Enhancing Scholarly Understanding: A Comparison of Knowledge Injection Strategies in Large Language Models

Andrea Cadeddu[1], Alessandro Chessa[1], Vincenzo De Leo[1,2,*], Gianni Fenu[2], Enrico Motta[3], Francesco Osborne[3,4], Diego Reforgiato Recupero[2], Angelo Salatino[3] and Luca Secchi[1,2]

[1]*Linkalab s.r.l., Cagliari, Italy*
[2]*Department of Mathematics and Computer Science, University of Cagliari, Cagliari, Italy*
[3]*Knowledge Media Institute, The Open University, Milton Keynes, United Kingdom*
[4]*Department of Business and Law, University of Milano Bicocca, Milan, Italy*

Abstract

The use of transformer-based models like BERT for natural language processing has achieved remarkable performance across multiple domains. However, these models face challenges when dealing with very specialized domains, such as scientific literature. In this paper, we conduct a comprehensive analysis of knowledge injection strategies for transformers in the scientific domain, evaluating four distinct methods for injecting external knowledge into transformers. We assess these strategies in a single-label multi-class classification task involving scientific papers. For this, we develop a public benchmark based on 12k scientific papers from the AIDA knowledge graph, categorized into three fields. We utilize the Computer Science Ontology as our external knowledge source. Our findings indicate that most proposed knowledge injection techniques outperform the BERT baseline.

Keywords

Knowledge Graphs, Natural Language Processing, BERT, Classification Tasks, Feature Engineering

1. Introduction

Transformer models, such as BERT [1] and GPT-4[1], have achieved state-of-the-art performance in a range of natural language processing tasks. However, despite these advancements, they still encounter considerable limitations, particularly in dealing with intricate concepts specific to specialized domains. This issue becomes particularly pronounced in the field of scientific research, which demands a nuanced grasp of highly specific concepts and their relations [2]. A key challenge in this context is the accurate classification of scientific articles [3]. This task is

Workshop at ISWC 2023 on Deep Learning for Knowledge Graphs, 6-7 November, 2023, Athens
*Corresponding author.
andrea.cadeddu@linkalab.it (A. Cadeddu); alessandro.chessa@linkalab.it (A. Chessa); vincenzo.deleo@linkalab.it (V. D. Leo); fenu@unica.it (G. Fenu); enrico.motta@open.ac.uk (E. Motta); francesco.osborne@open.ac.uk (F. Osborne); diego.reforgiato@unica.it (D. R. Recupero); angelo.salatino@open.ac.uk (A. Salatino); luca.secchi@linkalab.it (L. Secchi)

[1]https://arxiv.org/abs/2303.08774

crucial for structuring and retrieving scientific knowledge, thereby supporting researchers in staying up-to-date with the latest advancements [4].

A prevalent method to enhance the competence of transformers within specific domains involves continuous pretraining on domain-specific documents [5]. However, this approach poses significant hurdles, particularly due to the need to process a significant volume of unlabeled, domain-specific text to effectively fine-tune the model parameters [6]. To address these limitations and increase the accuracy of transformers within specialized domains, researchers have begun to explore the paradigm of knowledge injection [7]. These techniques incorporate external knowledge into transformer models, with the aim of enhancing their comprehension and, subsequently, their performance in pertinent tasks. They can handle a variety of structured data, but knowledge graphs (KGs) are now emerging as the prevalent choice [8].

In this paper, we analyse four primary strategies for integrating external knowledge into transformers and evaluate their effectiveness in the task of scientific article classification. For this purpose, we introduce a new benchmark for scientific article classification using 12k articles from AIDA KG [9] linked to the Computer Science Ontology (CSO) [10, 11]. Our findings reveal interesting insights about the efficacy of different strategies and their impact on scientific text classification, with a hybrid approach exploiting BERT and an MLP architecture yielding the best performance.

2. Related Work

The limitations of transformers prompted the research community to develop several strategies for knowledge injection [12, 13]. Yang et al. [7] proposed a classification for Knowledge-Enhanced Pre-trained Transformers (KEPTs), based on knowledge granularity, injection method, and degree of symbolic knowledge parameterization.

Having referred exclusively to the classification based on the knowledge injection method (KIM), this study focuses on two types of KEPT: i) data-structure-unified and ii) embedding-combined KEPTs, which are most pertinent in low-resource settings. *Data-structure-unified* KEPTs transform knowledge graph (KG) triplets into token sequences, providing a unified learning algorithm [14, 15]. The implementation of this general strategy can vary significantly, depending on the injection heuristics. *Embedding-combined KEPTs* use representation learning algorithms to translate symbolic knowledge into an embedding space, improving resolution capabilities by combining resultant vectors with additional knowledge vectors [16]. In this paper, we will both consider a simple method based on direct text injection and K-BERT [15], a more complex technique that controls the visibility of the injected triples to reduce noise.

3. Background

This paper investigates various strategies for injecting knowledge into transformer models and utilizes the task of classifying scientific papers as a use case to evaluate these approaches. Specifically, we designed different versions of a BERT-based text classifier using different strategies for knowledge injection. This process involves training the proposed models to

perform a single-label multi-class classification task. The objective is to correctly identify and assign the research field of a paper based on its title and abstract.

To evaluate the classifiers, we created a benchmark of 12K papers split equally between the fields of Artificial Intelligence (AI), Software Engineering (SE), and Human-Computer Interaction (HCI). As a source of additional knowledge to be integrated, we utilized a KG of research topics, which was extracted from CSO. The upcoming sections discuss BERT, AIDA KG, and CSO.

BERT [1] is a transformer-based pre-trained model. As a "masked language model", BERT is pre-trained by predicting missing words in a sentence, using both the left and right context. The bidirectional training allows it to capture comprehensive language representations. BERT can be fine-tuned for a wide range of NLP tasks. In the case of text classification, BERT is extended by adding a classification layer on top of the pre-trained model. Once the text is tokenized, special tokens [CLS] and [SEP] are added, segment and position embeddings are assigned, and the task-specific classification layer is set, the model can be fine-tuned using labelled data.

The Computer Science Ontology (CSO - https://w3id.org/cso) is a large-scale ontology which provides a comprehensive taxonomy of Computer Science topics, including over 14K topics. It was generated by applying the Klink-2 algorithm over a corpus of 16M scientific articles. CSO incorporates three primary semantic relationships: *superTopicOf*, *relatedEquivalent*, and *preferentialEquivalent*. It is used by academic institutions and commercial organizations for supporting a range of relevant tasks, including scholarly data exploration, community detection, document retrieval, and article recommendation.

The Academia/Industry DynAmics KG (AIDA KG - https://w3id.org/aida) [9] is a KG that describes 21M articles and 8M patents, categorized according to the research topics from CSO [10, 11] and relevant industrial sectors (e.g., automotive, financial, energy). It was generated by integrating data from various data sources in this space, such as Microsoft Academic Search, Dimensions, OpenAlex, DBLP, the Research Organization Registry (ROR), and DBpedia.

4. The Proposed Benchmark

In this study, we propose a new benchmark for comparing knowledge injection strategies in scientific paper classification that comprises three components: i) 12K labelled paper abstracts, ii) a KG of related research topics, and iii) supplementary data to support various KIMs. We used a venue-based labelling approach to categorize papers into Artificial Intelligence (AI), Software Engineering (SE), and Human-Computer Interaction (HCI) fields, selecting papers published after 2010 and with at least three citations. The data set is balanced with 4K papers per category. The KG for supporting the KIM was generated by extracting the portion of CSO describing the fields of AI, SE, and HCI. This includes 4,629 topics and 9,258 triples.

The supplementary data enables the KIMs to select the KG portions pertinent to a specific article. They include a paper-topic map and a specificity score for each topic. The paper-topic map links papers to relevant topics within the KG. The specificity score reflects how discriminative is a topic for the classification task and is equal to the highest frequency among the frequencies with which a topic is distributed among the three categories of texts (AI, SE, and HCI) of the training dataset.

5. Knowledge Injection Methodologies (KIMs)

This section outlines the four KIMs that have been compared.

Direct Text Injection. Relevant information can be directly integrated into the text, a process akin to prompt extension [14]. This solution appends relevant triples at the end of the abstracts. Specifically, for each entity linked to the paper, we append two relevant triples. The triples are modified by converting semantic relationships (e.g., *subTopicOf*) into English expressions ("is a narrower concept than"). The resulting sentences are added to the text.

K-BERT. K-BERT [15] is a popular technique for knowledge injection that augments the text with triples from the KG. It incorporates a Knowledge Layer that identifies the KG entities in the input text[2] and appends to them relevant triples, creating a "sentence tree". The sentence tree is processed by the Embedding Layer, assigning positional embeddings, and the Seeing Layer, filtering noise via the *visible matrix*. This matrix ensures that injected predicates and objects only influence the embeddings of the entities they were attached to. The output from the previous layers is then processed by the Mask-Transformer, which adapts the self-attention mechanism to accommodate the visible matrix. K-BERT has shown improved performance compared to BERT in specific domains like finance, medicine, and law [15].

To use K-BERT for our case, we adapted its implementation[3] to process English texts and incorporate the KG detailed in Section 4. We adjusted the Knowledge Layer to recognize CSO topics' surface form in each sentence and append relevant ontology triples.

Integration of Additional Features Using a Multilayer Perceptron. We adapted the embedding-combined KEPTs method from [16], which extends BERT with non-textual data using a multilayer perceptron (MLP). The method uses the BERT model to process the text, concatenates the resulting embeddings with additional features derived from relevant metadata, and feeds them into the MLP. We modified the original implementation in [16] to fit our purpose. We used a standard English BERT model and introduced a component for generating a vector of features from the KG of research topics. For each article, we selected three topics with the highest specificity and concatenated them. We then used Sentence BERT (SBERT) [17] to transform the resulting string into an embedding. Finally, we concatenated this vector with original text's embedding and fed it to the MLP, adjusting the final SoftMax layer to give an output probability for each category.

Domain-specific Pre-training. Additional pre-training of BERT on a specific domain can be seen as a knowledge injection [1]. It involves masking selected input tokens and tasking BERT with predicting them using the surrounding context. The best results are often achieved by masking about 15% of input tokens. We started with the standard bert-base-uncased model[4] and extended its pre-training using text representations of CSO ontology triples as input.

[2] K-BERT uses string match to identify entity labels in the text.
[3] Available at https://github.com/autoliuweijie/K-BERT
[4] https://huggingface.co/bert-base-uncased

6. Evaluation and Conclusions

In this section, we present the performance of both the vanilla BERT and the BERT models enhanced with the four KIMs, evaluated on the benchmark described in Section 4. In summary, we compare the following methods: i) **BERT**, the uncased BERT model trained on text features, used as baseline; ii) Direct Text Injection (**BERT-DTI**), which appends additional knowledge at the end of the input text; iii) **K-BERT**, which appends additional triples to entities in the text; iv) Integration of Additional Features Using a MLP (**BERT-MLP**), which combines the BERT outputs with additional features; v) Domain-specific Pre-training (**BERT-PT**), which extends BERT pre-training on all triples in the KG. In all experiments, we utilized a balanced test set consisting of 1,500 documents. The size of the training datasets was varied, with trials conducted using 3,000, 6,000, and 9,000 articles, to assess the impact of differing training sizes. In line with the findings reported in [18], we run each configuration with 10 different random seeds. The standard deviations of the F1-scores were typically below 1%, as detailed in Table 1, demonstrating consistency. **BERT-MLP** outperformed the other methods for the largest training size (9K), yielding an F1-score of 0.880. **K-BERT** exhibit the best performance for 3K and 6K training sizes, yielding respectively 0.869 and 0.871. **BERT-DTI** exhibited marginal improvements over the vanilla BERT, indicating that even a basic solution that appends knowledge to the text can yield some benefits. Nevertheless, more sophisticated methods seem to produce much better results. Finally, **BERT-PT** performed worse than the standard BERT, likely attributable to the relatively small knowledge base utilized for pretraining. This outcome may suggest that in similar cases, KIMs that enhance input could be more effective than pretraining the model on domain-specific data.

In summary, **BERT-MLP** and **K-BERT** seem the best options for this task, with **BERT-MLP** showing an advantage when larger training data is available. Future research will broaden the analysis of KIMs in complex domains. This will involve exploring other fields and use cases further to understand the potential and limitations of these methods. As future directions, we will extend this work with more complete evaluation metrics and statistical significance tests; we will also share a public repository with a full code and examples.

Train size	BERT		BERT-DTI		K-BERT		BERT-MLP		BERT-PT	
	avg	std	avg	std	avg	std	avg	std	avg	std
3000	0.855	0.008	0.858	0.009	**0.869**	0.009	0.850	0.010	0.850	0.002
6000	0.860	0.010	0.863	0.003	**0.871**	0.007	0.866	0.009	0.852	0.003
9000	0.868	0.005	0.868	0.005	0.870	0.006	**0.880**	0.019	0.865	0.005

Table 1
F1-scores obtained for the five models at different sizes of the training set.

References

[1] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, Naacl-Hlt 2019 (2018) 4171–4186. doi:10.18653/v1/N19-1423.

[2] S. Gao, M. Alawad, M. T. Young, J. Gounley, N. Schaefferkoetter, H. J. Yoon, X.-C. Wu,

E. B. Durbin, J. Doherty, A. Stroup, et al., Limitations of transformers on clinical text classification, IEEE journal of biomedical and health informatics 25 (2021) 3596–3607.

[3] S.-W. Kim, J.-M. Gil, Research paper classification systems based on tf-idf and lda schemes, Human-centric Computing and Information Sciences 9 (2019) 1–21.

[4] A. A. Salatino, F. Osborne, A. Birukou, E. Motta, Improving editorial workflow and metadata quality at springer nature, in: The Semantic Web–ISWC 2019: Auckland, New Zealand, October 26–30, Springer, 2019, pp. 507–525.

[5] F. Barbieri, J. Camacho-Collados, L. Espinosa Anke, L. Neves, TweetEval: Unified benchmark and comparative evaluation for tweet classification, in: Findings of the Association for Computational Linguistics: EMNLP, 2020, pp. 1644–1650.

[6] K. S. Kalyan, A. Rajasekharan, S. Sangeetha, AMMUS : A survey of transformer-based pretrained models in natural language processing, CoRR abs/2108.05542 (2021). URL: https://arxiv.org/abs/2108.05542. arXiv:2108.05542.

[7] J. Yang, G. Xiao, Y. Shen, W. Jiang, X. Hu, Y. Zhang, J. Peng, A survey of knowledge enhanced pre-trained models, CoRR abs/2110.00269 (2021). URL: https://arxiv.org/abs/2110.00269. arXiv:2110.00269.

[8] C. Peng, F. Xia, M. Naseriparsa, F. Osborne, Knowledge graphs: opportunities and challenges, Artificial Intelligence Review (2023) 1–32.

[9] S. Angioni, A. Salatino, F. Osborne, D. R. Recupero, E. Motta, Aida: A knowledge graph about research dynamics in academia and industry, QSS 2 (2021) 1356–1398.

[10] A. A. Salatino, F. Osborne, E. Motta, Augur: Forecasting the emergence of new research topics, in: Proc. of the 18th ACM/IEEE on JCDL, JCDL '18, ACM, NY, USA, 2018, p. 303–312. URL: https://doi.org/10.1145/3197026.3197052. doi:10.1145/3197026.3197052.

[11] A. A. Salatino, T. Thanapalasingam, A. Mannocci, F. Osborne, E. Motta, The computer science ontology: A large-scale taxonomy of research areas, in: The Semantic Web – ISWC 2018, Springer, 2018, pp. 187–205. doi:10.1007/978-3-030-00668-6_12.

[12] Y. Xu, M. Namazifar, D. Hazarika, A. Padmakumar, Y. Liu, D. Hakkani-Tür, Kilm: Knowledge injection into encoder-decoder language models, 2023. arXiv:2302.09170.

[13] D. Emelin, D. Bonadiman, S. Alqahtani, Y. Zhang, S. Mansour, Injecting domain knowledge in language models for task-oriented dialogue systems, 2022. arXiv:2212.08120.

[14] P. Liu, G. Neubig, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, G. Neubig, Pre-train , Prompt , and Predict : A Systematic Survey of Prompting Methods in Natural Language Processing, ACM Comput. Surv. 55 (2023) 1–46. doi:10.1145/3560815. arXiv:2107.13586v1.

[15] W. Liu, P. Zhou, Z. Zhao, Z. Wang, Q. Ju, H. Deng, P. Wang, K-BERT: enabling language representation with knowledge graph, CoRR abs/1909.07606 (2019). URL: http://arxiv.org/abs/1909.07606. arXiv:1909.07606.

[16] M. Ostendorff, P. Bourgonje, M. Berger, J. M. Schneider, G. Rehm, B. Gipp, Enriching BERT with knowledge graph embeddings for document classification, CoRR abs/1909.08402 (2019). URL: http://arxiv.org/abs/1909.08402. arXiv:1909.08402.

[17] N. Reimers, I. Gurevych, Sentence-bert: Sentence embeddings using siamese bert-networks, 2019, pp. 3973–3983. doi:10.18653/v1/D19-1410.

[18] J. Dodge, G. Ilharco, R. Schwartz, A. Farhadi, H. Hajishirzi, N. A. Smith, Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping, CoRR abs/2002.06305 (2020). URL: https://arxiv.org/abs/2002.06305. arXiv:2002.06305.