Specializing Static and Contextual Embeddings in the Medical Domain Using Knowledge Graphs: Let's Keep It Simple

Hicham El Boukkouri[1], **Olivier Ferret**[2], **Thomas Lavergne**[1], **Pierre Zweigenbaum**[1]
[1]Université Paris-Saclay, CNRS, LISN, Orsay, France,
[2]Université Paris-Saclay, CEA, List, F-91120, Palaiseau, France,
{elboukkouri,lavergne,pz}@lisn.fr, olivier.ferret@cea.fr

Abstract

Domain adaptation of word embeddings has mainly been explored in the context of retraining general models on large specialized corpora. While this usually yields good results, we argue that knowledge graphs, which are used less frequently, could also be utilized to enhance existing representations with specialized knowledge. In this work, we aim to shed some light on whether such knowledge injection could be achieved using a basic set of tools: graph-level embeddings and concatenation. To that end, we adopt an incremental approach where we first demonstrate that static embeddings can indeed be improved through concatenation with in-domain *node2vec* representations. Then, we validate this approach on contextual models and generalize it further by proposing a variant of BERT that incorporates knowledge embeddings within its hidden states through the same process of concatenation. We show that this variant outperforms plain retraining on several specialized tasks, then discuss how this simple approach could be improved further. Both our code and pre-trained models are open-sourced for future research. In this work, we conduct experiments that target the medical domain and the English language.

1 Introduction

The popularization of transfer learning, particularly in the context of pre-training language models to serve as encoders in downstream tasks, has led to an ever-expanding set of methods for representing textual data: e.g. ELMo (Peters et al., 2018), BERT (Devlin et al., 2019), XLNet (Yang et al., 2019). While these models continuously push forward the expected level of performance on so-called "general domain" tasks (e.g. GLUE[1]), they usually lag behind when it comes to more specialized areas like the medical domain (see BLUE[2] and BLURB[3] benchmarks). As a result, there is a growing interest in finding ways in which these out-of-the-box representations can be specialized, with most efforts focusing on retraining general models on specialized corpora: e.g. ClinicalBERT (Alsentzer et al., 2019), BioBERT (Lee et al., 2020), and BioMed-RoBERTa (Gururangan et al., 2020). However, pre-trained language models have also been shown to benefit from external knowledge injection, with approaches like LIBERT (Lauscher et al., 2020), KnowBERT (Peters et al., 2019), and KEPLER (Wang et al., 2021b) in the general domain, or (Hao et al., 2020) and (Lu et al., 2021) in the medical domain. Yet, these efforts usually involve complex modifications to the architecture of underlying models and/or their pre-training procedure, which may convey the impression that knowledge injection cannot be achieved in simpler ways.

In this work, we propose a simple approach to embedding specialization that relies on knowledge graph embeddings and concatenation. We argue that by building a simple but strong baseline first, we lay the foundation for future improvements that may be easily achieved by upgrading to more sophisticated knowledge embeddings or combination methods. In practice, we show that medical concept embeddings obtained from an in-domain knowledge graph can be combined through concatenation with word representations to effectively construct specialized "meta-embeddings" (Yin and Schütze, 2016). Moreover, in the specific case of contextual embeddings, we show that these concept embeddings can be combined either externally, with a general-domain model, or internally, during the pre-training of a specialized model, to achieve varying levels of model specialization. All our models are trained and evaluated in pairs, and in exactly the same conditions, to highlight to the greatest extent the impact of our strategies.

[1] https://gluebenchmark.com/leaderboard

[2] https://github.com/ncbi-nlp/BLUE_Benchmark#baselines

[3] https://microsoft.github.io/BLURB/leaderboard.html

Proceedings of the 13th International Workshop on Health Text Mining and Information Analysis (LOUHI), pages 69 - 80
December 7, 2022 ©2022 Association for Computational Linguistics

Our contributions are the following:

- We build two sets of knowledge representations by applying *node2vec* (Grover and Leskovec, 2016) to concepts from MeSH (biomedical) and SNOMED CT (clinical).
- We construct specialized meta-embeddings by concatenating fastText embeddings (Bojanowski et al., 2017) with the *node2vec* vectors. We show that this improves the performance of both general and medical domain representations on several medical tasks.
- We conduct the same experiments with contextual BERT and CharacterBERT (El Boukkouri et al., 2020) representations, and show similar improvements on most evaluation tasks with a slight edge for the character-based model.
- We generalize the meta-embedding approach to the pre-training of contextual models by introducing a 'Knowledge Injection Module' that combines incoming hidden states from a Transformer layer (Vaswani et al., 2017) with external knowledge representations through the same process of concatenation.
- We retrain both original and modified versions of BERT and CharacterBERT on a medical corpus and show that the modified models perform better on several medical tasks.
- We propose improvements to our methods and share our code and pre-trained models to facilitate future attempts at enhancing word embeddings using knowledge graphs.

Our experiments are conducted on general and medical corpora in the English language. Generalization to other cases is left for future work.

2 Related Work

Our approach is related to the similar but usually distinct topics of knowledge injection and domain adaptation. In fact, most attempts at domain adaptation do not aim to inject external knowledge directly into models but rather indirectly, through retraining on specialized corpora, as this is known to bring significant improvements when such in-domain corpora are available (Si et al., 2019). On the other hand, research concerned with knowledge injection usually tackles the problem within the same domain. For instance, SemBERT (Zhang et al., 2020), COMET (Bosselut et al., 2019), ERNIE (Zhang et al., 2019), K-BERT (Liu et al., 2020), and KEPLER all inject general knowledge into general-domain models. Similar efforts in the medical domain (Hao et al., 2020; He et al., 2020a; Michalopoulos et al., 2021; Lu et al., 2021) directly inject medical knowledge during medical pre-training. In this work, we first set out to determine whether the performance of general-domain models, both static and contextual, can be improved solely using specialized knowledge embeddings, then only do we incorporate this approach into the usual model adaptation via pre-training.

Methods that utilize knowledge graphs, for instance (Roy and Pan, 2021; Sharma et al., 2019; Chang et al., 2021), can be broadly grouped into two categories: those that use the structured data directly and those that encode this data into numerical representations. Instances of direct utilization include KG-BERT (Yao et al., 2019) where triples (`concept_1`, `relation`, `concept_2`) are used to inject BERT with medical information through auxiliary tasks like knowledge graph completion and triple classification. Entity linking in (Yuan et al., 2021) or more specialized tasks in (He et al., 2020c) are also used as auxiliary tasks for performing such injection. While these methods can be effective, we argue that an indirect approach is desirable as it presents the specialized knowledge in the same format as the word embeddings, thus reformulating knowledge injection as a meta-embedding problem.

Meta-embeddings combine two or more underlying sets of embeddings into a single final representation. There are many approaches to meta-embeddings like Dynamic Meta Embeddings (DME, Kiela et al. (2018)) where each embedding is projected down to the same dimension before being used in a linear combination, or Word Prisms (He et al., 2020b), which further improve upon DMEs by enforcing desirable orthogonality properties during training. In this work, we use a simple but strong baseline for meta-embeddings—concatenation—which ensures that both word and knowledge information is accessible at all times. More sophisticated approaches, although likely to improve our overall performance, are left for future work.

3 Knowledge Graph Representations

In order to use concatenation to specialize word embeddings with a knowledge base, we first need to be able to convert this knowledge base into dense numerical representations. There are several meth-

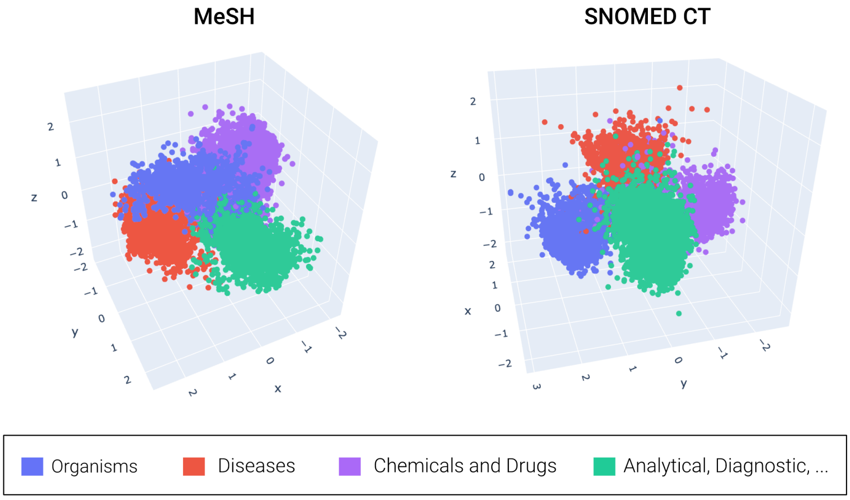

Figure 1: PCA of MeSH and SNOMED embeddings for four categories of medical concepts.

ods for embedding knowledge graphs (e.g. RotatE (Sun et al., 2019), TuckER (Balazevic et al., 2019)), and these usually produce multifaceted relation-dependent concept representations. However, for simplicity, we only consider a single relation which enables us to use a graph-level method instead, namely *node2vec* (Grover and Leskovec, 2016).

3.1 UMLS, MeSH, and SNOMED CT

The Unified Medical Language System (UMLS) (Lindberg et al., 1993) includes a meta-thesaurus that contains multiple subsets (called vocabularies) that organize specific groups of medical concepts according to a large number of varied relationships (e.g. `active_ingredient_of`, `associated_with`, `branch_of`). Among the many vocabularies in the UMLS, we use the Medical Subject Headings (MeSH)[4], which mainly organizes concepts from the biomedical domain, as well as the Systematized Nomenclature Of Medicine - Clinical Terms (SNOMED_CT),[5] which also has a coverage of the clinical domain. Given both vocabularies, we query[6] the UMLS and recover all pairs of Concept Unique Identifiers (CUI) for concepts related through the `is_a` relation (e.g. `Chronic Bronchitis` is a `Chronic disease`). Although many more types of relations are available, we focus on the single most frequent type `is_a`, which also allows us to extract a single graph and use a graph-level method like *node2vec*.

3.2 Dense Representations with *node2vec*

The *node2vec* (Grover and Leskovec, 2016) method effectively applies a *word2vec* (Mikolov et al., 2013) objective to learn node representations from a set of node sequences that are generated randomly using a flexible type of random walks on the knowledge graph. Running the official `Python` implementation[7] with default parameters allows us to learn 256-dimensional dense representations for each node of the provided graphs. This step yields 29,738 CUI embeddings for MeSH concepts and 389,872 CUI embeddings for SNOMED with 15,418 overlapping CUIs having both a MeSH and SNOMED representation. The visualization of these embeddings using a PCA (see Figure 1) shows that this method is able to separate different categories of medical concepts in different subspaces, which suggests some level of encoded medical knowledge.

Using *node2vec* Embeddings in Practice For each possible CUI, we concatenate both sets of knowledge embeddings and use zero-padding when a CUI does not appear in either MeSH or SNOMED. This produces a final 512-dimensional knowledge representation for each concept. However, using these representations in practice requires locating concept mentions in texts, which refers to the task of concept normalization. This normalization aims to identify the various linguistic forms that a given concept can take, which we perform in our case by running an exact string matching between the reference linguistic forms from the

[4] https://www.nlm.nih.gov/mesh/meshhome.html

[5] https://www.nlm.nih.gov/healthit/snomedct/index.html

[6] SQL scripts are available in our code repository.

[7] https://github.com/aditya-grover/node2vec

UMLS[8] and the target texts. Ultimately, the tokens from each mention are assigned the *node2vec* representation of their concept, with out-of-mention tokens getting an empty zero-valued vector instead.

4 Embedding-Specialization Methods

4.1 Static Representations

To determine whether word embeddings can be successfully specialized using in-domain knowledge representations, we first conduct experiments on static embeddings. In particular, we learn word representations using fastText[9] (Bojanowski et al., 2017) and then attempt to specialize these representations by concatenating fastText and node2vec vectors at the token level. We consider the following corpora for learning word embeddings:

Gigaword (Graff et al., 2003): a newswire corpus constructed from many sources including the New York Times. This is a general domain corpus with $\approx$ 1 billion tokens.

PubMed (MEDLINE): scientific abstracts from the biomedical literature.[10] This is a medical domain corpus with $\approx$ 2 billion tokens.

MIMIC (Johnson et al., 2016): clinical notes from several hospitals.[11] This is a medical domain corpus with $\approx$ 0.5 billion tokens.

4.2 Contextual Representations

We also experiment with contextual embeddings, namely: BERT (Devlin et al., 2019) and CharacterBERT (El Boukkouri et al., 2020).[12] The former is included as a strong baseline for transformer-based embeddings and the latter is included as it produces word-level representations and seems to perform well in the medical domain. Furthermore, considering these two models allows us to have a larger sample size for measuring the impact of our strategies on transformer-based models.

We specialize contextual embeddings in two ways: either externally, via token-level concatenation similar to static embeddings; or internally, by introducing the following specialization layers.

Knowledge Injection Modules (KIM) These are small layers that generalize the idea of concatenating word and knowledge embeddings to the internal states of a transformer-based model. When placed after a given layer, this module concatenates the hidden representations from that layer h_i with their corresponding knowledge representations KG_i. Then, it projects this concatenation to recover a set of "enhanced states" $\mathbb{h}_i$ with the same dimensionality as the original hidden representations. Since this operation may lose some of the information from the original hidden states, we compute a mixture of the enhanced and original states with trainable parameters $\alpha \in [0, 1]$ and $\beta = 1 - \alpha$. The final output $\boldsymbol{h}_i$ is fed to the next layer. In summary:

$$\boldsymbol{h}_i = \alpha\,\mathbb{h}_i + \beta\,\mathrm{h}_i$$

where $\mathbb{h}_i = [\mathrm{h}_i; \mathrm{KB}_i]\, W + b$ and W, b are respectively the weight matrix and bias of the linear projection operation (see Figure 2).

Our KIMs are loosely related to Adapter Modules (Houlsby et al., 2019; Wang et al., 2021a) but are conceptually simpler and only focus on incorporating external representations into the hidden states of transformer-based models.

5 Experiments

5.1 Embedding Models

Our final embeddings come in five configurations:

Random: randomly initialized 256-dimensional static embeddings used as a baseline for static word representations.

***Model*:** either 256-dimensional static embeddings of the form "fastText(*corpus*)" where *corpus* is one of the corpora presented in section 4.1, or a 768-dimensional BERT or CharacterBERT model.

[*Model*, node2vec]: token-level concatenation of *Model* with the pre-trained 512-dimensional node2vec representations from Section 3.2.

***Model*(medical):** a transformer model adapted via pre-training on a large medical corpus that consists of $\approx$ 0.5 billion tokens from MIMIC-III clinical notes and $\approx$ 0.5 billion tokens from abstracts extracted from PMC-OA[13] biomedical articles.

[8] These synonyms are available in the MRCONSO table.

[9] Training scripts are available in our code repository.

[10] Available at: https://www.nlm.nih.gov/databases/download/pubmed_medline.html

[11] Available at: https://physionet.org/content/mimiciii-demo/1.4/

[12] We use the "base-uncased" versions of these models.

[13] https://www.ncbi.nlm.nih.gov/pmc/tools/openftlist/

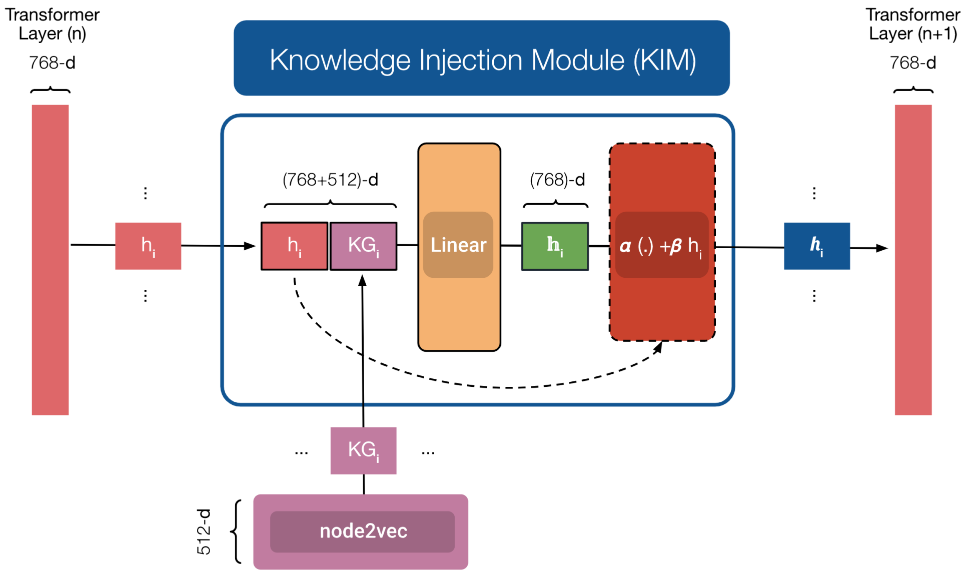

Figure 2: Detailed view of a Knowledge Injection Module (KIM) between two Transformer layers. Given an incoming hidden (h_i) and knowledge representation (KG_i), the module concatenates both vectors ($[\mathrm{h}_i; \mathrm{KG}_i]$), applies a linear projection down to the original size ($\mathbb{h}_i$), then computes a mixture of the enhanced and original states using parameters $\alpha \in [0, 1]$ and $\beta = 1 - \alpha$. The output ($\boldsymbol{h}_i$) is ultimately fed to the next Transformer layer.

Enhanced*Model*(medical): same as the configuration above but this time, the architecture is changed to use a KIM after each transformer layer, as well as either the WordPiece embeddings (Wu et al., 2016) for BERT, or Character-CNN (Peters et al., 2018) for CharacterBERT.

For the last two configurations, we follow a standard pre-training procedure comprising Masked Language Modeling (MLM) and Next Sentence Prediction (NSP), and adapt the implementation from El Boukkouri et al. (2020) while keeping the same hyper-parameters.[14]

5.2 Evaluation Tasks

Insights from model evaluation can be misleading, especially when only a few tasks are considered. To conduct a thorough evaluation of our models, we consider multiple tasks from both the biomedical and clinical domains (see Table 1):

i2b2 This is the i2b2/VA 2010 clinical concept extraction task (Uzuner et al., 2011), which is a sequence labeling task that aims to detect three categories of clinical entities: PROBLEM (e.g. “headache"), TREATMENT (e.g. “oxycodone") and TEST (e.g. “MRI"). The exact match F1-score is used as an evaluation metric.

BC5-Disease/Chemical These are two sequence labeling tasks from BioCreative V CDR (Li et al., 2016), which respectively aim to detect DISEASE (e.g. “hepatitis") and CHEMICAL (e.g. “corticosteroid") entities. The exact F1 is used as a metric.

DDI This is a relation extraction task from SemEval 2013 - Task 9.2. (Herrero-Zazo et al., 2013), which focuses on classifying drug-drug interactions into five categories: ADVISE (DDI-advise), EFFECT (DDI-effect), MECHANISM (DDI-mechanism), INTERACTION (DDI-int), and DDI-false for no interaction. The micro-averaged F1 over all four non-negative classes is used as a metric.

ChemProt This is a relation extraction task from BioCreative VI (Krallinger et al., 2017), which focuses on classifying chemical-protein relations into six categories: ACTIVATOR (CPR:3), INHIBITOR (CPR:4), AGONIST (CPR:5), ANTAGONIST (CPR:6), SUBSTRATE (CPR:9) and FALSE for no relation. The micro-averaged F1-score over non-negative classes is used as a metric.

BIOSSES This is a small sentence similarity dataset in the biomedical domain (Soğancıoğlu et al., 2017). The Pearson correlation of predicted and gold similarities is used as a metric.

ClinicalSTS This is a clinical sentence similarity task from the OHNLP Challenge 2018 (Wang et al., 2018). It uses Pearson correlation as well.

[14] Specifically, we use the parameters at this URL.

	i2b2	BC5-Disease	BC5-Chemical	ChemProt	DDI	BIOSSES	ClinicalSTS	MEDNLI
Train	22,263	4,182	5,203	4,154	2,937	64	600	11,232
Val.	5,565	4,244	5,347	2,416	1,004	16	150	1,395
Test	45,009	4,424	5,385	3,458	979	20	318	1,422

Table 1: Number of examples (entities, positive relations, or samples) for each evaluation task.

MedNLI This is a clinical natural language inference task (Romanov and Shivade, 2018), which aims to classify pairs of sentences into three categories: ENTAILMENT, CONTRADICTION, and NEUTRAL. The classification accuracy is used as a metric.

5.3 Evaluation Architectures

We use different architectures depending on the model and fine-tuning tasks at hand.

Sequence Labeling The architecture for tagging uses an encoder followed by a classification layer and a CRF (Lafferty et al., 2001). The encoder changes according to the type of input embeddings: **fastText** and **[fastText, node2vec]** are fed to a Bi-LSTM,[15] variants of **BERT** are their own encoders, and variants of **[BERT, node2vec]** concatenate knowledge (node2vec) embeddings with token (BERT) representations and feed it forward.

Classification The architecture for relation extraction is similar but requires a summarized representation at the example level to be fed to a classification layer. Here again, **fastText** and **[fastText, node2vec]** are fed to a Bi-LSTM, but this time, the output is average-pooled to produce a single feature vector. With variants of **BERT**, the pooler output is used. Finally, when using variants of **[BERT, node2vec]**, the knowledge representations are average-pooled before being concatenated with the pooler representation.

Sentence Similarity For STS tasks, we use a different approach for static and contextual embeddings. For **static embeddings**, we compute a bag-of-word representation for each sentence, then measure the cosine similarity between the two representations. When **contextual embeddings** are involved, we treat the task as a regression problem and use the same encoder as for classification.

Natural Language Inference For NLI tasks, we require a summarized representation at the sentence-pair level that we can ultimately feed to a classification layer. For **static embeddings**, we compute an average-pooled Bi-LSTM representation for the first sentence u as well as for the second one v, then compute a global feature vector $[u, v, |u - v|, u * v]$ following the approach of InferSent (Conneau et al., 2017). When using variants of **BERT**, we simply use the pooler representation as these models can accept sentence pairs. Finally, with variants of **[BERT, node2vec]**, we concatenate the pooler output with InferSent-style features computed from the node2vec vectors.

5.4 Evaluation Method

Optimization All parameters (including static and knowledge embeddings) are fine-tuned using the following hyper-parameters:

- **Validation Ratio:** when no validation set is available, we use 20% of the training data.
- **Epochs:** we run 15 epochs for all tasks, except for BIOSSES and ClinicalSTS for which we run 100 and 50 epochs respectively.
- **Batch Size:** we use batches of 32 examples.
- **Optimizer & Learning Rate:** we use the Adam optimizer (Kingma and Ba, 2015) with a learning rate of 1e-3 for non-transformer weights and a learning rate of 3e-5 for transformer weights. We also use a weight decay of 10% and a linear schedule with a 10% warmup for transformer weights.

Model Ensembles To account for some of the randomnesses during fine-tuning, we evaluate each model on each task using 10 different random seeds. Given these single models, we compute ensembles using a majority vote, except for STS tasks where we use the average similarity instead. Then, to account for the variance of the ensembles as well, we compute 10 different ensembles by excluding a single seed from the ensemble set and repeating this process. The average ensemble score is then used as the final performance for the model.

Statistical Significance We use Almost Stochastic Order (ASO) tests from Dror et al. (2019) in an attempt to rigorously compare our models. In this framework, the test takes a set of scores from

[15] All future mentions of a Bi-LSTM refer to a 3-layer network with 50% recurrent dropout and an output size of 512.

	i2b2	bc5-disease	bc5-chemical	chemprot	ddi	biosses	clinical_sts	mednli
Random	83.42	74.12	75.91	51.72	64.92	**64.42**	62.09	**70.68**
[Random, node2vec]	**84.57**	**80.93**	**84.91**	**53.66**	**65.14**	59.83	**63.72**	70.36
fastText(Gigaword)	84.70	76.79	82.76	52.06	62.83	**82.43**	**71.93**	69.66
[fastText(Gigaword), node2vec]	**84.99**	**80.86**	**86.50**	**52.64**	**64.44**	53.55	65.95	**70.08**
fastText(PubMed)	85.16	79.71	88.67	**54.62**	66.17	**91.49**	**72.10**	70.51
[fastText(PubMed), node2vec]	**85.49**	**82.62**	**89.45**	54.36	**67.11**	62.32	67.46	**70.82**
fastText(MIMIC)	85.49	78.92	84.93	51.54	67.12	**76.94**	**72.42**	**71.74**
[fastText(MIMIC), node2vec]	**86.01**	**80.70**	**86.38**	**52.90**	**67.59**	51.72	69.17	70.96
BERT	**88.16**	79.56	88.63	**71.75**	**79.95**	81.94	**84.71**	**79.75**
[BERT, node2vec]	87.76	**80.66**	**88.88**	71.36	79.60	**85.93**	84.34	79.30
BERT(medical)	**89.45**	81.88	**90.67**	**71.96**	**79.79**	89.14	**84.21**	**83.66**
EnhancedBERT(medical)	89.40	**83.48**	90.36	71.22	78.74	**92.12**	83.59	83.03
CharacterBERT	**88.08**	80.90	88.73	70.61	79.42	90.58	84.49	78.85
[CharacterBERT, node2vec]	87.81	**81.63**	**89.39**	**71.01**	**81.23**	**91.03**	**84.89**	**79.19**
CharacterBERT(medical)	**89.82**	83.60	92.07	**73.63**	**80.67**	87.52	83.63	**84.66**
EnhancedCharacterBERT(medical)	89.76	**85.05**	**92.08**	73.01	79.39	**92.65**	**84.42**	84.46

Table 2: Performance of model ensembles on evaluation tasks from the medical domain. Results are displayed in pairs: baseline model on the top line and specialized version (either through concatenation or KIM) on the bottom line. The colors show statistical significance, with bluer colors meaning the specialized models improve more significantly over the baselines and redder colors showing a more significant degradation in performance.

a model A and a model B, then returns a value $\epsilon \in [0, 1]$ that quantifies the stochastic order between A and B, with $\epsilon = 0$ meaning A $\succeq$ B, $\epsilon = 1$ meaning B $\succeq$ A, and $\epsilon = 0.5$ meaning that no stochastic order can be found for A and B.

6 Results and Discussion

For better visibility and given the large number of experiments, we present our results in pairs composed of a baseline and a specialized version of that baseline. We report the performances of each model pair as a set of two consecutive rows with the baseline on top (see Table 2). We also emphasize in bold the best performance on each task (column) and color the specialized version according to its ASO distance (ϵ) to the baseline model.[16]

Random vs. [Random, node2vec] It is interesting to note that randomly initialized static embeddings manage to achieve reasonable results, sometimes even outperforming pre-trained fastText representations (see Random vs. Gigaword or PubMed on MedNLI). However, given the random nature of these vectors, we can easily expect in-domain knowledge representations to be able to improve the performance on downstream specialized tasks. While this is verified in most situations, we note a degradation on BIOSSES and MedNLI. This could point to situations where external knowledge is not relevant to the task at hand.

fastText(X) vs. [fastText(X), node2vec] Overall, using concatenation to combine knowledge representations with fastText embeddings seems to result in consistent gains, notably on tagging and classification tasks (see the top-left section of the table). Moreover, these results seem to hold regardless of the domain of origin, as word embeddings trained on Gigaword (general domain), PubMed (biomedical domain), and MIMIC (clinical domain) all seem to benefit from this combination. However, we can see that the results on STS are significantly worse with drops of up to 30 points of correlation on BIOSSES with fastText(Gigaword). This may be due to the “bag-of-word + cosine similarity”

[16] Colors range from red ($\epsilon = 0$) for a significant degradation, to blue ($\epsilon = 1$) for a significant improvement.

approach not being suited for meta-embeddings made of both word and knowledge representations, especially since the node2vec vectors are rather sparse (most concepts do not have both a MeSH and SNOMED representation) and twice as large as the word representations.

BERT vs. [BERT, node2vec] When looking at the results for contextual embeddings, we can see several instances where the concatenation with *node2vec* proves to be beneficial. However, there seems to be a discrepancy where sometimes this concatenation does improve the CharacterBERT baseline on one hand but impairs the BERT baseline on the other (see ChemProt and DDI). A closer look at these cases shows that plain CharacterBERT performs slightly lower than plain BERT in these situations, which may mean that the knowledge representations compensate for any information that may be missing in the baseline CharacterBERT model, relative to the task.

BERT(medical) vs. EnhancedBERT(medical) The addition of KIMs seems to give variable results depending on the evaluation task. In fact, we can see that EnhancedBERT and EnhancedCharacterBERT respectively lose 1.05 and 1.28 F1 relative to their baselines on the DDI task, however, we also see that these same models gain 1.6 and 1.45 F1 on the BC5-Disease task. Incidentally, the BC5 tasks are interesting as they use the exact same corpus but focus on two different types of entities: DISEASE and CHEMICAL. Therefore, given that EnhancedBERT(medical) performs better than BERT(medical) on BC5-Disease and worse on BC5-Chemical, we can safely assume that this is not due to the KIMs being particularly harmful but rather to the information available in the knowledge representations being, relative to what is already available in the base model, more relevant for the first task than for the second one. Consequently, we may assume that the KIMs can successfully incorporate external information into a model but that the downstream performance may depend on the relevance of this information for any given task.

Observed Trends All in all, we notice that the best models remain either BERT or CharacterBERT-based models and that the addition of external knowledge to static representations is not sufficient to make them outperform their contextual counterparts. This is globally true with a few exceptions. In fact, we may observe in the case of sequence labeling tasks (i2b2 and BC5-Disease/Chemical) that the addition of knowledge is often beneficial for static models. The matter is more complex for contextual models however, where the benefits are less clear but for which it may still be desirable to use external knowledge as any potential degradation seems to be relatively minor. In the case of relation classification tasks (ChemProt and DDI), leveraging external knowledge is once again positive for static models but seems to be harmful to some Transformer-based models (especially BERT). Finally, for semantic similarity and inference tasks (BIOSSES, ClinicalSTS, and MedNLI), we may not recommend using our methods as any existing gains are relatively small when compared to the potential losses, although there may be some benefit for contextual models. Overall, we can see that our knowledge enhancement methods, either by external concatenation or through KIMs, always benefit CharacterBERT with appreciable gains in performance: choosing CharacterBERT with KIMs ensures obtaining the highest performance or being very close to it.

	all	no	some	full	homog
fastText(PubMed)	+3.3	-4.6	+4.5	+5.1	+6.2
CharacterBERT	+0.3	-1.7	+0.6	+0.9	+1.1
EnhancedCBERT	+1.4	-1.7	+1.8	+2.1	+2.5

Table 3: Variations (percentages) of True Positives for the BC5-Disease task according to the coverage of the gold entities by concepts of our knowledge graph.

Contribution of the Knowledge Graph To measure the contribution of external knowledge, specifically in the case of sequence labeling tasks, we compute, for each gold entity of the test set, the average change in *true positives* brought by the use of the knowledge embeddings. To dig a bit deeper, we compute this change in buckets with varying the degrees of coverage of gold entities by a concept of the knowledge graph: **no** coverage; **some** coverage; all the tokens are **full**y covered; and finally, a full and **homog**eneous coverage (i.e. same CUI everywhere). We display the results for BC5-Disease and three different models in Table 3: fastText(PubMed) and CharacterBERT, which both rely on token-level concatenations, and EnhancedCharacterBERT (EnhancedCBERT), which leverages KIMs. While the over**all** contribution is positive, we can see that this effect increases with the coverage of gold entities by the knowledge base.

Moreover, when the coverage is null, the impact becomes negative, emphasizing the importance of choosing a complete and adequate knowledge base when using such knowledge injection methods.

7 Conclusion and Future Work

In this paper, we focused on exploring the extent to which specialized information from a knowledge graph could be injected into existing word embeddings using a very simple set of tools: graph embeddings and concatenation. While focusing on the medical domain in the English language, we conducted multiple evaluations on tasks ranging from entity recognition to sentence similarity. These evaluations demonstrated that concatenation with in-domain graph representations can be a simple yet effective approach to model specialization, with significant gains on several tasks. Moreover, applying the same process of concatenation within transformer-based contextual models proved beneficial as well, with notable improvements using Knowledge Injection Modules (KIMs) on several downstream tasks.

As mentioned in Section 3.1, many more types of relations beyond `is_a` could be used to improve the quality of the generated knowledge representations. An interesting path for future work may be to use recent meta-embedding methods like Word Prisms to learn multifaceted knowledge representations from multiple underlying representations corresponding to two or more types of relations.

8 Acknowledgements

This work was funded by the French National Research Agency (ANR) under the ADDICTE project (ANR-17-CE23-0001) and was performed using HPC resources from GENCI-IDRIS (Grant AD011011698R1).

References

Emily Alsentzer, John Murphy, William Boag, Wei-Hung Weng, Di Jindi, Tristan Naumann, and Matthew McDermott. 2019. Publicly available clinical BERT embeddings. In *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, pages 72–78, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Ivana Balazevic, Carl Allen, and Timothy Hospedales. 2019. TuckER: Tensor factorization for knowledge graph completion. 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 5185–5194, Hong Kong, China. Association for Computational Linguistics.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi. 2019. COMET: Commonsense transformers for automatic knowledge graph construction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4762–4779, Florence, Italy. Association for Computational Linguistics.

David Chang, Eric Lin, Cynthia Brandt, and Richard Andrew Taylor. 2021. Incorporating domain knowledge into language models by using graph convolutional networks for assessing semantic textual similarity: Model development and performance comparison. *JMIR medical informatics*, 9(11):e23101.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680, Copenhagen, Denmark. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Rotem Dror, Segev Shlomov, and Roi Reichart. 2019. Deep dominance - how to properly compare deep neural models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2773–2785, Florence, Italy. Association for Computational Linguistics.

Hicham El Boukkouri, Olivier Ferret, Thomas Lavergne, Hiroshi Noji, Pierre Zweigenbaum, and Jun'ichi Tsujii. 2020. CharacterBERT: Reconciling ELMo and BERT for word-level open-vocabulary representations from characters. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6903–6915, Barcelona, Spain (Online). International Committee on Computational Linguistics.

David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. 2003. English gigaword. *Linguistic Data Consortium, Philadelphia*, 4(1):34.

Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.

Boran Hao, Henghui Zhu, and Ioannis Paschalidis. 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). International Committee on Computational Linguistics.

Bin He, Di Zhou, Jinghui Xiao, Xin Jiang, Qun Liu, Nicholas Jing Yuan, and Tong Xu. 2020a. 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. Association for Computational Linguistics.

Jingyi He, Kc Tsiolis, Kian Kenyon-Dean, and Jackie Chi Kit Cheung. 2020b. Learning efficient task-specific meta-embeddings with word prisms. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1229–1241, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Yun He, Ziwei Zhu, Yin Zhang, Qin Chen, and James Caverlee. 2020c. Infusing Disease Knowledge into BERT for Health Question Answering, Medical Inference and Disease Name Recognition. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4604–4614, Online. Association for Computational Linguistics.

María Herrero-Zazo, Isabel Segura-Bedmar, Paloma Martínez, and Thierry Declerck. 2013. The DDI corpus: An annotated corpus with pharmacological substances and drug–drug interactions. *Journal of biomedical informatics*, 46(5):914–920.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In *International Conference on Machine Learning*, pages 2790–2799. PMLR.

Alistair EW Johnson, Tom J Pollard, Lu Shen, H Lehman Li-Wei, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. 2016. MIMIC-III, a freely accessible critical care database. *Scientific data*, 3(1):1–9.

Douwe Kiela, Changhan Wang, and Kyunghyun Cho. 2018. Dynamic meta-embeddings for improved sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1466–1477, Brussels, Belgium. Association for Computational Linguistics.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *ICLR*.

Martin Krallinger, Obdulia Rabal, Saber A Akhondi, Martın Pérez Pérez, Jesús Santamaría, Gael Pérez Rodríguez, et al. 2017. Overview of the bioCreative VI chemical-protein interaction track. In *Proceedings of the sixth BioCreative challenge evaluation workshop*, volume 1, pages 141–146.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning (ICML 2001)*, pages 282–289, Williams College, Williamstown, MA, USA. Morgan Kaufmann.

Anne Lauscher, Ivan Vulić, Edoardo Maria Ponti, Anna Korhonen, and Goran Glavaš. 2020. Specializing unsupervised pretraining models for word-level semantic similarity. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1371–1383, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Jiao Li, Yueping Sun, Robin J Johnson, Daniela Sciaky, Chih-Hsuan Wei, Robert Leaman, Allan Peter Davis, Carolyn J Mattingly, Thomas C Wiegers, and Zhiyong Lu. 2016. BioCreative v CDR task corpus: a resource for chemical disease relation extraction. *Database*, 2016.

D. A. Lindberg, B. L. Humphreys, and A. T. McCray. 1993. The unified medical language system. *Methods of information in medicine*, 32(4):281–291.

Weijie Liu, Peng Zhou, Zhe Zhao, Zhiruo Wang, Qi Ju, Haotang Deng, and Ping Wang. 2020. K-BERT: Enabling language representation with knowledge graph. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(03):2901–2908.

Qiuhao Lu, Dejing Dou, and Thien Huu Nguyen. 2021. Parameter-efficient domain knowledge integration from multiple sources for biomedical pre-trained language models. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3855–3865, Punta Cana, Dominican Republic. Association for Computational Linguistics.

George Michalopoulos, Yuanxin Wang, Hussam Kaka, Helen Chen, and Alexander Wong. 2021. Umls-BERT: Clinical Domain Knowledge Augmentation of Contextual Embeddings Using the Unified Medical Language System Metathesaurus. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1744–1753, Online. Association for Computational Linguistics.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Matthew E. Peters, Mark Neumann, Robert Logan, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A. Smith. 2019. Knowledge enhanced contextual word representations. 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 43–54, Hong Kong, China. Association for Computational Linguistics.

Alexey Romanov and Chaitanya Shivade. 2018. Lessons from natural language inference in the clinical domain. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1586–1596, Brussels, Belgium. Association for Computational Linguistics.

Arpita Roy and Shimei Pan. 2021. Incorporating medical knowledge in BERT for clinical relation extraction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5357–5366, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Soumya Sharma, Bishal Santra, Abhik Jana, Santosh Tokala, Niloy Ganguly, and Pawan Goyal. 2019. Incorporating Domain Knowledge into Medical NLI using Knowledge Graphs. 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 6092–6097, Hong Kong, China. Association for Computational Linguistics.

Yuqi Si, Jingqi Wang, Hua Xu, and Kirk Roberts. 2019. Enhancing clinical concept extraction with contextual embeddings. *Journal of the American Medical Informatics Association*, 26(11):1297–1304.

Gizem Soğancıoğlu, Hakime Öztürk, and Arzucan Özgür. 2017. BIOSSES: a semantic sentence similarity estimation system for the biomedical domain. *Bioinformatics*, 33(14):i49–i58.

Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2019. RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space. In *International Conference on Learning Representations*.

Özlem Uzuner, Brett R South, Shuying Shen, and Scott L DuVall. 2011. 2010 i2b2/VA challenge on concepts, assertions, and relations in clinical text. *Journal of the American Medical Informatics Association*, 18(5):552–556.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Ruize Wang, Duyu Tang, Nan Duan, Zhongyu Wei, Xuanjing Huang, Jianshu Ji, Guihong Cao, Daxin Jiang, and Ming Zhou. 2021a. K-Adapter: Infusing Knowledge into Pre-Trained Models with Adapters. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1405–1418, Online. Association for Computational Linguistics.

Xiaozhi Wang, Tianyu Gao, Zhaocheng Zhu, Zhengyan Zhang, Zhiyuan Liu, Juanzi Li, and Jian Tang. 2021b. KEPLER: A Unified Model for Knowledge Embedding and Pre-trained Language Representation. *Transactions of the Association for Computational Linguistics*, 9:176–194.

Yanshan Wang, Naveed Afzal, Sijia Liu, Majid Rastegar-Mojarad, Liwei Wang, Feichen Shen, Sunyang Fu, and Hongfang Liu. 2018. Overview of the BioCreative/OHNLP challenge 2018 task 2: clinical semantic textual similarity. *Proceedings of the BioCreative/OHNLP Challenge*, 2018.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. XLNet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. KG-BERT: BERT for knowledge graph completion. *arXiv preprint arXiv:1909.03193*.

Wenpeng Yin and Hinrich Schütze. 2016. Learning word meta-embeddings. In *Proceedings of the 54th Annual Meeting of the Association for Computational*

Linguistics (Volume 1: Long Papers), pages 1351–1360, Berlin, Germany. Association for Computational Linguistics.

Zheng Yuan, Yijia Liu, Chuanqi Tan, Songfang Huang, and Fei Huang. 2021. Improving biomedical pretrained language models with knowledge. In *Proceedings of the 20th Workshop on Biomedical Language Processing*, pages 180–190, Online. Association for Computational Linguistics.

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 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. Association for Computational Linguistics.

Zhuosheng Zhang, Yuwei Wu, Hai Zhao, Zuchao Li, Shuailiang Zhang, Xi Zhou, and Xiang Zhou. 2020. Semantics-aware bert for language understanding. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):9628–9635.