Adapting without forgetting: KnowBert-UMLS

Guilhem Piat*†‡, Nasredine Semmar*†, Alexandre Allauzen‡, Hassane Essafi*, Gaël Bernard* and Julien Tourille*
* CEA List, F-91120, Palaiseau, France
† Université Paris Saclay, 91190 Gif-sur-Yvette, France
‡ LAMSADE, Université Paris Dauphine, F-75775, Paris Cedex 16, France

***Abstract*—Domain adaptation in pretrained language models usually comes at some cost, most notably out-of-domain performance. This type of specialization typically relies on pre-training over a large in-domain corpus, which has the side effect of causing catastrophic forgetting on general text. We seek to specialize a language model by incorporating information from a knowledge base into its contextualized representations, thus reducing its reliance on specialized text. We achieve this by following the KnowBert method, applied to the UMLS biomedical knowledge base. We evaluate our model on in-domain and out-of-domain tasks, comparing against BERT and other specialized models. We find that our performance on biomedical tasks is competitive with the state-of-the-art with virtually no loss of generality. Our results demonstrate the applicability of this knowledge integration technique to the biomedical domain as well as its shortcomings. The reduced risk of catastrophic forgetting displayed by this approach to domain adaptation broadens the scope of applicability of specialized language models.**

***Index Terms*—Knowledge based systems, Deep learning, Transfer learning, Biomedical Computing, Natural language processing**

I. Introduction

With the density and recurrence of specialized vocabulary in some fields such as STEM or law, transformer-based contextualized language models (LMs) such as BERT [1], trained on general text, tend to underperform in those fields. One obvious solution to this out-of-domain performance problem is to reduce the distributional shift between pre-training and deployment by pre-training on in-domain text. This solution is the most commonly used, yet Arumae and Bhatia [2] demonstrate that this approach tends to create models that perform well on the target domain, but poorly on general text, even when such text is part of the pre-training curriculum.

Our approach to language model specialization is to provide it with an external source of relevant knowledge, reducing the need for in-domain text during pre-training as well as the risk of Catastrophic Forgetting (CF). This allows the model to access information pertaining to concepts not seen in the training corpus. This type of approach would usually imply performing an Entity Linking (EL) step – *i.e.* identifying the mentions of concepts in the input text – ahead of leveraging the information in the knowledge base (KB). In practice, satisfying this requirement with a good enough degree of accuracy to render knowledge integration useful is a problem that has yet to be solved. Peters *et al.* [3] on the other hand describe KnowBert, a method to enable a pretrained LM such as BERT to utilize information from a KB which relaxes this constraint, requiring only *candidate* mentions. Following this procedure, we inject knowledge from the Unified Medical Language System (UMLS) Metathesaurus into a BERT-based language model. We name the resulting model KnowBert-UMLS.

We expand on the context for this work and discuss other approaches used for solving these problems in Section II. Because of the scale, variety, and lexical polymorphism of the concepts recorded in UMLS, as well as the relative scarcity of corpora containing labeled examples, there are specific challenges linked to applying KnowBert to UMLS, which we detail in Section III. In Section IV, we discuss the relative performance of our model with respect to relevant baselines on in-domain and out-of-domain tasks. Finally, Section V is dedicated to our conclusions and future work.

II. Related Work

The most common approach for adapting LMs to a given domain is to include text from that domain into the model's pre-training corpus. This is the approach taken, for instance, by models such as BioBERT [4] and BlueBERT [5] in the biomedical domain. The performance of this method and its tendency for CF have been investigated by, among others, Arumae and Bhatia [2]. Some solutions for domain adaptation do not share this flaw, but are typically applied at the fine-tuning stage, which means the process must be carried out for each individual sub-task, and requires sets of labeled data for both the source and target domain.

In contrast, knowledge integration typically leverages pre-existing KBs at the pre-training stage, meaning the domain adaptation step needs only be carried out once to benefit all of the various downstream tasks. In addition, KBs are a much denser source of information than raw text, potentially reducing the amount of in-domain text required. In the interest of reducing the pre-training burden of Transformer-based LMs such as BERT and expanding the range of concepts that they can predict, multiple knowledge integration methods have been developed. One of the main categories of approaches is to rely on the Transformer's attention mechanism to combine entity and word information, as do ERNIE [6] and KnowBert [3]. Another common type of approach is to align entity representations with token representations as does CODER [7]. UmlsBERT [8], in contrast, does not fit into the aforementioned categories as it mainly consists of biasing the BERT input vectors for entity mentions with a topic vector, and changing the Cross-Entropy loss in the Masked Language Modeling (MLM) objective to a

978-1-6654-6975-3/22/$31.00 ©2022 IEEE

Binary Cross Entropy Loss, setting all synonyms of a medical term as valid targets.

An important distinction between methods is whether they require an upstream EL step. This property is one of the most important criteria for selecting a knowledge integration method in the biomedical case, and with UMLS in particular, as it is currently an unsolved task. On the MedMentions corpus [9], for instance, the Entity Linker used by UmlsBERT reaches an F_1 score of 0.178 as reported by Kraljevic *et al.*. The aforementioned knowledge integration methods are subject to this limitation, with the exception of KnowBert, which relaxes the EL requirement, calling only for candidate entity mentions. KnowBert is thus less limited by the EL performance than other knowledge-based models, and does not require as much in-domain text as pre-training-based approaches. It therefore has considerable potential to effectively utilize the KB and is unlikely to suffer from CF on general text.

III. KnowBert-UMLS

The blueprint for KnowBert-UMLS, detailed in Fig. 1 (a), is based on KnowBert, and comprises three main sections: the pretrained LM backbone, the KB with its candidate generator, and a Knowledge Attention and Recontextualization module (KAR).

A. Architecture

1) Pretrained LM backbone: BERT-based models comprise L Transformer Blocks, with each block i taking as input N partially contextualized token representations in $\mathbb{R}^H$, arranged as a matrix $\mathbf{H}_{i-1} \in \mathbb{R}^{N \times H}$, recontextualizing them using attention, and returning a same-size matrix $\mathbf{H}_i$. As a backbone, we use $\text{BERT}_{\text{BASE}}$, which is pretrained on the Wikipedia and Books [11] corpora containing approximately 3.3 billion words, and for which $N = 512$, $L = 12$ and $H = 768$. Despite having, in theory, the option to use any transformer-based language model, in an effort to isolate variables, we do not choose to use a higher-performing or specialized alternative. Moreover, BERT is better suited to the objective of this study, which is to pursue knowledge enrichment without CF, rather than to top state-of-the-art performance in any given task.

2) Candidate Generator: Whilst KnowBert does not require an upstream Entity Linking step, it does require a set of *candidate mentions* $\mathcal{C}$ in order to incorporate information from the KB. Each candidate mention comprises a *candidate span* s and a set $\mathcal{E}_s$ of corresponding *candidate entities* from the KB. Formally:

$$\mathcal{C} = \{(s, \mathcal{E}_s) | \forall s\} \tag{1}$$

Each candidate entity $e \in \mathcal{E}_s$ represents a concept in the KB and is composed of an embedding $\mathbf{e}$, and a prior probability p:

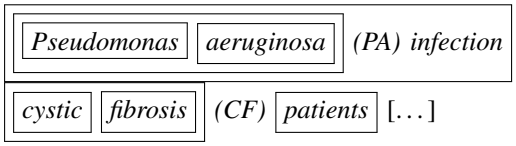
$$\mathcal{E}_s = \{e : (\mathbf{e}_e, p_e) \mid \mathbf{e} \in \mathbb{R}^K, \ \sum_e p_e = 1\} \tag{2}$$

where K is determined by the algorithm used to derive entity embeddings from the KB. In the case of UMLS, we use the pretrained embeddings provided by Maldonado *et al.* [12] which set $K = 50$.

A candidate span can be any sub-string in the sentence which is deemed sufficiently similar to an entity in the KB, and can overlap, or be nested with other spans. For an instance, consider the following phrase:

> *Pseudomonas aeruginosa (PA) infection in cystic fibrosis (CF) patients* [...]

A candidate generator might generate the following candidate spans, outlined in boxes:



Each of these candidate spans would be paired with a set of candidate entities $\mathcal{E}$.

3) KAR: On an abstract level, the KAR remains largely unchanged from KnowBert. It slots in-between two BERT layers i and $i+1$ and functions similarly to a Transformer Block, taking as input partially contextualized word representations $\mathbf{H}_i$ and outputting knowledge-enriched, recontextualized word representations $\mathbf{H'}_i \in \mathbb{R}^{N \times H}$. As an additional input, it takes a set of *candidate mentions* $\mathcal{C}$.

The knowledge incorporation step is performed in the entity embedding space $\mathbb{R}^{N \times K}$; the KAR thus linearly projects the partially contextualized wordpiece embeddings to the entity space and back:

$$\begin{aligned} \mathbf{H}_i^{proj} &= \mathbf{H}_i \mathbf{W} + \mathbf{b} \\ \mathbf{H}'_i &= \mathbf{H}_i'^{proj} \mathbf{W}' + \mathbf{b}' + \mathbf{H}_i \end{aligned} \tag{3}$$

Where $\mathbf{W}$, $\mathbf{W}'$, $\mathbf{b}$ and $\mathbf{b}'$ are learned and $\mathbf{H}_i'^{proj}$ is the matrix of knowledge-enriched token representations embedded in entity space (see Fig. 1 (b)).

The knowledge integration process itself comprises four main steps. First, the token representations for each candidate mention are pooled using an attention-based weighted sum following Lee *et al.* [13] into a matrix $\mathbf{S} \in \mathbb{R}^{|\mathcal{C}| \times K}$. In order to identify false positives among nested or overlapping candidate spans as well as commonly co-occurring entities, the span representations exchange information using Multi-Head self-Attention as in a standard Transformer block, resulting in the contextualized span representations $\mathbf{S^e}$.

For every given span s, we write the corresponding contextualized span embedding from $\mathbf{S^e}$ as $\mathbf{s^e} \in \mathbb{R}^K$, the vector of prior probabilities of corresponding candidate entities $\mathbf{p}_s \in \mathbb{R}^{|\mathcal{E}_s|}$, and the matrix of corresponding candidate entity embeddings $\mathbf{E}_s \in \mathbb{R}^{K \times |\mathcal{E}_s|}$.

$$\begin{aligned} \boldsymbol{\psi}_s &= \text{Softmax}(\text{MLP}(\mathbf{p}_s, \mathbf{s^e} \cdot \mathbf{E}_s)) \\ \tilde{\mathbf{e}}_s &= \boldsymbol{\psi}_s \cdot \mathbf{E}_s^\intercal, \quad \in \mathbb{R}^K \end{aligned} \tag{4}$$

Where $\boldsymbol{\psi}_s$ is an estimate of the posterior probabilities of each candidate entity for s, and $\tilde{\mathbf{e}}_s$ is a weighted average

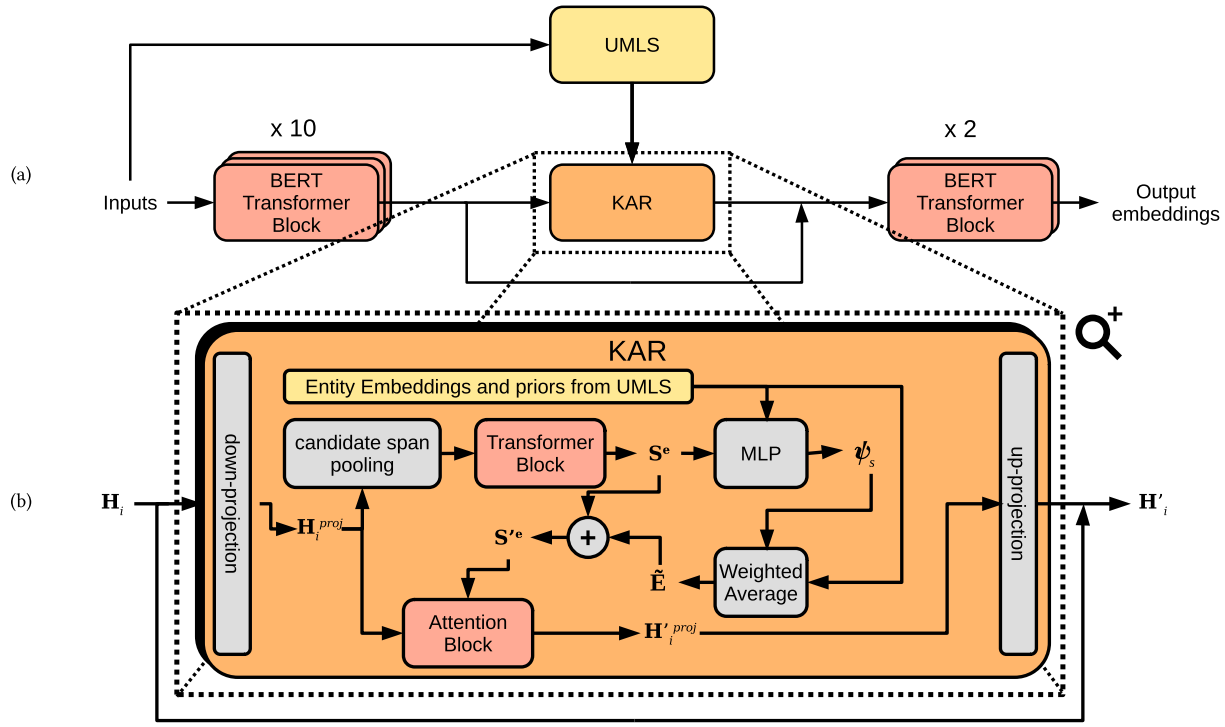


Fig. 1. Overview of the structure of KnowBert-UMLS (a) with detailed breakdown of the KAR (b).

of candidate entity vectors for s. As an additional benefit beyond knowledge integration, this can be used for Entity Linking by simply choosing the entity in the KB which has the embedding most similar to $\tilde{\mathbf{e}}_s$ or the one with the highest estimated posterior probability.

The knowledge-enriched span representations $\mathbf{S'^e} \in \mathbb{R}^{|\mathcal{C}| \times K}$ are then defined as the sum of the contextualized span representations $\mathbf{S^e}$ and the matrix of computed entity vectors $\tilde{\mathbf{E}}$, and the knowledge is transferred from the span representations to the wordpiece embeddings using Multi Head Attention (MHA) followed by a position-wise Multi-Layer Perceptron (MLP), similarly to a Transformer block:

$$\mathbf{H}_i'^{proj} = \mathrm{MLP}(\mathrm{MHA}(\mathbf{H}_i^{proj}, \mathbf{S'^e}, \mathbf{S'^e})) \tag{5}$$

The KAR can be inserted between any two transformer blocks i and $i+1$, and multiple KARs can be inserted simultaneously between different blocks. In the single-KAR case however, using $\mathrm{BERT}_{\mathrm{BASE}}$ as the pretrained backbone, insertion is most effective at block $i = 10$.

B. Training

The pre-training for KnowBert models is a three step process. First the backbone is pre-trained on a language modeling objective, specifically on a combination of the MLM and next sentence prediction objectives in the case of BERT. Then, the KAR is trained on the Entity Linking (EL) objective, minimizing the log-likelihood of the estimated posterior probabilities of candidate entities:

$$\mathcal{L}_{\mathrm{EL}} = -\sum_s log\left(\frac{\exp(\psi_{sg})}{\sum_{k=1}^{n}\exp(\psi_{sk})}\right) \tag{6}$$

The model is subsequently trained on both the language modeling objective used in pre-training and EL. This step is similar to an extended pre-training, but its main objective is to allow the Transformer Blocks which receive knowledge enriched information from the KAR to learn to interpret and integrate it. To avoid ambiguity with regular pre-training, we call this step *re-training*. Optimizing only one of the two objectives during this phase leads to CF of the other, even when the weights of the KAR are frozen. Once the KAR is fully integrated into the model with this step, the model can be fine-tuned to any task much like a typical BERT-based model.

C. Leveraging UMLS

UMLS indexes over four million biomedical concepts, such as *headache*, *spleen*, or *acetylsalicylic acid*, grouped into 135 semantic types including *organisms*, *anatomical structures*, and *diseases or syndromes*. So far, due in part to the difficulty of reliably identifying UMLS concepts in text, leveraging the concepts themselves from UMLS has been out of reach of knowledge integration techniques. UmlsBERT, for instance, only uses embeddings for clusters of semantic types and uses UMLS as a thesaurus for single-word concepts.

Whilst the computational impact of the KAR is negligible, making candidate generation tractable is non-trivial at the scale of our KB and re-training corpus as discussed by Piat *et al.* [14]. We have benchmarked a variety of algorithms, most of

which were far too computationally inefficient for practical use in this context. The combination of neural networks and rules used by ScispaCy [15] was the most efficient candidate generation method, yet was still too computationally expensive to use in realtime during re-training, and the candidates had to be pre-generated. Candidate generation on a 153.6 Billion sentence corpus took approximately three days on a computing cluster, using fourteen Xeon 36-thread CPUs clocked at 3GHz.

We compared the performance of KnowBert using both UMLS Concepts and UMLS Semantic Types as KB entities forming the basis of our knowledge integration. Attempting to integrate knowledge from UMLS concepts did not work, resulting in the model performing on par with an unmodified BERT, as it learned to not take into account the knowledge integrated by the KAR. We suspect two main factors are at play, leading to the KAR being unable to accurately learn to estimate the posterior probabilities for the candidate concepts. First, the precision of the candidate generator is lower for UMLS concepts. For the candidate generator, Recall bounds the model's ability to incorporate knowledge (since no knowledge can be incorporated from an entity not identified by the candidate generator), and Precision affects the imbalance between positive and negative samples during the EL objective. Therefore, whilst maximizing recall maximizes the model's potential, doing so at the cost of precision increases noise in the EL dataset and makes learning more difficult.

The second factor we believe to be responsible for the underperformance of Concept knowledge integration is the lack of concept coverage in the training data, which is under 1% of all concepts. Consequently, setting the weight of the KAR's contributions to 0 is the policy which most accurately predicts masked tokens during training. Despite the Semantic Type information being less insightful than the Concept information, the increased quality of candidates and density of training data in annotated examples (approximately 94% coverage of all types) make Semantic Type information worth incorporating. Henceforth, all mentions of KnowBert-UMLS assume that we use Semantic Types as knowledge base entities.

IV. Experiments

To evaluate our model, we choose two in-domain and two out-of-domain tasks. For our in-domain tasks, we choose Named Entity Recognition (NER) on the n2c2 (formerly known as i2b2) 2010 dataset [16] and Relation Extraction (RE) on the ChemProt [17] dataset. These tasks are fairly standard, and are part of the BLUE [5] benchmark. For our out-of-domain tasks, we choose Natural Language Inference (NLI) and Linguistic Acceptability as Arumae and Bhatia [2] demonstrated that these tasks were particularly affected by extended-pretraining-induced CF with BioBERT. Specifically, we choose the SNLI [18] dataset, and an altered version of the CoLA [19] dataset (see section IV-D) respectively.

All performance scores have been scaled up (from $[0, 1]$) by a factor of 100 for readability. In all tables, the underlined result is BERT which, as a general language model, is expected to have the best performance on the general language tasks (CoLA & SNLI). In bold is the best specialized model.

TABLE I
Pre-Training Corpus Size (Billions of Words) by Type for Baselines Versus KnowBert-UMLS.

Model	Biomedical	General
$\text{BERT}_{\text{BASE}}$	0.0	3.1
BioBERT	18.0	3.1
PubMedBERT	3.2	0.0
BlueBERT	4.5	3.1
UmlsBERT	18.5	3.1
KnowBert-UMLS	**2.2**	**3.1**

We choose our baselines to represent various amounts of in-domain and out-of-domain pre-training corpus sizes, which we break down in table I.

For all models and all tasks, final token or sequence classification is performed using a linear classifier. For all experiments, all models were fine-tuned with the following hyperparameters: our models are trained for 10 epochs with an initial learning rate of 2×10^{-5} and weight decay of 0.01. Our optimization algorithm is AdamW. The model state which performed best on the validation split of each dataset was evaluated on the test set. Results are averaged over multiple experiments.

A. Biomedical NER

As the methodology of previously published results for our baselines is inconsistent, we evaluate the various BERT-based language models on this task ourselves. We use the micro-averaged F_1 score as computed by Seqeval [20] in strict mode and provide a breakdown of precision and recall, as sub-sequence classification (as opposed to sequence classification) does not cause micro-averaged precision and recall to be equal. We use IOB2 as our annotation and prediction scheme.

We expect an improvement over BERT due to specialization, whilst UmlsBERT, as the most heavily pretrained model, should perform best overall. From the results in Table II, we gather that the KAR succeeded in specializing the model, as KnowBert-UMLS outperforms $\text{BERT}_{\text{BASE}}$ by a considerable margin. However, the KAR seems to not have been quite as effective of a specialization method for NER as the others. We hypothesize that its comparatively low precision is due in part to the fairly high False Positive rate of the candidate generator, which may falsely identify mentions of UMLS entities, and through knowledge integration, lead the contextualized word representations to include misleading information.

B. Biomedical Relation Extraction

This is a sequence classification task, wherein two entities per sequence are marked with special characters, and the model must determine which of five relation types (or no relation) exists between them. Scores for BioBERT, PubMedBERT and BlueBERT are self-reported scores of the overall best-performing version of each model. The performance of

TABLE II
PERFORMANCE ON THE N2C2 2010 NER TASK.

Model	P	R	F_1
$BERT_{BASE}$	82.71	86.21	84.42
BioBERT	85.20	87.74	86.46
PubMedBERT	86.62	88.28	87.44
BlueBERT	86.68	88.71	87.68
UmlsBERT	86.92	89.46	**88.18**
KnowBert-UMLS	86.63	85.84	86.23

TABLE III
PERFORMANCE ON THE CHEMPROT RE TASK, MICRO-F_1

Model	micro F_1
$BERT_{BASE}$	66.51
BioBERT	75.14
PubMedBERT	**77.24**
BlueBERT	69.15
KnowBert-UMLS	70.74

TABLE IV
PERFORMANCE ON THE SNLI TASK, MICRO-F_1.

Model	micro F_1
$BERT_{BASE}$	89.24
BioBERT	88.90
PubMedBERT	88.81
BlueBERT	88.20
UmlsBERT	88.59
KnowBert-UMLS	**89.03**

TABLE V
PERFORMANCE ON THE MODIFIED COLA TASK, MICRO-F_1.

Model	Matthews Corr.
$BERT_{BASE}$	60.50
BioBERT	49.30
PubMedBERT	42.90
BlueBERT	39.76
UmlsBERT	44.24
KnowBert-UMLS	**58.52**

$BERT_{BASE}$ was measured by us using the version of the ChemProt corpus distributed by Peng *et al.* [5].

As entities are explicitly marked in the training set, the importance of a good model for grammar is lessened with respect to other tasks. We therefore do not expect general language understanding to be highly predictive of performance on this task. Rather, knowledge of biomedical entities and their relations is expected to be of greater importance. We therefore expect KnowBert-UMLS, which seeks to acquire specifically this type of knowledge, as well as the heavily pretrained BioBERT, to perform well on this task, with $BERT_{BASE}$ performing worst. Our results in Table III largely agree with our predictions. PubMedBERT, however, which was pretrained only on biomedical text, performs better than expected, implying out-of-domain pre-training may in fact be detrimental.

Once more, we observe that KnowBert-UMLS outperforms BERT, implying that specialization was successful, yet it underperforms in comparison to other pretrained models.

C. General NLI

We evaluate all models ourselves on the SNLI task, in which two sequences are fed to the LM. It must determine whether their relationship is one of entailment, contradiction, or neither. As this is a general language task, we don't expect any specialized model to significantly outperform $BERT_{BASE}$, and we expect KnowBert-UMLS to perform on par with BERT. Due to the similarities with the WNLI task from GLUE and given the findings of Arumae and Bhatia [2], we expect the models with extended pre-training to perform poorly on this task.

Table IV shows results in line with our predictions, *i.e.* KnowBert-UMLS is the closest to $BERT_{BASE}$ in terms of F_1 score. However, the gap in performance between the models with extended pre-training and the others is narrower than expected. This is likely due to SNLI not being as adversarial as WNLI.

D. Linguistic Acceptability

Our dataset for the Linguistic Acceptability task is based on the CoLA task from the GLUE benchmark. Since CoLA does not make the labels of its test split public however, and due to various submission restrictions, we have rearranged the available annotated examples into new training, validation, and test splits. In an effort to make our tests reproducible, we use the validation split for final testing, and use the un-shuffled final 500 entries of the train split of version 1.1 as our new validation set.

The objective for this task is to classify sequences as "linguistically acceptable" (*i.e.* grammatically correct and natural-sounding) or not. We evaluate our model and baselines using the Matthews Correlation Coefficient, which is the metric used by GLUE and is generally preferred to F_1 in a binary classification setting as it isn't biased in favor of the positive class. Consistent with our predictions, our results in Table V show that $BERT_{BASE}$, as the general purpose language model, performs the best, and KnowBert-UMLS comes second as the way it is trained is meant to reduce CF.

V. CONCLUSIONS

KnowBert-UMLS outperforms BERT on Biomedical tasks whilst outperforming every other specialized model in out-of-domain tasks. Reducing extended pre-training in favor of Knowledge integration therefore proves to be a successful way of specializing a language model such as BERT to a given domain whilst reducing the impact of CF. However, KnowBert-UMLS does not perform as well as some other models in the biomedical domain, meaning that its specialization method is less effective. We explain this by the fact that the Concepts in the UMLS KB are too numerous and annotated text too rare to learn from effectively, and Semantic Type knowledge is not as informative on an entity-mention level as being familiar with the vocabulary is.

Our results indicate several areas for improvement. In particular, the n2c2 2010 NER task suggests that an improved identification of False Positives from the candidate generator may improve performance by reducing the amount of noise the KAR includes as it incorporates knowledge. Furthermore, the use of semantic types as a source of knowledge may not be as effective to leverage in UMLS as a subset of commonly used concepts, or a more fine-grained clustering.

As Peters *et al.* have shown, the KnowBert architecture is capable of supporting multiple KBs concurrently. KnowBert-UMLS may therefore be further specialized in the biomedical domain with the integration of an additional KB, or it may even support multi-specialization, using KBs from different fields.

Lastly, the improvements brought by this method of knowledge integration may be orthogonal to the improvements brought by extended pre-training or other knowledge integration methods. Using a biomedical LM as a pretrained backbone may lead to a new state of the art in biomedical language modeling.

Acknowledgment

This publication was made possible by the use of the FactoryIA supercomputer, financially supported by the Ile-De-France Regional Council.

References

[1] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the NAACL: Human Language Technologies, Volume 1*, 2019, pp. 4171–4186.

[2] K. Arumae and P. Bhatia, "CALM: Continuous Adaptive Learning for Language Modeling," *arXiv preprint arXiv:2004.03794*, 2020.

[3] M. E. Peters, M. Neumann, R. L. Logan, R. Schwartz *et al.*, "Knowledge enhanced contextual word representations," in *EMNLP*, 2019.

[4] J. Lee, W. Yoon, S. Kim, D. Kim *et al.*, "BioBERT: a pre-trained biomedical language representation model for biomedical text mining," *Bioinformatics*, Sep. 2019.

[5] Y. Peng, S. Yan, and Z. Lu, "Transfer learning in biomedical natural language processing: An evaluation of bert and elmo on ten benchmarking datasets," in *Proceedings of the 18th BioNLP Workshop and Shared Task*, 2019, pp. 58–65.

[6] Z. Zhang, X. Han, Z. Liu, X. Jiang *et al.*, "ERNIE: Enhanced language representation with informative entities," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019, pp. 1441–1451.

[7] Z. Yuan, Z. Zhao, H. Sun, J. Li *et al.*, "CODER: Knowledge-infused cross-lingual medical term embedding for term normalization," *Journal of biomedical informatics*, vol. 126, p. 103983, 2022.

[8] G. Michalopoulos, Y. Wang, H. Kaka, H. Chen, and A. Wong, "UmlsBERT: Clinical domain knowledge augmentation of contextual embeddings using the unified medical language system metathesaurus," in *Proceedings of the 2021 Conference of the NAACL: Human Language Technologies*, 2021, pp. 1744–1753.

[9] S. Mohan and D. Li, "MedMentions: A Large Biomedical Corpus Annotated with UMLS Concepts," *arXiv:1902.09476 [cs]*, Feb. 2019.

[10] Z. Kraljevic, T. Searle, A. Shek, L. Roguski *et al.*, "Multi-domain clinical natural language processing with medcat: the medical concept annotation toolkit," *Artificial Intelligence in Medicine*, 2021.

[11] Y. Zhu, R. Kiros, R. Zemel, R. Salakhutdinov *et al.*, "Aligning books and movies: Towards story-like visual explanations by watching movies and reading books," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 19–27.

[12] R. Maldonado, M. Yetisgen, and S. M. Harabagiu, "Adversarial learning of knowledge embeddings for the unified medical language system," *AMIA Summits on Translational Science Proceedings*, vol. 2019, p. 543, 2019.

[13] K. Lee, L. He, M. Lewis, and L. Zettlemoyer, "End-to-end neural coreference resolution," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2017, pp. 188–197.

[14] G. Piat, N. Semmar, A. Allauzen, H. Essafi, and J. Tourille, "Enriching contextualized representations with biomedical ontologies: Extending KnowBert to UMLS," in *Science and Information Conference*. Springer, 2022, pp. 760–773.

[15] M. Neumann, D. King, I. Beltagy, and W. Ammar, "ScispaCy: Fast and Robust Models for Biomedical Natural Language Processing," in *Proceedings of the 18th BioNLP Workshop and Shared Task*. Florence, Italy: Association for Computational Linguistics, Aug. 2019, pp. 319–327.

[16] Ö. Uzuner, B. R. South, S. Shen, and S. L. DuVall, "2010 i2b2/va challenge on concepts, assertions, and relations in clinical text," *Journal of the American Medical Informatics Association*, vol. 18, no. 5, pp. 552–556, 2011.

[17] M. Krallinger, O. Rabal, S. A. Akhondi, M. P. Pérez *et al.*, "Overview of the biocreative vi chemical-protein interaction track," in *Proceedings of the sixth BioCreative challenge evaluation workshop*, vol. 1, 2017, pp. 141–146.

[18] S. R. Bowman, G. Angeli, C. Potts, and C. D. Manning, "A large annotated corpus for learning natural language inference," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. ACL, 2015.

[19] A. Warstadt, A. Singh, and S. R. Bowman, "Neural network acceptability judgments," *arXiv preprint arXiv:1805.12471*, 2018.

[20] H. Nakayama, "seqeval: A python framework for sequence labeling evaluation," 2018.