones. We retrain and specialize these redundant components in a Multi-Task training scheme enabling the model to abstract information from the structured knowledge sources. We use common tasks from the Knowledge Graph Completion field to facilitate this training.

Focusing on the clinical domain, we choose Clinical Answer Passage Retrieval(CAPR) and Clinical Outcome Prediction(COP) as downstream tasks. Medical knowledge graphs like UMLS (Bodenreider, 2004) contain commonly known medical knowledge like disease-symptom or drug interactions, while clinical notes often represent the current health state of a particular patient. Therefore, both can effectively complement each other for a deep patient representation. Additionally, we probe our models with GLUE (Wang et al., 2019) to assess the effect on the general language abilities that KIMERA retains after the domain transfer. We evaluate the effects of KIMERA on BERT and BioBERT (Lee et al., 2020). BioBERT serves as a strong baseline that is trained with medical data, and our method manages to further improve on its results.The contributions of this paper are as follows:

- Applying model compression-based analysis for targeted retraining of attention heads
- A novel Multi-Task retraining scheme based on Knowledge Graph Completion to integrate structured knowledge
- Experiments on 5 different strategies to employ our method
- An evaluation on domain adaptation to the medical domain in 8 downstream tasks over both BERT-base and BioBERT
- We publish PyTorch code[1] and plan to upload trained models to `huggingface.co`

The remainder of this paper is structured as follows: Section 2 illustrates KIMERA's process; 3 introduces the downstream tasks and Knowledge Graphs that we use in our experiments, Section 4 discusses the experiments and results on these tasks, Section 5 contains an analysis on the actual impact the retraining has on the model, Section 6 showcases related work and finally Section 7 discusses future work and conclusions.

[1] https://anonymous.4open.science/r/kg-transformers/README.md

2. Methodology

An overview of our method is depicted in Figure 1 **A)**. We start with a *pre-trained transformer model*, a domain-specific *knowledge graph*, and a *downstream task* within that domain that we desire to improve on. KIMERA is composed of three major steps:

1. Compute the **attention head importance** of a fine-tuned model on the downstream task we intend to improve on.
2. **Retrain** the less essential heads (using the attention mask generated in step 1) of a pre-trained model using a multi-task knowledge graph generation scheme.
3. **Fine-tune** and evaluate the retrained model on the downstream task.

2.1. Compute Attention Head Importance

This first step enables the detection of the parameter redundancy that we aim to re-purpose. We start with the model fine-tuned on a downstream task that we intend to improve on. We use recent findings in transformer pruning to identify a subset of the model parameters (attention heads) that can be targeted in the subsequent retraining step. Specifically, we follow (Michel et al., 2019) in their computation of the *head importance* and *head pruning mask*, where they modify multi-head-attention $MHAtt$(Vaswani et al., 2017) into

$$MHAtt(\mathbf{x}, q) = \sum_{h=1}^{N_h} \xi_h Att(\mathbf{x}, q) \tag{1}$$

where Att is the vanilla attention, **x** is a sequence of d-dimensional vectors, and q is a d-dimensional query vector. This proposed modification of the Multi-Head-Attention adds ξ_h as a binary control variable that turns on or off a specific attention head h. Based on this modification (Michel et al., 2019) introduce a score of the relevance of each attention head.

$$I_h = \mathbb{E}_{x \sim X} \left| \frac{\partial \mathcal{L}(x)}{\partial \xi_h} \right| \tag{2}$$

This importance score of each head I_h approximates the expected absolute sensitivity of the loss in the downstream task $\mathcal{L}(x)$ to ξ_h, i.e. the sensitivity to having a specific head enabled for a subset of the training or validation data X. In practice I_h is approximated by accumulating the absolute of the gradients w.r.t the parameter ξ_h for each of the samples in X, then it is normalized resulting in a value ranging from 0 to 1. Based on the importances I_h, the computation of the pruning mask follows an iterative ablation of a proportion ρ of the attention heads, setting their corresponding ξ_h to 0. This process halts once a threshold τ of the overall performance on the downstream task is reached. The result of this process is a pruned fine-tuned model and a mask of L layers × M attention heads with values in $\{0, 1\}$ which we denote M_{hard}, where 0 implies a redundant head and 1 an attention-head that is relevant for the downstream task.

Since our intention is not to compress the model, we diverge from (Michel et al., 2019) by discarding the pruned model, only keeping M_{hard} for our retraining in step 2. Our main contribution here lies in interpreting these redundancies not as parameters to cut away, but instead as something to be repurposed. Specifically, we use these masks to selectively weigh the retraining of the network:

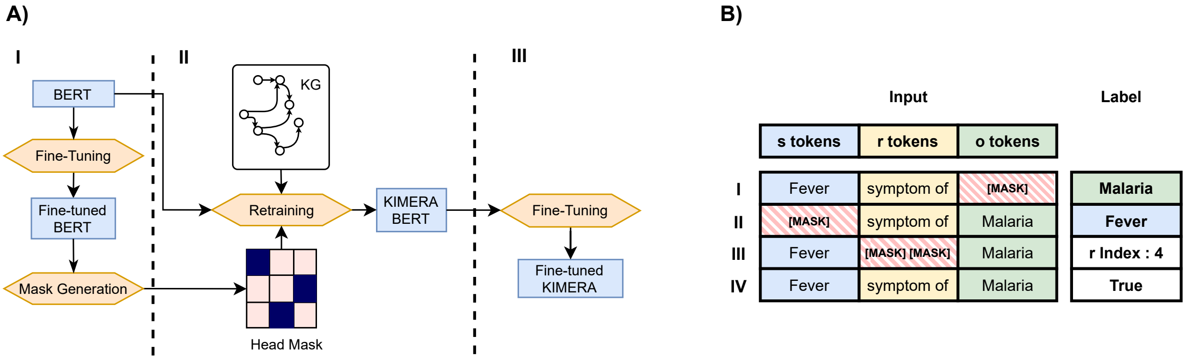
$$W_{i+1}^{lh} = W_i^{lh} - \eta(1 - m^{lh})\nabla\mathcal{L} \tag{3}$$

where W_i^{lh} is one of the (Q, K, V) attention matrices or the weight matrix of the dense output layer (O) for the attention head h in layer l at training iteration i, $\nabla\mathcal{L}$ is the loss gradient applied during the backward pass, η is the general learning rate and m^{lh} is the mask value of head h at layer l. We explore the following three settings for this learning rate adaptation.

Discrete learning rate adaptation. This involves selectively freezing attention heads using directly the information of the pruning (hard) mask. In this case the values m^{lh} are strictly in $\{0, 1\}$. Following our retraining step in equation 3, the mask values yield a non zero learning rate only for the unimportant heads that could be pruned. With this we focus only on retraining and improving the unimportant heads, leaving the important ones untouched.

Soft attention-head mask. To address the fact that partially freezing specific heads during the retraining could yield two sub-networks within the model that result in a disjointed representation, we slightly modify the computation of the head-mask. Here we also iteratively score the heads with I_h. However, we omit the pruning of the unimportant heads in each iteration, and instead of setting their ξ_h to 0 we set it to the last normalized I_h that would have made them pruning candidates, retaining their importance in the resulting soft mask. This guarantees that the values of the attention of the unimportant heads are not entirely removed in the forward pass, but rather weighted according to their importance. We again stop the process once the performance of the network on the downstream task has reached a proportion τ of the metric. The resulting mask M_{soft} can be used as a soft weighting of the learning rate in our retraining step (3).

Weighing the forward pass. In addition to selectively weighing the backward pass, we explore applying the attention-head masks in the forward pass during retraining. Model predictions are then only calculated using non-masked heads. This is to control the level of isolation of the targeted heads as a sub-network. We treat this behaviour and the masks as another hyper-parameter of the retraining stage.



	Input			Label
	s tokens	r tokens	o tokens	
I	Fever	symptom of	[MASK]	Malaria
II	[MASK]	symptom of	Malaria	Fever
III	Fever	[MASK] [MASK]	Malaria	r Index : 4
IV	Fever	symptom of	Malaria	True

Figure 1: **A)** KIMERA consists of three phases: **I** A transformer model is fine-tuned and a head-mask is computed by identifying redundancies. **II** The computed mask is then used in conjunction with a multi-task training based on knowledge graph completion. Finally, the model is fine-tuned on the target task. **III** The retrained model is fine-tuned on the domain-specific task to culminate the domain transfer. **B)** Examples of KG retraining tasks. **I** and **II** *Entity Prediction* with a Masked Language Modelling objective. **III** *Relation Prediction* with a multi-class classification objective, and **IV** *Triplet Classification* with a binary classification objective.

2.2. Retraining

This step uses a pre-trained model, an attention mask computed in the previous step, and a knowledge graph, resulting in a model that can be fine-tuned on the final downstream task. We follow a multi-task training scheme with tasks based on knowledge graph triplets. We adopt the common Knowledge Graph Completion tasks of *entity prediction*, *relation prediction*, and *triplet classification*, e.g. (Bordes et al., 2011; Socher et al., 2013; Yao et al., 2019), and apply them in this novel way. These tasks are intended to specialize the redundant or unimportant attention heads into the domain of the knowledge base.

Multitask Training Scheme. We follow a multi-task scheme to force the target models to generalize by having a combination of multiple competing losses. We explore two different settings. First, we attempt to improve existing pre-trained transformer models, namely BERT or BioBERT, by retraining them. In the second setting, we train BERT from scratch exclusively on the knowledge graph completion tasks to measure the extent of the complementary information added by a knowledge graph. In each task, we target a single knowledge graph triplet denoted in a directed graph by (s, r, o): subject node, relation edge, and object node, respectively. We adopt three link prediction tasks focusing each on completing one of these s, r, or o triplet elements, and a fourth task validating the plausibility of the whole triplet. Figure 1 **B)** depicts examples for these tasks. Each input row depicted in this figure is embedded as a single input sequence, with separator tokens between the columns.

Entity Prediction. We frame entity prediction as a Masked Language Modelling task (Devlin et al., 2019). In our multi-task setting, this results in two tasks: given (s, r) or (r, o), o or s have to be generated correspondingly. In contrast to (Devlin et al., 2019), we mask and predict all tokens of o or s. In both cases, this generation results in a sequence of tokens denoting the model's predictions for the masked component. The loss being optimized is token-wise cross-entropy over the model vocabulary.

Relation Prediction. In this task, given (s, o), the objective is to predict r. While this task could also be modeled with a (masked) language modeling objective similar to the Entity Prediction tasks, we opt to implement this task as a multi-class classification since, in our case, the number of relations in the graph is very small compared to BERT's vocabulary. This simplifies the task substantially.

Triplet Classification. This task tests if a graph triplet is a valid triplet present in the knowledge graph. Given a triplet (s, r, o), this task involves a binary classification to determine its plausibility. We take valid samples directly from the knowledge graph and generate an equal amount of invalid samples by replacing one of the three components with the same component from a different randomly selected triplet.

Multitask model architecture. To implement this multi-task setting we use the encoder part of the transformer model, pool the output, and add linear layers, one for each task. These output layers have the same size as the hidden size of the transformer model used. We experiment with different pooling techniques as hyper-parameters, e.g. [CLS] token for BERT, average pooling, max pooling, and a learned pooling method using an additional linear layer.

Optimization Objective. During training, we sample batches randomly from all tasks and compute the main loss as a weighted sum of losses corresponding to each one of the tasks

$$\mathcal{L} = \alpha_1\mathcal{L}_1 + \alpha_2\mathcal{L}_2 + ... + \alpha_n\mathcal{L}_n \qquad (4)$$

where $\alpha_1, ..., \alpha_n$ are scalar loss weights which are regarded as hyperparameters, and $\mathcal{L}_1, ..., \mathcal{L}_n$ are the per-task loss functions, namely Categorical Cross Entropy in all tasks. This weighted sum over the tasks is to weigh difficult tasks more strongly to prevent overfitting on some of the simpler tasks.

2.3. Fine-tuning

This is the final step proposed in KIMERA and it involves extracting the encoder from the retrained model and fine-tuning it on the final downstream task as is common practice, yielding a model with specific domain knowledge.

3. Datasets and Downstream Tasks

Ideally, the knowledge graph that we instill into a language model has large amounts of complementary information and is relevant for solving the downstream task. The performance of our retraining method relies on the combination of *knowledge graph, language model, downstream task* fitting appropriately. We leave metrics and an algorithm for automatically evaluating the fitness of such a combination to future work. To evaluate our method, we choose eight datasets from the clinical domain with challenging tasks such as zero shot-retrieval and extreme multi-class classification on hundreds of classes. The clinical domain in particular exhibits issues like limited training data, due to privacy and regulatory issues, and idiosyncratic language, which may highlight insufficiencies in BERT's capabilities (Kalyan and Sangeetha, 2020). Additionally, there is reasonable structured data available for this domain in the form of UMLS(Bodenreider, 2004). It is for these reasons that we decide on the clinical domain to evaluate KIMERA. We specifically highlight the clinical domain, which is closely concerned with direct patient care, as a subset of the general biomedical domain. We choose our tasks in favor of common tasks such as Named Entity Recognition and Relation Extraction since in a clinical setting doctors do not find this type of information extraction sufficient. Instead, they deem complex downstream tasks such as patient cohort retrieval and outcome prediction more useful (Miotto et al., 2016; Topol, 2019).

3.1. Knowledge Graphs

We combine three knowledge graphs into one dataset: UMLS(Bodenreider, 2004), HSDN(Zhou et al., 2014), and the graph from (Rotmensch et al., 2017). We gather ∼2.5M knowledge graph triplets with 43 unique relation types. We limit the sequence length of nodes to 100 tokens and edges to 10 tokens, and pad accordingly. This is done to optimize computation speed while truncating $< 0.1\%$ of triplets.

UMLS(Bodenreider, 2004) The Unified Medical Language System is an aggregation of different medical knowledge sources. This work specifically focuses on UMLS' Metathesaurus, which contains diseases, symptoms, medications, etc., and the relations between them. From the 80 million relationship triplets in UMLS, we filter for relevant relation types, triplets that are complete, and choose to keep only well-populated sub-relations with more than 10k sample triplets. This results in our training corpus of ∼600k triples.

HSDN(Zhou et al., 2014) is constructed from ∼7M PubMed(Sayers et al., 2018) bibliographic records. MeSH(Medical Subject Headings)(Lowe and Barnett, 1994) metadata is used to identify symptom and disease terms. The co-occurrence of at least one symptom and one disease term is then utilized to filter the PubMed records further. From these records, symptom-disease relations are then extracted, resulting in ∼150k triplets.

(Rotmensch et al., 2017) create a knowledge graph from electronic health records collected between 2008 and 2013 from a trauma center and tertiary academic teaching hospital. Concepts are extracted by applying UMLS as well as other sources to these records. The graph is then constructed by a set of 3 probabilistic models which relate symptoms and diseases. The resulting graph contains ∼3k symptom-disease triplets.

3.2. Clinical Answer Passage Retrieval(CAPR)

Retrieving documents and passages from clinical documents is an important task in the medical domain. We evaluate our models on the clinical answer passage retrieval task(CAPR) (Grundmann et al., 2021) in a *zero-shot setting* and across four different datasets. The zero-shot setting puts an even higher burden on each individual model since each model is evaluated as-is, and not fine-tuned to the evaluated datasets. We follow (Grundmann et al., 2021) and evaluate our models using the Cross Encoder Architecture (Humeau et al., 2020), which calculates matching scores over the joint sequence of all query and passage pairs. We use the same training and evaluation described in (Grundmann et al., 2021) and train on Wikipedia articles, and evaluate on WikiSectionQA(Arnold et al., 2020), Mimic-III clinical notes(Johnson et al., 2016), MedQuad(Abacha and Demner-Fushman, 2019), and HealthQA(Zhu et al., 2019) datasets. In this setting, we create only one joint attention-head mask for all four tasks. This mask is generated on a dataset that is combined from held out parts of the test sets of each of the datasets.

3.3. Clinical Outcome Prediction(COP)

We adopt the admission notes dataset by (van Aken et al., 2021) for the Clinical Outcome Prediction tasks. They are based on special filtering of Mimic-III's discharge summaries that simulate patient information at the time of admission. This is achieved by only keeping the following sections: *Chief complaint, (History of) Present illness, Medical history, Admission Medications, Allergies, Physical exam, Family history, Social history.* In particular, this filtering hides all information about the course and outcome of treatment of the patient during their stay.

In-hospital Mortality Prediction Task (MP) This task is a binary classification task, in which the model determines whether a patient deceased during the hospital stay or not. The data is heavily imbalanced with 90% of patients surviving their stay.

Length of Stay Prediction Task (LOS) Here the model classifies a patient's stay at the hospital into 4 classes regarding the length of their stay: $<$ 3 *days,* $3 - 7$ *days,* $1 - 2$ *weeks,* 2+ *weeks.*

Diagnosis Prediction Task (DIA) In this extreme multi-label classification task the model is tasked with assigning ICD-9 diagnosis codes to a patient. Instead of 4-digit codes, we reduce the problem to 3-digit codes, which results in 1266 ICD-9 codes with a power-law distribution.

Procedures Prediction Task (PRO) This task follows the diagnosis prediction task, being a multi-label task utilizing 3-digit ICD-9 codes. There are 711 procedure codes that we use from Mimic-III.

4. Experiments and Results

Our Experiments and Baselines are based on either BERT-base or BioBERT. Although Clinical-BERT(Alsentzer et al., 2019) is another option for comparison, we do not consider it for our evaluation since it is already trained on Mimic-III, skewing the results especially in the zero-shot CAPR scenario.

For BioBERT we choose *dmis-lab/biobert-v1.1* from the huggingface transformers repository (Wolf et al., 2020), and for BERT-base experiments we choose the best model out of BERT-base-uncased and BERT-base-cased. For the Clinical Answer Passage Retrieval, we find that hyperparameter optimization does not have a significant impact, and manually choose reasonable values from several trials. In contrast, Clinical Outcome Prediction is very sensitive to hyperparameters. Therefore we carry out a thorough hyperparameter optimization based on HyperOpt (Bergstra et al., 2013) for all evaluated models. All KIMERA models are trained on the full set of knowledge graph triplets and for a maximum of 5 epochs, but most models converged after a single epoch. Although the parameter α could weigh partially the loss on the tasks, in our experiments it was only used discretely to enable or disable distinct tasks. We find in our experiments that it is usually most beneficial to keep all α_n at 1 and leave the exploration of soft weightings to further research. On a single Nvidia V100 GPU, one epoch takes 18 hours. We choose the head masks resulting from the best base model, calculated with performance threshold $\tau \in [0.95, 0.98, 0.99]$ and a per step pruning ratio $\rho = 0.1$. We explore the effect of the selective retraining of attention heads with KIMERA is done in 5.

4.1. Models and Baselines

We focus on the BERT architecture and the domain specific BioBERT, exploring different variations of KIMERAtrained from these base models.

BERT-base (Devlin et al., 2019) We focus on the smaller BERT-base and choose from the English pre-trained models and use the best of BERT-base-uncased and BERT-base-cased for each task.

BERT-base(pruned). This model is created applying the pruning scheme of (Michel et al., 2019) to BERT-base. The authors showed that this model sometimes outperforms BERT-Base solely due to pruning. Therefore, we include this baseline to confirm that the improvements of our methods cannot be achieved solely by pruning.

BioBERT (Lee et al., 2020) follows the same architecture as BERT-base-cased. This model is a state of the art biomedical language model, and is pre-trained on PubMed for 23 days on 8 V100 GPUs. This is up to 50-250 times slower than using KIMERA to create a domain-specific model.

KIMERA no-mask, **hard-mask**, **soft-mask** make use of different types of masks during the retraining step. *no-mask* uses no mask at all, whereas *hard-mask* and *soft-mask* explore the corresponding discrete and soft learning rate adaptation proposed in 2.1.

KIMERA from-scratch. We investigate the KG retraining as the sole pre-training step. We randomly initialize BERT-base apply the multi-task KG training, before fine-tuning on the downstream tasks.

KIMERA b+f. We base KIMERA b+f on KIMERA hard-mask, but apply the mask both in the backward and forward pass as discussed in 2, which leads to a strict isolation between frozen and unfrozen heads.

KIMERA BioBERT follows *KIMERA hard-mask* but uses BioBERT as a base model. Here we probe if KIMERA can also be used for improving already domain-specific models with additional structured data, besides efficient domain transfer.

4.2. Clinical Answer Passage Retrieval

We choose to calculate only one joint attention mask ahead of retraining instead of individual ones for each task, this due to the zero-shot setting of this benchmark. Table 1 reports results in these tasks. The Cross Encoder shows significant performance differences between models. Most notably *KIMERA hard-mask* and *KIMERA soft-mask* outperform BERT-base across all tasks with a margin of up to 20% in R@1 and up to 35% in R@5. Even *KIMERA no-mask* achieves notable performance boosts. This can be ascribed to the functioning domain transfer with the help of information from UMLS. We also evaluate our methodology on BioBERT and manage to overcome it in all the retrieval tasks, suggesting that KIMERA serves as well to further specialize BioBERT in the medical domain. In the case of Mimic-III, BioBERT is only marginally ahead of BERT-base. KIMERA only beats both of them by a few percentage points, in contrast to the other tasks. One reason for this could be that domain-specific data here is less relevant than for the other tasks.

Model	MedQuad		HealthQA		Mimic-III		Wiki		MP	LOS	DIA	PRO
	R@1	R@5	R@1	R@5	R@1	R@5	R@1	R@5	AUROC	AUROC	AUROC	AUROC
BERT-base	52.63	60.80	40.30	81.82	59.74	72.07	35.44	77.66	81.13	**70.40**	82.08	85.84
BERT-base(pruned)	50.71	60.45	39.92	78.12	61.96	72.64	35.23	75.12	81.07	70.14	80.21	83.48
KIMERA scratch	32.88	74.17	31.23	**83.45**	23.63	41.77	20.63	59.85	75.75	65.74	51.1	64.91
KIMERA no-mask	64.68	92.33	49.01	80.31	65.68	79.78	50.38	80.44	81.63	69.55	82.47	85.91
KIMERA hard-mask	**71.94**	**94.52**	**50.53**	82.71	67.13	80.52	**51.73**	80.72	**81.88**	69.02	**82.59**	**85.95**
KIMERA soft-mask	70.33	93.81	49.50	81.69	67.94	**81.82**	51.25	**81.31**	81.20	68.11	82.35	85.49
KIMERA b+f	70.41	93.91	49.22	80.99	**68.07**	80.43	50.81	81.24	65.72	55.36	81.45	84.21
BioBERT	78.86	97.06	62.07	91.59	64.89	78.81	61.31	90.69	82.55	**71.59**	82.81	86.36
KIMERA BioBERT	**79.74**	**97.93**	**64.14**	**92.26**	**65.22**	**79.02**	**62.48**	**94.32**	**82.87**	71.42	**83.56**	**88.44**

Table 1: Results across the four CAPR datasets using the Cross Encoder architecture(left) and four COP tasks(right). Top part shows scores for models based on BERT-base, bottom part scores for models on BioBERT. KIMERA improves on both BERT-base and BioBERT performance, with the exception of the LOS task.

In general, using an attention-head mask during the re-training does lead to a performance increase over our no-mask approach. However, none of the masking strategies is clearly better than the others. KIMERA from-scratch generally under-performs in all of the retrieval tasks. This reinforces the fact that the information contained in UMLS is only complementary and not a replacement to the general language capabilities of a pre-trained model. Simply pruning the model did also not improve performance for these tasks with the exception of Mimic-III. This demonstrates that the performance increases we observe for KIMERA do not stem from the pruning alone.

4.3. Clinical Outcome Prediction

For this benchmark an attention mask is generated for each of the tasks individually. In contrast to the Passage Retrieval tasks, the COP results show significantly lower variance in the performance between models. (van Aken et al., 2021) highlight numerical errors as one of the major error classes in these tasks, emphasizing that their evaluated models do not follow medical reasoning, but focus on statistical observations. This fact in combination with the already strong performance of the base architecture of BERT-base could account for the small variance.

As shown by Table 1, KIMERA BioBERT achieves the best results with the exception of the LOS task. Similarly, when applying KIMERA to BERT-base we achieve consistent improvements. The different masking strategies of KIMERA performed closely without any particular one standing out as the best. The results of KIMERA *from-scratch* confirm the complementary nature of the UMLS data we found also in the Passage Retrieval tasks. The pruned BERT-base model did not provide performance benefits in these tasks either.

For both the Mortality Prediction and Length of Stay task the back+forward approach performed significantly worse. Given the almost equal performance to other KIMERA models in other tasks, we deem these as outliers that are caused by an insufficient amount of hyper parameter optimization.

The LOS stands out as the only downstream task, including the results in CAPR, where KIMERA did not achieve improvements.

4.4. General Language Understanding (GLUE)

We evaluate KIMERA on GLUE (Wang et al., 2019) and compare it to BERT-base and BioBERT. The results are detailed in Table 2. KIMERAmodels for this evaluation have been trained on the medical KGs with masks generated in CAPR, in order to assess how the medical transfer learning impacts the language capabilities. As expected, BERT-base outperforms the biomedically trained BioBERT across all tasks with its general language pre-training. Furthermore, the comparison between KIMERA no-mask and KIMERA hard-mask shows that the hard-mask version, where only a subset of the attention heads have been retrained, is consistently superior to the non-mask version. This supports our intuition that the masking process enables the model to retain more of its language ability during the transfer learning process. Notably, KIMERA outperforms even BERT-base in 3 of the GLUE tasks. While we expected KIMERA with clinical training to perform slightly worse than BERT-base since the knowledge graph task data does not contain proper grammar in its triplets and therefore skews language perception, the results show that for CoLA, QQP and WNLI tasks this training is particularly beneficial and leads to significant improvements over BERT-base.

5. Discussion and Analysis

We inspect qualitatively the effects of our selective retraining of the attention heads for the Clinical Answer Passage Retrieval setting. We do this for our KIMERA hard-mask experiment.

Model Downstream Redundancy Figure 2 **A** presents the mask for freezing the important (yellow) heads and retraining the unimportant (purple) heads. The most noticeable aspect of this mask is the high number of heads that are rendered as unimportant, namely 102 heads or 70.8% of the model. This high level of re-

Model	CoLA	SST-2	MRPC	STS-B	QQP	MNLI	QNLI	RTE	WNLI	Mean
BERT-base	59.05	**93.34**	**89.37**	**88.79**	89.84	**85.12**	**91.78**	**69.31**	49.30	79.54
BioBERT	43.70	91.28	88.51	88.15	89.59	83.97	90.84	67.50	32.39	75.10
KIMERA no-mask	60.17	92.20	87.71	88.12	89.53	84.49	90.35	67.50	60.17	80.02
KIMERA hard-mask	**62.06**	93.00	88.93	88.53	**90.63**	84.65	91.15	69.12	**62.05**	**81.13**

Table 2: Results of the GLUE benchmark, choosing the best of 10 seeds. KIMERA consistently outperforms BioBERT, and shows improvements over BERT-base in 3 tasks, having the highest mean score of tested models.

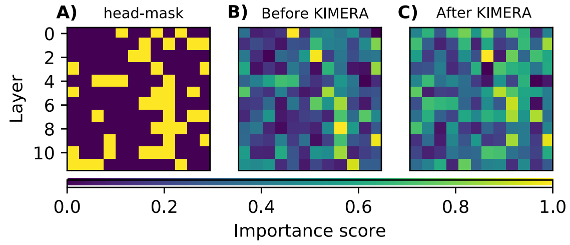


Figure 2: Attention head importance with and without KIMERA for the CAPR task. **A)** Head mask used for retraining. **B)** and **C)** present the head importances I_h before and after using KIMERA, respectively. Our method results in relatively higher and more homogeneous importance of the heads.

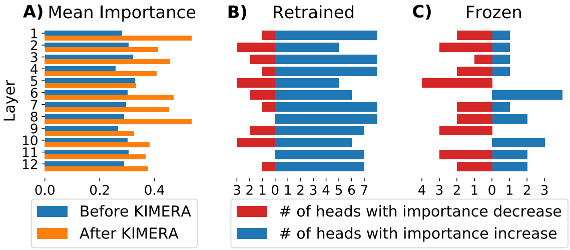


Figure 3: Importance changes per layer for the CAPR task. **A)** Average importance I_h per layer before and after KIMERA. **B)** Number of *retrained* heads that saw an increase/decrease in their importance after KIMERA. **C)** Number of *frozen* heads that saw an increase/decrease in importance with our method. The retrained heads present an overall increase in importance, whereas the frozen heads show mixed results.

Heads	I_h Before KIMERA	I_h After KIMERA
Frozen	0.60	0.53
Retrained	0.17	0.37

Table 3: Mean importance scores I_h before and after KIMERA for frozen and retrained heads in the CAPR task. I_h more than doubles for the retrained heads while it moderately decreases for the frozen heads.

dundancy is compatible with the performance gains we see for this task after applying KIMERA.

Head relevance improvement. In parallel, Figure 2 **B** and **C** show the attention-head importance scores, the former corresponds to the fine-tuned BERT-base and the latter to the KIMERA hard-mask using **A** for the retraining step. It can be seen that the general head-attention importance shown after KIMERA tends to be higher on average (yellow) and more homogeneous. We expand on this by analyzing the mean improvement of the importance scores I_h per layer which is shown in Figure 3 **A**). All the layers present an overall increase in importance w.r.t. the downstream task. Furthermore, we count the number of heads that increase or decrease in importance, but now accounting for the retrained and frozen heads separately; this is shown in Figure 3 **B)** and **C)** accordingly. For the retrained heads, the positive increase of importance is dominant across all layers. This is true not only for their count, but also for the retrained-average importance score I_h shown in Table 3, which more than doubles from 0.17 to 0.37. In contrast, the split of frozen heads that increase or decrease in importance is more mixed, and we notice only a moderate decrease of their average importance scores from 0.60 to 0.53 after applying KIMERA. This behavior supports the intuition that not only the model is better at the downstream task, but also that the retrained heads are more relevant for this improvement.

Limitations. Our proposed methodology is only suitable in the case of domain transfer when the underlying multi-head transformer model underperforms significantly on a benchmark. This is evident in the stark contrast between the gains achieved by KIMERA in the CAPR and COP tasks. The main factors behind this are the level of redundancy of the model for the task, which we gauge by the head-masks, and how complementary the target Knowledge Graph is. The latter is an open question that we leave for further work.

6. Related Work

Our work stands separate from Graph Neural Networks where the focus lies on creating graph embeddings, these are orthogonal to our approach. We base our findings on recent advancements in three different areas of research: model compression, domain transfer, and Knowledge Graph Completion/Generation.

Model Compression is an area of research focused on retaining the original performance of a given model while reducing the number of its parameters. Notable examples are (See et al., 2016), who among others popularized pruning techniques in NLP and specifically NMT, and (Sanh et al., 2019), who use a student-teacher approach (Knowledge Distillation) to yield a smaller but powerful BERT model. Most closely related to our work are (Michel et al., 2019) with an analysis of the efficacy of attention heads. The authors successfully prune a substantial number of attention heads, while retaining, or in some cases even improving, on the original network's performance. We follow their method to determine the importance of attention heads concerning our downstream tasks, but instead utilize it to boost performance and inject new knowledge.

Domain Adaptation. While Transfer Learning (Pan and Yang, 2009) is common for transformer networks due to widely available pre-trained models, domain transfer is a more narrow sub-field. (Xu et al., 2019) demonstrate the efficacy of a post-training or retraining step while (Du et al., 2020) create two retraining tasks: domain distinguishing, and target domain masked language modeling. Instead of relying on self-supervised tasks on raw text, our retraining is based on structured data and knowledge graph completion. We target specifically the medical domain. (Bapna and Firat, 2019) explore domain adaptation in the field of Neural Machine Translation. Their solution adds feedforward-based adapter layers into the network, that contain domain-specific knowledge. Our work instead focuses on implicitly merging domain-specific and general knowledge in the network, rather than adding separate modules.

Medical Language Models. BioBERT (Lee et al., 2020) demonstrate how domain-specific models can be created via pre-training directly on domain-specific data. (Chakraborty et al., 2020) and others follow the same approach utilizing different pre-training corpora. In contrast, we explore leveraging already trained general-purpose pre-trained transformers and re-purposing them for niche domains. Thus, we substantially ease the requirements of data and computational resources in comparison to aforementioned models. (Zhang et al., 2020) and (Hao et al., 2020) train models using UMLS, but do so with significantly different training objectives, and evaluate on the biomedical domain instead of the clinical one.

Structured Knowledge Integration enhances results in NLP tasks by querying external Knowledge Graphs or adding complementary architectural modifications to language models. (Zhao et al., 2020), (Bosselut and Choi, 2019), (Liu et al., 2020), (Zhong et al., 2019) and others make use of explicit sub graphs, which are sometimes dynamically generated. (Zhang et al., 2019) align entities and integrate their matching embedding of a knowledge graph introducing an additional objective to mask language modelling at pre-training. (Peters et al., 2019),(He et al., 2020) and (Wang et al., 2020) train additional transformer-based sub-networks specialized on KG information, and which are used in addition to or are integrated into other networks. In contrast to these works, KIMERA works entirely on the existing architecture of a pre-trained transformer language model. It does not integrate additional modules nor parameters and does not require access to the knowledge graph once the retraining has been completed, containing its knowledge only implicitly.

Knowledge injection involves specializing the knowledge of language models during the training process. (Faruqui et al., 2015) refine word representations with an objective function, which optimizes words that are close in a knowledge graph to be close in the embedding vector space. (Ye et al., 2019) incorporate commonsense knowledge into transformers via pre-training by constructing a multiple-choice Question Answering dataset from a knowledge graph. (Zhang et al., 2020) focus on UMLS, however use Concept Alignment as a training objective, integrating PubMed and other medical literature. Furthermore, (Wang et al., 2021) and (Hao et al., 2020) inject factual knowledge from UMLS and Wikidata, by adding additional objectives to common transformer pre-training. Closest to our work, (Kim et al., 2020) use a multi-task setting to solve two knowledge graph completion tasks and a graph-triple ranking objective in a re-training scheme. As opposed to these works, KIMERA uses a specific multi-task intermediate retraining scheme, which is based on Knowledge Graph Completion/Generation, driven by a selective freezing of the attention heads.

Knowledge Graph Generation focuses on extending knowledge graphs by generating new triplets. (Petroni et al., 2019), (Yao et al., 2019) and (Bosselut et al., 2019) demonstrate the Knowledge Graph Generation capabilities of Transformers in particular. We build on these works, by using this generation as an intermediate step to ground the knowledge into the language model and improve downstream task objectives. (Joulin et al., 2017) propose a fastText-based architecture for node generation, while also combining it with a question answering objective. We extend these tasks with a triplet classification objective and apply them in a different setting to a pre-trained transformer.

7. Conclusion

We propose a novel training methodology for improving pre-trained Language Models and adapting them to the clinical domain. Further, we demonstrate the efficacy of utilizing structured knowledge from clinical knowledge graphs in a domain adaptation training scenario via knowledge graph generation. We explore different strategies for freezing attention heads during retraining and achieve a significant and consistent improvement over strong baseline models. Our careful experiments confirm our hypothesis that KIMERA adequately compensates for limited training data and domain knowledge. It makes large transformer models adaptable with limited effort and our results show that KIMERA manages to improve on the already strong biomedical baseline of BioBERT.

8. Bibliographical References

Alsentzer, E., Murphy, J., Boag, W., Weng, W.-H., Jin, D., Naumann, T., and McDermott, M. (2019). Publicly available clinical BERT embeddings. In *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, pages 72–78, Minneapolis, Minnesota, USA, June. Association for Computational Linguistics.

Bapna, A. and Firat, O. (2019). Simple, scalable adaptation for neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1538–1548, Hong Kong, China, November. Association for Computational Linguistics.

Bergstra, J., Yamins, D., and Cox, D. D. (2013). Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. In *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*, ICML'13, page I–115–I–123. JMLR.org.

Bordes, A., Weston, J., Collobert, R., and Bengio, Y. (2011). Learning structured embeddings of knowledge bases. In Wolfram Burgard et al., editors, *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2011, San Francisco, California, USA, August 7-11, 2011*. AAAI Press.

Bosselut, A. and Choi, Y. (2019). Dynamic knowledge graph construction for zero-shot commonsense question answering. *CoRR*, abs/1911.03876v2.

Bosselut, A., Rashkin, H., Sap, M., Malaviya, C., Çelikyilmaz, A., and Choi, Y. (2019). COMET: commonsense transformers for automatic knowledge graph construction. In Anna Korhonen, et al., editors, *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 4762–4779. Association for Computational Linguistics.

Chakraborty, S., Bisong, E., Bhatt, S., Wagner, T., Elliott, R., and Mosconi, F. (2020). BioMedBERT: A pre-trained biomedical language model for QA and IR. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 669–679, Barcelona, Spain (Online), December. International Committee on Computational Linguistics.

Devlin, J., Chang, M., Lee, K., and Toutanova, K. (2019). BERT: pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, et al., editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Du, C., Sun, H., Wang, J., Qi, Q., and Liao, J. (2020). Adversarial and domain-aware BERT for cross-domain sentiment analysis. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4019–4028, Online, July. Association for Computational Linguistics.

Faruqui, M., Dodge, J., Jauhar, S. K., Dyer, C., Hovy, E., and Smith, N. A. (2015). Retrofitting Word Vectors to Semantic Lexicons. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1606–1615, Denver, Colorado. Association for Computational Linguistics.

Grundmann, P., Arnold, S., and Löser, A. (2021). Self-supervised answer retrieval on clinical notes. *CoRR*, abs/2108.00775.

Hao, B., Zhu, H., and Paschalidis, I. (2020). Enhancing clinical BERT embedding using a biomedical knowledge base. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 657–661, Barcelona, Spain (Online), December. International Committee on Computational Linguistics.

He, B., Zhou, D., Xiao, J., Jiang, X., Liu, Q., Yuan, N. J., and Xu, T. (2020). BERT-MK: Integrating graph contextualized knowledge into pre-trained language models. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2281–2290, Online, November. Association for Computational Linguistics.

Humeau, S., Shuster, K., Lachaux, M., and Weston, J. (2020). Poly-encoders: Architectures and pre-training strategies for fast and accurate multi-sentence scoring. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Joulin, A., Grave, E., Bojanowski, P., Nickel, M., and Mikolov, T. (2017). Fast linear model for knowledge graph embeddings. In *6th Workshop on Automated Knowledge Base Construction, AKBC@NIPS 2017, Long Beach, California, USA, December 8, 2017*. OpenReview.net.

Kalyan, K. S. and Sangeetha, S. (2020). Secnlp: A survey of embeddings in clinical natural language processing. *Journal of Biomedical Informatics*, 101:103323.

Kim, B., Hong, T., Ko, Y., and Seo, J. (2020). Multi-task learning for knowledge graph completion with pre-trained language models. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1737–1743, Barcelona, Spain (Online), December. International Committee on Computational Linguistics.

Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C. H., and Kang, J. (2020). Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinform.*, 36(4):1234–1240.

Liu, W., Zhou, P., Zhao, Z., Wang, Z., Ju, Q., Deng, H., and Wang, P. (2020). K-BERT: enabling language representation with knowledge graph. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 2901–2908. AAAI Press.

Lowe, H. J. and Barnett, G. O. (1994). Understanding and Using the Medical Subject Headings (MeSH) Vocabulary to Perform Literature Searches. *JAMA*, 271(14):1103–1108, 04.

Michel, P., Levy, O., and Neubig, G. (2019). Are sixteen heads really better than one? In Hanna M. Wallach, et al., editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pages 14014–14024.

Miotto, R., Li, L., and Kidd, B. (2016). Deep patient: An unsupervised representation to predict the future of patients from the electronic health records. *Scientific Reports*, 6:26094, 05.

Pan, S. J. and Yang, Q. (2009). A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359.

Peters, M. E., Neumann, M., IV, R. L. L., Schwartz, R., Joshi, V., Singh, S., and Smith, N. A. (2019). Knowledge enhanced contextual word representations. In Kentaro Inui, et al., editors, *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*, pages 43–54. Association for Computational Linguistics.

Petroni, F., Rocktäschel, T., Riedel, S., Lewis, P. S. H., Bakhtin, A., Wu, Y., and Miller, A. H. (2019). Language models as knowledge bases? In Kentaro Inui, et al., editors, *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*, pages 2463–2473. Association for Computational Linguistics.

Sanh, V., Debut, L., Chaumond, J., and Wolf, T. (2019). Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR*, abs/1910.01108v4.

Sayers, E. W., Agarwala, R., Bolton, E. E., Brister, J., Canese, K., Clark, K., Connor, R., Fiorini, N., Funk, K., Hefferon, T., Holmes, J., Kim, S., Kimchi, A., Kitts, P. A., Lathrop, S., Lu, Z., Madden, T. L., Marchler-Bauer, A., Phan, L., Schneider, V. A., Schoch, C. L., Pruitt, K. D., and Ostell, J. (2018). Database resources of the National Center for Biotechnology Information. *Nucleic Acids Research*, 47(D1):D23–D28, 11.

See, A., Luong, M., and Manning, C. D. (2016). Compression of neural machine translation models via pruning. In Yoav Goldberg et al., editors, *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning, CoNLL 2016, Berlin, Germany, August 11-12, 2016*, pages 291–301. ACL.

Socher, R., Chen, D., Manning, C. D., and Ng, A. Y. (2013). Reasoning with neural tensor networks for knowledge base completion. In Christopher J. C. Burges, et al., editors, *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pages 926–934.

Topol, E. (2019). High-performance medicine: the convergence of human and artificial intelligence. *Nature Medicine*, 25, 01.

van Aken, B., Papaioannou, J.-M., Mayrdorfer, M., Budde, K., Gers, F., and Loeser, A. (2021). Self-supervised knowledge integration for clinical outcome prediction from admission notes. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2021, April 19-23, 2021, Volume 1: Long Papers*, apr.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. In Isabelle Guyon, et al., editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Wang, R., Tang, D., Duan, N., Wei, Z., Huang, X., Ji, J., Cao, G., Jiang, D., and Zhou, M. (2020). K-adapter: Infusing knowledge into pre-trained models with adapters. *CoRR*, abs/2002.01808v5.

Wang, X., Gao, T., Zhu, Z., Zhang, Z., Liu, Z., Li, J., and Tang, J. (2021). KEPLER: A Unified Model for Knowledge Embedding and Pre-trained Language Representation. *Transactions of the Association for Computational Linguistics*, 9:176–194, 03.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Scao, T. L., Gugger, S., Drame, M., Lhoest, Q., and Rush, A. M. (2020). Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, October. Association for Computational Linguistics.

Xu, H., Liu, B., Shu, L., and Yu, P. S. (2019). Bert post-training for review reading comprehension and aspect-based sentiment analysis. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*, jun.

Yao, L., Mao, C., and Luo, Y. (2019). KG-BERT: BERT for knowledge graph completion. *CoRR*, abs/1909.03193v2.

Ye, Z., Chen, Q., Wang, W., and Ling, Z. (2019). Align, mask and select: A simple method for incorporating commonsense knowledge into language representation models. *CoRR*, abs/1908.06725v5.

Zhang, Z., Han, X., Liu, Z., Jiang, X., Sun, M., and Liu, Q. (2019). ERNIE: Enhanced language representation with informative entities. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1441–1451, Florence, Italy, July. Association for Computational Linguistics.

Zhang, X., Dou, D., and Wu, J. (2020). Learning conceptual-contextual embeddings for medical text. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 9579–9586. AAAI Press.

Zhao, C., Xiong, C., Qian, X., and Boyd-Graber, J. (2020). Complex factoid question answering with a free-text knowledge graph. In *Proceedings of The Web Conference 2020*, WWW '20, page 1205–1216, New York, NY, USA. Association for Computing Machinery.

Zhong, P., Wang, D., and Miao, C. (2019). Knowledge-enriched transformer for emotion detection in textual conversations. In Kentaro Inui, et al., editors, *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*, pages 165–176. Association for Computational Linguistics.

9. Language Resource References

Asma Ben Abacha and Dina Demner-Fushman. (2019). *A question-entailment approach to question answering*.

Arnold, Sebastian and van Aken, Betty and Grundmann, Paul and Gers, Felix A. and Löser, Alexander. (2020). *Learning Contextualized Document Representations for Healthcare Answer Retrieval*. Association for Computing Machinery, WWW '20.

Olivier Bodenreider. (2004). *The Unified Medical Language System (UMLS): integrating biomedical terminology*.

Johnson, Alistair E.W. and Pollard, Tom J. and Shen, Lu and Lehman, Li-wei H. and Feng, Mengling and Ghassemi, Mohammad and Moody, Benjamin and Szolovits, Peter and Anthony Celi, Leo and Mark, Roger G. (2016). *MIMIC-III, a freely accessible critical care database*.

Rotmensch, Maya and Halpern, Yoni and Tlimat, Abdulhakim and Horng, Steven and Sontag, David. (2017). *Learning a Health Knowledge Graph from Electronic Medical Records*.

Alex Wang and Amanpreet Singh and Julian Michael and Felix Hill and Omer Levy and Samuel R. Bowman. (2019). *GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding*.

Zhou, Xuezhong and Menche, Jörg and Barabasi, Albert-Laszlo and Sharma, Amitabh. (2014). *Human symptoms–disease network*.

Zhu, Ming and Ahuja, Aman and Wei, Wei and Reddy, Chandan K. (2019). *A hierarchical attention retrieval model for healthcare question answering*.