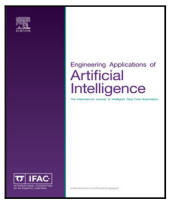
Engineering Applications of Artificial Intelligence 133 (2024) 108166

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www elsevier com/locate/engappai



Research paper

A comparative analysis of knowledge injection strategies for large language models in the scholarly domain

Andrea Cadeddu [a], Alessandro Chessa [a], Vincenzo De Leo [a,b], Gianni Fenu [b], Enrico Motta [c], Francesco Osborne [c,d], Diego Reforgiato Recupero [b,*], Angelo Salatino [c], Luca Secchi [a,b]

[a] Linkalab s.r.l., Cagliari, Italy
[b] Department of Mathematics and Computer Science, University of Cagliari, Cagliari, Italy
[c] Knowledge Media Institute, The Open University, Milton Keynes, United Kingdom
[d] Department of Business and Law, University of Milano Bicocca, Milan, Italy

ARTICLE INFO

Keywords:
Knowledge injection
Knowledge graphs
Large language models
Transformers
BERT
Classification
Natural language processing

ABSTRACT

In recent years, transformer-based models have emerged as powerful tools for natural language processing tasks, demonstrating remarkable performance in several domains. However, they still present significant limitations. These shortcomings become more noticeable when dealing with highly specific and complex concepts, particularly within the scientific domain. For example, transformer models have particular difficulties when processing scientific articles due to the domain-specific terminologies and sophisticated ideas often encountered in scientific literature. To overcome these challenges and further enhance the effectiveness of transformers in specific fields, researchers have turned their attention to the concept of knowledge injection. Knowledge injection is the process of incorporating outside knowledge into transformer models to improve their performance on certain tasks. In this paper, we present a comprehensive study of knowledge injection strategies for transformers within the scientific domain. Specifically, we provide a detailed overview and comparative assessment of four primary methodologies, evaluating their efficacy in the task of classifying scientific articles. For this purpose, we constructed a new benchmark including both 24K labelled papers and a knowledge graph of 9.2K triples describing pertinent research topics. We also developed a full codebase to easily re-implement all knowledge injection strategies in different domains. A formal evaluation indicates that the majority of the proposed knowledge injection methodologies significantly outperform the baseline established by Bidirectional Encoder Representations from Transformers.

1. Introduction

In recent years, transformer-based models have emerged as powerful tools for natural language processing tasks, demonstrating remarkable performance in several domains. For instance, BERT (Bidirectional Encoder Representations from Transformers) introduced a revolutionary approach that leveraged bidirectional context, significantly improving the state of the art in several tasks, such as text classification, named entity recognition, sentiment analysis, and question answering (Devlin et al., 2019). More recently, GPT-4 (OpenAI, 2023) has demonstrated remarkable proficiency in generating coherent text and facilitating more sophisticated interactions between humans and machines (Kung et al., 2023).

However, transformers still suffer from some limitations. These shortcomings become particularly apparent when dealing with highly specific and complex concepts, particularly within the scientific domain (Gao et al., 2021; Kumar, 2023). One crucial task in this space is to efficiently and accurately classify scientific articles (Kim and Gil, 2019). A good quality classification plays a crucial role in organising and retrieving knowledge, aiding researchers in staying up-to-date with the latest advancements and facilitating the dissemination of information within the scientific community (Salatino et al., 2019a). However, scientific article classification poses unique challenges for transformer models due to the intricate language and nuanced domain-specific concepts prevalent in scientific literature. Consequently, transformers can struggle to differentiate between concepts that are quite dissimilar for domain experts, and in some cases, they may even generate completely fictional information, a phenomenon known as hallucination (Alkaissi

* Corresponding author.
E-mail addresses: andrea.cadeddu@linkalab.it (A. Cadeddu), alessandro.chessa@linkalab.it (A. Chessa), vincenzo.deleo@linkalab.it (V. De 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. Reforgiato Recupero), angelo.salatino@open.ac.uk (A. Salatino), luca.secchi@linkalab.it (L. Secchi).

https://doi.org/10.1016/j.engappai.2024.108166
Received 18 September 2023; Received in revised form 13 November 2023; Accepted 24 February 2024
Available online 29 February 2024

and McFarlane, 2023). Further pre-training of existing transformers on specialised documents is a common technique to extend a model with domain-specific knowledge (Caselli et al., 2020; Barbieri et al., 2020; Lee et al., 2019; Leivaditi et al., 2020). However, this approach is quite demanding since it requires processing a large volume of domain-specific unlabelled text to adapt the model parameters in an effective way (Kalyan et al., 2021). It also has limitations in domains that utilise very specific terminologies (Kalyan et al., 2021).

To overcome these challenges and improve the effectiveness of transformers in specific fields, researchers have turned their attention to the concept of knowledge injection (Yang et al., 2021). Knowledge injection involves integrating external knowledge sources into the transformer models to augment their understanding and consequently their performance in relevant tasks. Knowledge injection methodologies can handle many types of structured information. Notably, knowledge graphs (KGs) have gained prominence as powerful tools for representing and organising structured data in a semantically meaningful manner (Peng et al., 2023). KGs adeptly capture the intricate relationships that exist between entities and attributes, offering a machine-readable representation of the domain for the benefit of various intelligent services (Dessí et al., 2022b; Chessa et al., 2023). They typically structure information based on a domain ontology, which serves as a formal description of entity types and their relationships while supporting reasoning processes (Hitzler, 2021).

This paper presents a comprehensive study of knowledge injection approaches for transformers within the scientific domain. Specifically, we provide a detailed overview and comparative assessment of four primary methodologies, evaluating their efficacy in the task of classifying scientific articles. We also develop and share a full codebase to easily re-implement all knowledge injection strategies in different domains.

In order to perform this study, we constructed *AIDA24k*, a new public benchmark for scientific article classification based on 24k scientific articles extracted from the Academia/Industry DynAmics Knowledge Graph (AIDA KG)[1] (Angioni et al., 2021). As external knowledge for the knowledge injection methodologies, we adopted the Computer Science Ontology (CSO) (Salatino et al., 2018a). CSO is a large-scale ontology of research areas in the field of Computer Science. Compared to other solutions in this space (e.g., the ACM Computing Classification System), it offers a much more granular representation of research concepts. For this reason, CSO was adopted by Springer Nature to automatically annotate proceeding books in Computer Science (Salatino et al., 2019a) and it is routinely used by a large number of tools to explore and analyse the scholarly domain (Löffler et al., 2020; Zhang et al., 2021; Vergoulis et al., 2020; Beck et al., 2020; Chatzopoulos et al., 2020).

We report several insightful findings about the impact of knowledge injection strategies on scientific text classification. Interestingly, even a straightforward method based on directly appending domain knowledge to the text showed a notable improvement in performance. K-BERT (Liu et al., 2019b), a more sophisticated version of this strategy, which controls the visibility of the injected knowledge to affect only relevant tokens, achieved significantly better results on the smaller training sets ($p < 0.0001$). The most effective strategy employed a hybrid architecture, integrating BERT with a multilayer perceptron (MLP) to merge textual data with external knowledge (Ostendorff et al., 2019). This approach significantly outperformed all the other strategies for the larger training sets ($p < 0.0001$). For instance, it achieved a 3.3 F1-score enhancement over the BERT baseline when considering a training set of 21K papers.

In summary, the main contributions of this paper are the following:

- We present a comparative analysis of different strategies for injecting knowledge into transformers, specifically for the task of scientific article classification.
- We provide AIDA24k, a new benchmark composed of 24K research articles, evenly categorised into three research fields: Artificial Intelligence (AI), Software Engineering (SE), and Human-Computer Interaction (HCI). The benchmark also includes a knowledge graph of pertinent research topics and other additional information to support the knowledge injection methodologies.
- We release the complete codebase for implementing the four knowledge injection strategies under analysis.[2] This provision aims to facilitate researchers in assessing the efficacy of these strategies across diverse domains and enables developers to seamlessly integrate them into their projects.

The remainder of this paper is organised as follows. Section 2 provides a review of knowledge injection strategies. Section 3 defines the classification task under study and provides an overview of the BERT transformer, the AIDA Knowledge Graph, and the Computer Science Ontology. Section 4 describes in detail the new benchmark. Section 5 illustrates four general methodologies for knowledge injection. In Section 6, we report and discuss the results of the comparative evaluation. Section 7 highlights the limitations of the study and outlines potential directions for future research. It also includes some preliminary results for other tasks and domains. Finally, Section 8 discusses the implications of our findings and ends the paper.

2. Related work

Since the introduction of transformers, the scientific community has been aware of their limitations and thus started working on Knowledge-Enhanced Pre-trained Transformers (KEPTs). For instance, Xu et al. (2023) developed a novel approach that injects entity-related knowledge into encoder–decoder large pre-trained language models. This is achieved via a generative knowledge infilling objective through continued pre-training. Emelin et al. (2022) proposed to inject domain-specific knowledge prior to fine-tuning task-oriented dialogue tasks via lightweight adapters that can be integrated with language models and serve as a repository for facts learned from different knowledge bases. Moiseev et al. (2022a) described a method to infuse structured knowledge into LLMs, by directly training T5 models on factual triples of knowledge graphs. Wang et al. (2021b) proposed a method that keeps the original parameters of the pre-trained model fixed and supports continuous knowledge infusion via a neural adapter for each type of infused knowledge, as a plug-in connected to the language model.

Given the recent explosion of works about KEPTs, authors in Yang et al. (2021) have proposed a classification according to three properties: (i) the granularity of knowledge, (ii) the method of knowledge injection, and (iii) the degree of symbolic knowledge parameterisation. Regarding the first property, KEPTs integrate knowledge at different levels of granularity depending on the underlying task. For example, tasks of sentimental analysis mainly rely on word features and thus require more information about individual entities whereas tasks of text generation rely on commonsense knowledge. Regarding the second property, the method of knowledge injection plays an essential role in the effectiveness and efficiency of the integration between Pre-Trained Models (PTM) and infused knowledge, as well as with respect to knowledge management and storage. Indeed, the method used to inject knowledge determines what knowledge can be integrated and the form of that knowledge. Regarding the third property KEPTs are based on the concept that knowledge can be harnessed by PTMs in the form of symbols or semantic embeddings. To bridge the symbolic knowledge and neural networks, the former is projected into a dense, low-dimensional semantic space and presented by distributed vectors through knowledge representation learning.

[1] Academia/Industry DynAmics Knowledge Graph — http://w3id.org/aida/.

[2] https://github.com/vincenzodeleo/kims-bert

Yang et al. (2021) discusses six different strategies for injecting knowledge within pre-trained models. *Feature-fused KEPTs* (e.g., SentiLARE (Ke et al., 2020) and Ernie (Zhang et al., 2019)), *Embedding-combined KEPTs* (e.g., Luke (Yamada et al., 2020), Cokebert (Su et al., 2021)), *Data-structure-unified KEPTs* (e.g., K-BERT (Liu et al., 2019b), K-LM (Kumar et al., 2022), Colake (Sun et al., 2020), Comet (Bosselut et al., 2019)), *Knowledge-supervised KEPTs* (e.g., Kepler (Wang et al., 2021a) and ERICA (Qin et al., 2021)), *Retrieval-based KEPTs* (e.g., Realm (Guu et al., 2020; Joshi et al., 2021)), *Rule-guided KEPTs* (e.g., Gangopadhyay et al. (2021), Amizadeh et al. (2020)). However, many of the approaches mentioned are unsuitable for this study's use case. This is due to their requirements for pre-training the transformer on sizeable textual data, establishing domain-specific rules, or querying external sources of knowledge.

Here we focus our attention on methods that do not require extensive pre-training and can be easily generalised for multiple domains. We set this requirement to explore knowledge injection in a low-budget/low-computation resource setting. In particular, we consider three general methods of knowledge injection.

The first strategy can be mapped to the data-structure-unified KEPTs as discussed in Yang et al. (2021). It aims to convert the relational triplets from the knowledge graph into token sequences that are incorporated in the input text (Sun et al., 2020; Bosselut et al., 2019). The main advantage of this approach is that the same encoder can be used to learn embeddings for both the text and the injected knowledge. Thus, this solution bypasses the structural incompatibility hurdle between the pre-trained models and knowledge-injected data. However, it is worth mentioning that the creation of this unified data structure depends on which heuristic to choose for knowledge injection. A basic implementation of this approach may simply add specific triples to the text. In contrast, a more advanced method like K-BERT (Liu et al., 2019b) strategically places triples within the text and regulates their visibility, ensuring only relevant tokens are influenced. A possible drawback of this approach is that the transformation process discards the structural information inherent in the knowledge graph. This may limit the amount of context and relational data available for understanding complex relationships in the data.

The second method involves conducting a lightweight pre-training (Liu et al., 2019a) on a version of the knowledge base that has been converted to text (Sun et al., 2020; Moiseev et al., 2022b). The effectiveness of this method largely depends on the nature and size of the knowledge base.

The third strategy involves incorporating the knowledge as additional feature data during the classifier's training. These features may be quantitative, such as the number of authors of a document (Ostendorff et al., 2019), or encoded as the embeddings of specific entities (Yamada et al., 2020; Su et al., 2021). In this framework, the embedding vectors produced by the pre-trained model are combined with integrative features representing the additional symbolic knowledge to enhance its ability to resolve specific tasks.

3. Background

In this study, we evaluate various knowledge injection strategies for transformers, assessing their impact on enhancing the performance of text classifiers. More specifically, we focus on enhancing a BERT-based model for the classification of research papers. In fact, BERT (Devlin et al., 2019) is one of the most widely used Transformer-equipped pre-trained language models to determine the contextualised representation of input text. The classification of research papers is crucial for the effective dissemination of scientific literature and is commonly executed by digital libraries, publishers, and analytical platforms (Kim and Gil, 2019; Salatino et al., 2022).

Specifically, this task involves training a classifier to solve a single-label multi-class classification problem where, given the research papers' title and abstract, the classifier assigns each paper its main relevant research field. More formally, given an array x of n input samples, and a number of labels l, the model's objective f is to assign each $x[i]$ to one of the labels l. That is, computing $f\ x[i]) = c$, where $c = 0\ \ldots\ l-1$ and for $i = 0\ \ldots\ n-1$.

To train and evaluate relevant classifiers, we constructed a benchmark of 24K research papers, equally split into three research areas: Artificial Intelligence (AI), Software Engineering (SE), and Human-Computer Interaction (HCI). This benchmark also includes a knowledge graph of relevant research topics extracted from CSO. The underlying premise of this resource revolves around utilising the knowledge graph as a means to integrate additional knowledge into the classification process, with the ultimate objective of enhancing the overall performance.

In the following, we will describe BERT and the background data used for building the benchmark: the AIDA Knowledge Graph and CSO.

3.1. BERT

BERT (Devlin et al., 2019) is a pre-trained language model built on the Transformer architecture, achieving top-tier results across multiple natural language processing (NLP) tasks. As a "masked language model", BERT's pre-training process involves predicting masked words in sentences. Unlike many other models that only used left-to-right or right-to-left context, BERT uses bidirectional training, enabling it to capture more comprehensive language representations. BERT can be fine-tuned for a wide range of NLP tasks, including Sentence Classification, Named Entity Recognition, Part-of-Speech Tagging, Question Answering, Text Summarisation, Text Classification, and Semantic Textual Similarity.

To fine-tune BERT for a classification task, a classification layer is added to the pre-trained model. This layer processes the final hidden state from the transformer network, mapping it to the required number of classes with a fully connected layer followed by a Softmax layer. During the fine-tuning process, the text is first tokenised, and special tokens [CLS] and [SEP] are incorporated at the appropriate positions. Each token is then assigned segment and position embeddings. Finally, the entire model, including the pre-trained BERT weights, is fine-tuned on the specific classification task utilising the labelled data.

3.2. The AIDA knowledge graph

The AIDA Knowledge Graph (AIDA KG) (Angioni et al., 2021) is a large-scale knowledge base that describes 21M publications and 8M patents. These entries are characterised using an extensive range of metadata, including authors, organisations, countries, and venues. Furthermore, they are linked with research topics from the Computer Science Ontology (CSO) (Salatino et al., 2018b) and the relevant industrial sectors from the Industrial Sectors Ontology (INDUSO).[3]

AIDA KG was generated by an automatic pipeline that integrates data from several sources, such as OpenAlex, DBLP, the Research Organisation Registry (ROR), DBpedia and Wikidata. It is publicly available under CC BY 4.0 and can be downloaded as a dump or queried via a triplestore https://aida.kmi.open.ac.uk/sparql/.

This knowledge graph is structured according to the Resource Description Framework (RDF), which is a World Wide Web Consortium (W3C) standard.[4] Specifically, the information is stored through triples (or statements), such as <*subject*, *predicate*, *object*>, in which the *subject* represents an entity, the *object* can also represent an entity or simply a general text (e.g., author name), and the *predicate* describes the relationship between the subject and object. For instance, the triple <*Prof Yoshua Bengio*, *affiliation*, *University of Montreal*> states that the entity associated to *Prof Yoshua Bengio* is affiliated with the *University of Montreal* organisation entity.

[3] INDUSO — https://aida.kmi.open.ac.uk/downloads/induso.ttl.

[4] RDF Standard — https://www.w3.org/TR/rdf-concepts/.

AIDA KG describes eight types of entities: papers, patents, authors, affiliations, journals, conferences, topics, and industrial sectors. These entities are connected by 22 relations, such as: (i) `hasAffiliation`, which specifies the affiliations of the authors, (ii) `hasGridType`, which identifies the type of the affiliation according to the GRID classification (e.g., Education, Company, Government, Non-profit, and other), (iii) `schema:creator`, which indicates the author of a paper. The full list of relationships in AIDA KG is available at https://w3id.org/aida.

3.3. The computer science ontology

The Computer Science Ontology (CSO) is a large-scale, automatically generated ontology of research areas. It includes about 14K research topics and 159K statements, making it a comprehensive taxonomy of research areas in Computer Science. It was produced by applying the Klink-2 algorithm (Osborne and Motta, 2015) on a dataset of 16M scientific articles.

CSO structures research topics poly-hierarchically. While 'Computer Science' is the primary root of this hierarchy, it also encompasses several other foundational categories, including Linguistics, Geometry, Semantics, among others.

CSO is also structured according to RDF, like AIDA KG, and it includes three main semantic relationships:

1. `superTopicOf`: indicating that a topic is a super-area of another topic. It represents a hierarchical relationship where a broader concept encompasses a narrower concept. For instance, "software engineering" is a supertopic of "software design".
2. `relatedEquivalent`: indicating that two topics can be treated as equivalent for the purpose of exploring research data. For instance, "haptic device" is equivalent to "haptic interface".
3. `preferentialEquivalent`: indicating the main label for topics belonging to a group of relatedEquivalent. For instance, "ontology" and "ontologies" will both have "ontology" as `preferentialEquivalent`.

Other relationships available in CSO include `contributesTo`, indicating that the research output of one topic contributes to another topic, and `sameAs` mapping research topics to similar entities in other knowledge graphs such as DBpedia,[5] Wikidata,[6] YAGO,[7] and Cyc.[8]

CSO is openly available and can be downloaded in various RDF formats (NT, TTL, XML) from the CSO Portal.[9]

Unlike other existing approaches (such as the ACM Computing Classification System), the CSO covers a much larger number of research topics, which can enable a fine-grained description of the content of research papers. Indeed, CSO has been proven to be eclectic as it effectively supports a wide range of tasks, such as exploring and analysing scholarly data (e.g., ConceptScope (Zhang et al., 2021), ScholarLensViz (Löffler et al., 2020), Rexplore (Osborne et al., 2013)), inspecting scholarly data with conversational agents (e.g., AIDA-Bot (Meloni et al., 2023)), detecting research communities (e.g., ACE (Rizvi et al., 2023),), improved retrieval of research documents (e.g., CDSS (Mardiah et al., 2023)), identifying domain experts (e.g., VeTo (Vergoulis et al., 2020)), refining the selection of keywords (e.g., R-Classify (Aggarwal et al., 2022), ASM (Chamorro-Padial and Rodríguez-Sánchez, 2023)), recommending articles (Thanapalasingam et al., 2018) and video lessons (Borges and dos Reis, 2019), expanding existing ontology models (Han, 2023; Chari et al., 2023), generating knowledge graphs (e.g., Temporal KG (Rossanez et al., 2020), AIDA KG (Angioni et al., 2021), CS KG (Dessí et al., 2022b), KGs for education (Li et al., 2023)), knowledge graph embeddings (e.g., Trans4E (Nayyeri et al., 2021)), topic models (e.g., CoCoNoW (Beck et al., 2020)), and analysing the impact of research teams (e.g., (Salatino et al., 2023)).

Finally, Springer Nature, one of the leading global academic publishers, has also integrated CSO into several of its innovative applications, such as (i) Smart Topic Miner (Salatino et al., 2019a), a tool that helps the Springer Nature editorial team to categorise proceedings, (ii) Smart Book Recommender (Thanapalasingam et al., 2018), an ontology-driven recommender system for choosing books to promote at academic events, and (iii) the AIDA Dashboard (Angioni et al., 2020), a web application for exploring and analysing scientific venues, countries, and research topics.

[5] https://www.dbpedia.org/
[6] https://www.wikidata.org
[7] https://yago-knowledge.org/
[8] https://cyc.com/
[9] Download CSO from https://w3id.org/cso.

Table 1
Distribution of venues according to the three disciplines.

Venues/Discipline	AI	SE	HCI	Total
Conferences	90	62	79	231
Journal	9	8	19	36
Total	99	70	98	267

4. The AIDA24k benchmark

In this section, we describe AIDA24k, the new benchmark we constructed to compare different strategies for knowledge injections in the context of scientific paper classification. This dataset includes three primary components: (i) a collection of 24K labelled papers (title and abstract), (ii) a knowledge graph describing pertinent research topics, and (iii) the supplementary material to support the knowledge injection methodologies.

4.1. The scientific articles

The process of attributing a specific research field to a given paper poses a challenging task, even for domain experts. In light of this, we opted to employ a labelling approach based on the publication venue. To facilitate this, we carefully curated a selection of 35 journal and 231 conference papers for three chosen disciplines: Artificial Intelligence (AI), Software Engineering (SE), and Human-Computer Interaction (HCI). As an example, for the field of AI, we included articles published at the Neural Information Processing Systems (NeurIPS) conference and the Nature Machine Intelligence journal. Whereas, for the field of Software Engineering we included papers coming from the ACM's Model-Driven Engineering Languages and Systems (MODELS) conference and the IEEE Transactions On Software Engineering journal. Finally, for HCI we included articles from the Human Factors in Computing Systems (CHI) conference and Elsevier's International Journal Of Human-Computer Studies. Table 1 shows the number of conferences and journals distributed for each discipline, whereas, within the online repository, we reported the full list of venues.

We extracted the set of articles from the AIDA Knowledge Graph that were published in these designated venues after the year 2010 and received at least 3 citations. In order to obtain a balanced dataset, we then selected a random subset of 8k papers for each category, for a total of 24k articles. Each article in AIDA24k is represented by an ID, alongside its corresponding title and abstract.

4.2. The knowledge graph

To support our knowledge injection process, we constructed a knowledge graph that includes 4629 topics and 9258 statements, relevant to the fields of AI, SE, and HCI, which are drawn from the CSO ontology. In particular, we first identified the unique set of CSO topics that the AIDA24k papers are annotated with. Then, for each

topic, we selected the top two statements, with the topic as their subject, by ranking all triples based on their predicates. The ranking followed the order of 'subTopicOf' (the inverse of 'superTopicOf', materialised for this purpose), 'superTopicOf', 'preferentialEquivalent', and 'relatedEquivalent'.

As an example, the following is the description of the concept *image retrieval*, as derived from CSO:

```
<image retrieval, subTopicOf, pattern recogni-
tion>
<image retrieval, superTopicOf, color and tex-
ture features>
```

4.3. Supplementary material

The supplementary material enables the knowledge injection methodologies to select the KG portions pertinent to a specific article. We included them with the purpose of ensuring a fair and consistent comparison of various methodologies. By doing so, we guarantee that all methodologies involved in the evaluation receive the same set of information, eliminating potential variations arising from diverse implementations. Specifically, we provide (i) a mapping between each paper and the CSO topics, and (ii) a specificity score for each topic.

4.3.1. Mapping between scientific articles and CSO

The mapping was included since knowledge injection strategies frequently rely on entity-linking techniques to select the most pertinent portions of knowledge that are relevant to the item under consideration (Liu et al., 2019b). Entity-linking techniques serve the purpose of connecting segments of text to the corresponding entities within a knowledge base (Al-Moslmi et al., 2020). Therefore, we applied the CSO Classifier (Salatino et al., 2019b) to all the abstracts to identify which terms corresponded to research topics in CSO. The CSO Classifier is an unsupervised entity linking approach that uses a combination of string and word embeddings similarity for identifying concepts described in CSO. For instance, the paper with ID 4,[10] is associated with six topics, such as *natural language processing online learning environment*, and *recurrent neural networks*. This information will be leveraged by the knowledge injection methodologies described in the following section to select the concepts and triples most relevant to a specific article.

4.3.2. Specificity scores of topics

The second part of supplementary material consists of the specificity score. This is essential in optimising the knowledge injection process, taking into account various constraints that limit the amount of information that can be incorporated when classifying a specific document. Such constraints are influenced by factors such as the 512-token input limitation inherent to the BERT model, as well as specific restrictions of the methodology itself. For example, the standard implementation of K-BERT incorporates only two triples for each entity recognised in the text.

In the AIDA24k dataset, papers are associated with an average of 14 topics, varying from a minimum of 1 topic to a maximum of 92 topics. Most of the injection methods discussed in this work cannot handle the full range of topics associated with a paper. Therefore, they require the adoption of prioritisation criteria to determine which topics are the most significant and, thus, should be considered for injection and which ones should be omitted.

In order to assist this process and align all the methodologies, we compute the specificity score ss_i of topic i, which indicates its discriminative power with respect to the classification task. For this, we employed Eq. (1), which takes the maximum number of times a given entity e_i is found within the abstract of the papers associated with different research fields e_i^F, with $F = \{AI\ SE\ HCI\}$, and divides it by the number of times the same entity has been found within the whole AIDA24k train benchmark $e_i^{AI} + e_i^{SE} + e_i^{HCI}$.

$$ss_i = \frac{\max\ \{e_i^{AI}\ e_i^{SE}\ e_i^{HCI}\}}{e_i^{AI} + e_i^{SE} + e_i^{HCI}} \tag{1}$$

For example, if $ss_i = 0.9$, the topic i is frequently associated with just one research field and is thus highly specific. On the other hand, if $ss_i = 0.33$, the topic i is associated with an equal probability with all the research fields and thus is unspecific.

5. Knowledge injection methodologies

This section describes the four general knowledge injection approaches that were chosen for the comparative analysis. For each approach, we first outline the general methodology and then discuss how it was adapted for the classification of the scientific text we tackled. We also briefly touch upon significant aspects of the implementation. All the code for implementing the following methodology is included in the GitHub repository previously mentioned.

5.1. Direct text injection

A first simple strategy for knowledge integration is to directly augment the input texts with additional knowledge, leveraging the principle of prompt extension (Liu et al., 2023). This enriched text is employed during both the fine-tuning phase and the later classification procedure.

The most straightforward implementation of this solution involves converting all pertinent information from the knowledge graph into a string, which is then appended to the end of the original text. Given the limited context size of current transformers, it is crucial to develop a strategy to select only the most relevant information for the task at hand.

5.1.1. Adaptation to scientific article classification

We developed from scratch a method that implements this strategy by injecting triples from CSO at the end of the text to classify. We assign to each entity detected in the text two pertinent triples from CSO. Next, we employ a series of heuristics to translate the RDF triple into English sentences. For instance, the 'subTopicOf' relation is converted to the phrase "is a narrower concept than". The resulting strings are then appended to the end of the text. To illustrate, the sentence "User comfort-oriented residential power scheduling in smart homes" would be extended with: "Smart homes is a narrower concept than ambient intelligence and a broader concept than smart manufacturing" as depicted in Fig. 1. Note that the entity "smart homes" is recognised as the surface form of "smart environment", which is a topic of the CSO.

5.2. K-BERT

K-BERT (Liu et al., 2019b) is a well-established approach for integrating knowledge into BERT. Similarly to the first strategy, the core idea is augmenting the textual data through the direct injection of triples selectively picked from the knowledge graph. However, K-BERT takes a more refined approach by appending triples only after specific entities and ensuring that the introduced knowledge only affects the relevant tokens (see Fig. 2).

In K-BERT, the input text is first passed to the Knowledge Layer, which identifies the surface form of the entities in the knowledge graph within the text.[11] A *surface form* is the exact literal representation of an

[10] https://aclanthology.org/2020.bea-1.13.pdf

[11] K-BERT employs a string match approach to identify entity labels in the text.

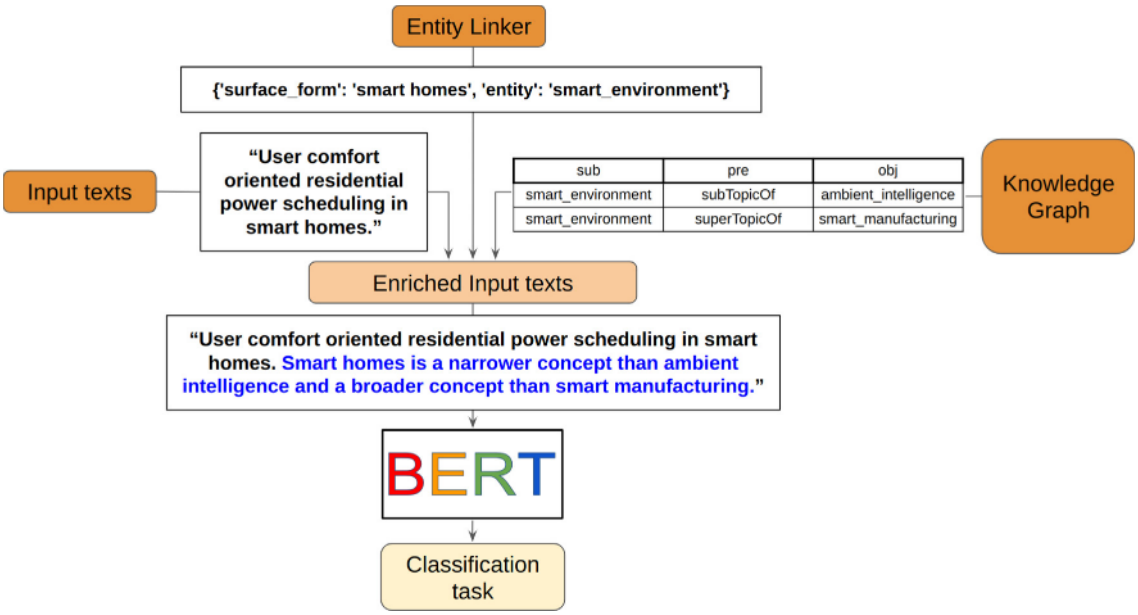


Fig. 1. Direct text injection.

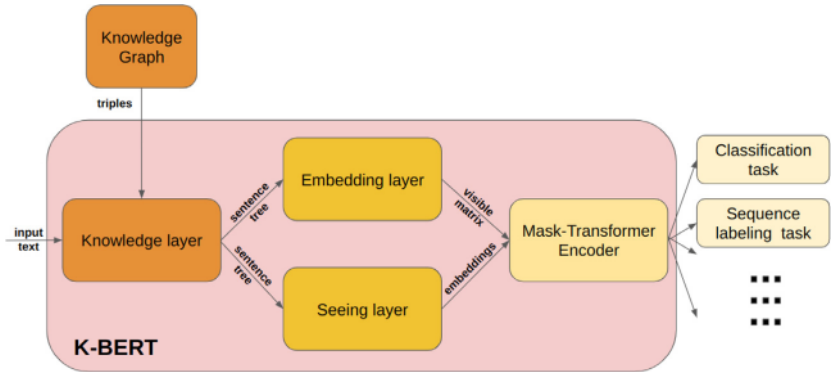


Fig. 2. K-BERT architecture.

entity as it appears in the text, and it may vary from the canonical label assigned to the entity within the knowledge graph. This distinction allows for flexibility in recognising entities as they naturally occur in the text, accommodating variations in spelling, abbreviation, or other forms of textual representation. For instance, the surface form of the entity *peer-to-peer* can be *p2p networks* or *peer-to-peer systems*. Next, K-BERT concatenates to the surface form the predicate and object of a selection of pertinent triples (typically two) from the knowledge graph. This results in the "sentence tree" data structure exemplified in Fig. 3.

The sentence tree is processed by two layers: the Embedding Layer and the Seeing Layer. The Embedding Layer assigns to each token in the sentence tree two types of positional embeddings according to their soft-position and hard-position indexes. The soft-position index of a token represents its distance from other tokens that should be considered pertinent during attention calculation, while the hard-position index represents the actual position of the token in the sequence from left to right. The Seeing Layer addresses the potential noise issue that may arise when appending triples within the text by introducing the *visible matrix*. The visible matrix governs the visibility of injected predicates and objects, ensuring they solely influence the embedding of relevant surface forms. As a result, this mechanism enables the acquisition of more comprehensive embeddings for knowledge graph entities without introducing excessive noise to the representation of other text components.

Finally, the output of the Embedding and the Seeing layers is fed to the Mask-Transformer. This component, composed of a stack of mask-self-attention blocks, implements a modified version of the self-attention mechanism in BERT that considers the visible matrix.

To better illustrate how K-BERT operates, we can consider the sentence *"Real-time recognition of dynamic hand gestures from video streams is a challenging task"*. In the first phase of the knowledge injection, K-BERT recognises two surface forms: *real-time recognition* and *hand gestures*. It then concatenates to them the predicate and object of two triples each. Therefore, the sentence becomes *"Real-time recognition* ***is n rrower concept th n ction recognition is bro der concept th n motion history im ges*** *of dynamic hand gestures* ***is n rrower concept th n gesture recognition is bro der concept th n h nd posture*** *from video streams is a challenging task"*. As discussed before, the tuples *<predicate, object>* injected in the sentence will be only visible when computing the embedding of *real-time recognition* and *hand gestures*. Therefore, they will not introduce noise when considering the other tokens.

Fig. 3 demonstrates the application of soft position indexes in this example. While hard position indexes measure the actual distance between two tokens, soft position indexes represent the distance when considering it as a sentence tree. For instance, in the augmented sentence, the distance between "real-time recognition" and "motion history image" is larger than the one between "real-time recognition" and "action recognition". However, when using soft indexes, both have an identical distance of six tokens from "real-time recognition".

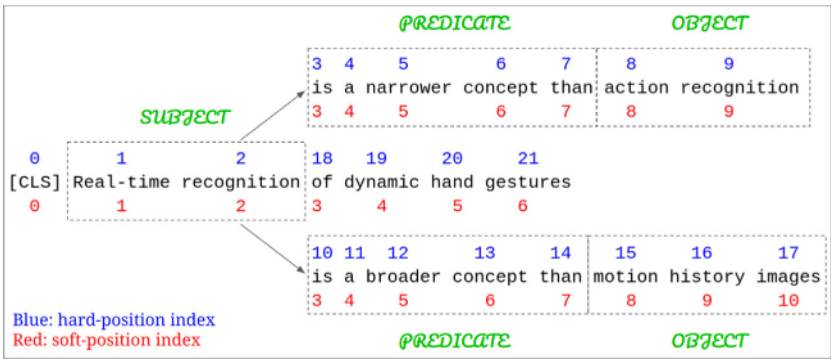


Fig. 3. Basic principle of indexing in sentence trees.

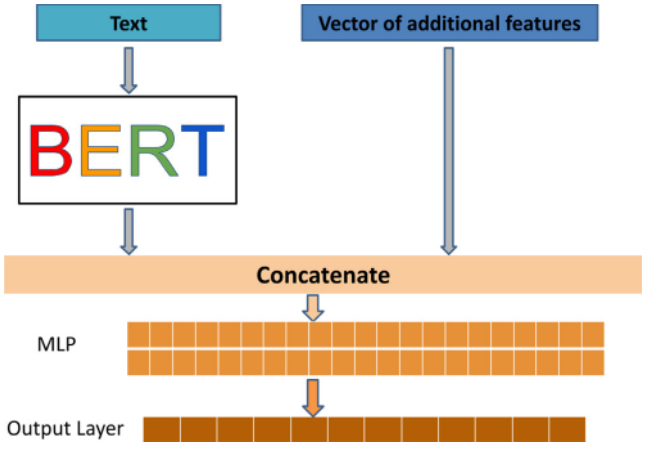


Fig. 4. Integration of additional features using a multilayer perceptron.

Previous work (Liu et al., 2019b) shows that K-BERT can significantly increase the performance of BERT on a range of tasks, especially when applied to specific domains like finance, medicine, and law.

5.2.1. Adaptation to scientific article classification

Applying the implementation of K-BERT[12] to our use case required two main adaptations of the original code. First, we extended K-BERT so that it could also process English texts. The original code, in fact, was developed to handle only Chinese. Second, we modified the Knowledge Layer to use the knowledge graph described in Section 4. Specifically, the Knowledge Layer identifies the surface form of the topics from CSO in each sentence and concatenates to them the relevant triples from the ontology. In accordance with the original K-BERT implementation, we assign two triples to each entity in the texts.

5.3. Integration of additional features using a multilayer perceptron

Rather than directly injecting the knowledge into the text, it is possible to incorporate it as additional feature data during the classification process. As a representative example of this concept, we consider the method for enhancing BERT with additional metadata presented in Ostendorff et al. (2019). The proposed neural network architecture extends BERT by combining text with additional features using a multilayer perceptron (MLP). Fig. 4 displays the general architecture.

To derive contextualised representations from textual features, BERT processes the text and returns the relevant embeddings. These embeddings are then concatenated with additional features derived by the representation of the item to classify in a knowledge base. This augmented representation is fed to the MLP. The MLP uses a SoftMax output layer that carries out a multi-class multi-label classification task, producing the probability for each classification label.

The additional features in the original implementation, which addressed the classification of books, included both numeric features (e.g., number of authors, length of the title) and the graph embedding of relevant entities (Lerer et al., 2019) (the authors of the books). However, this is a flexible strategy that can leverage many types of features.

5.3.1. Adaptation to scientific article classification

Our implementation is based on an adapted version of the code released by Ostendorff et al. (2019). To this purpose, we introduced a few notable modifications. First, we switched to a standard BERT model in English, while the original implementation adopted a BERT model that was exclusively pre-trained on German text. Second, we introduced a component that produces a vector of features from the knowledge graph described in Section 4. For each article in the dataset, we select the three topics with the highest specificity and concatenate them. For instance, the paper with ID 3[13] has the following topics represented as tuples where the number next to the topic is the specificity score of the topic: `(vocabulary, 0.425), (linkage analysis, 0.571), (wordnet, 0.862), (lexical database, 0.925), (word sense, 0.944), (synsets, 1.00)`.

We then concatenate the three selected topics and obtain the following string: “lexical database word sense synsets”.

The resulting string is processed with Sentence BERT (Reimers and Gurevych, 2019) to produce an embedding vector. Sentence BERT is a technique for creating fixed-length sentence embeddings that capture the semantic meaning of entire sentences. The embedding of the original text and the embeddings of the topics are then concatenated and fed to the MLP.

5.4. Additional pre-training on the KG

Another strategy involves further pre-training a model on a version of the knowledge graph that has been converted to text (Sun et al., 2020; Moiseev et al., 2022b). There are various methods to transform knowledge bases into a textual format. A basic approach is to utilise the complete set of triples, omitting the prefixes. For a more refined conversion, a series of heuristics may be employed to render the triples

[12] The code is freely accessible on GitHub https://github.com/autoliuweijie/K-BERT.

[13] https://aclanthology.org/W18-0514/

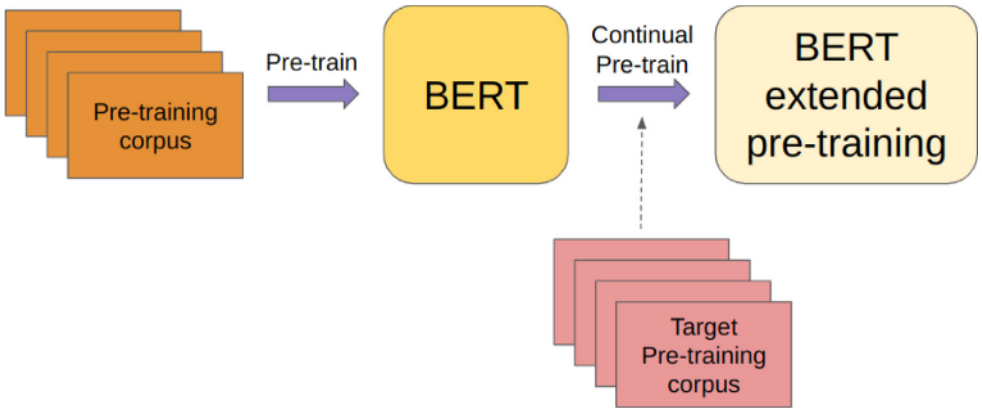


Fig. 5. Knowledge Injection with random masking.

into more fluent textual expressions. Some systems even extend this approach by integrating the knowledge graph with additional information obtained from a connected textual corpus, thereby incorporating details related to the practical utilisation of the entities (Sun et al., 2020). Nevertheless, this method demands substantial resources for processing extensive textual data.

The pre-training process involves substituting selected input tokens with a [MASK] token. The BERT model is then tasked with predicting these obscured tokens, utilising the context provided by the adjacent, unmasked tokens. Optimal performance is typically achieved when approximately 15 of input tokens are randomly replaced with the mask token. The resultant model, after this additional pre-training phase, should exhibit enhanced suitability for the specific domain it was trained for.

5.4.1. Application to scientific article classification

We selected as the base model the standard bert-base-uncased model[14] and applied the continual pre-train over the KG triples, as shown in Fig. 5 . We conducted a domain-specific extension of the pre-training of the model utilising a textual representation of all triples in the CSO ontology. Each triple, denoted as *s, p, o)*, was transformed into a string, following the pattern "s p o". This transformation involved the removal of prefixes and substituting the underscore character ("_") with a space. For instance, the triple

`s: <https://cso.kmi.open.ac.uk/topics/display_devices>,`
`p: <http://cso.kmi.open.ac.uk/schema/cso superTopicOf>,`
`o: <https://cso.kmi.open.ac.uk/topics/3-d_displays>`

was modified to form the triple `(display devices, is a broader concept than, 3-d displays)`. All the resulting strings were concatenated to form a single, continuous text (e.g., "display devices is a broader concept than 3-d displays").

6. Evaluation

In this section, we provide a comparative evaluation of the standard BERT against the BERT models that have been enhanced by using the four knowledge injection methodologies we discussed in Section 5. The models were assessed using the benchmark detailed in Section 4.

[14] bert-base-uncased model — https://huggingface.co/bert-base-uncased.

6.1. Experiment design

We considered the following five approaches:

1. **BERT**, the uncased BERT model trained on text features that we adopted as a baseline, as described in Section 3.1.
2. **BERT DTI**: Direct Text Injection, which appends additional knowledge at the end of the input text, as described in Section 5.1.
3. **K BERT**: Knowledge BERT, which augments identified entities with relevant predicates and objects from the knowledge graph, as described in Section 5.2.
4. **BERT MLP**: Integration of additional features using a Multilayer Perceptron, the strategy that combines the BERT outputs with symbolic features as described in Section 5.3.
5. **BERT PT** Additional pre-training on the KG, which further pre-train BERT on a text generated by concatenating all triples in the KG, as described in Section 5.4.

For all the experiments reported in this manuscript, we employed 1500 documents for the development dataset and another 1500 documents for the test dataset. Each set comprises 500 documents for each label (AI, SE, and HCI). By consistently using the same development and test datasets across all approaches, we ensured that the performance outcomes were comparable.

We varied the size of the training datasets to evaluate the effect of different training sizes on model performance. Dataset sizes ranged from 3000 to 21,000 articles, increasing in increments of 3000 articles. In line with the findings reported in Dodge et al. (2020), we run each configuration with ten different random seeds. The performances of the proposed approaches have been measured by using the micro-average of the F1-score. The standard deviations of the F1-scores were consistently below 1 , as shown in Table 2, indicating robust and reliable results.

In all experiments, BERT was fine-tuned over five epochs. The training learning rate was set at 2×10^{-5}, the size of the sentence embedding vector was configured to 384, and the batch size was set to 6. The optimisation method used was Adam, and the dropout probability was set at 0.1. The statistical tests followed the standard procedure for constructing and comparing Receiver Operating Characteristic (ROC) curves in a paired-sample scenario. The significance of the difference between the two methods was assessed based on the difference in the two Area Under the Curve (AUC) values, the relative standard deviation, and the number of true positives and true negatives associated with each curve. A p-value 0.05 was considered significant, following standard practice.

Table 2
Precision, Recall, and F1-scores obtained for the five models at different sizes of the training set. We highlight in bold the best results.

Train size	BERT						BERT-DTI					
	PRECISION		RECALL		F1-SCORE		PRECISION		RECALL		F1-SCORE	
	Avg	Std	Avg	Std	Avg	Std	Avg	Std	Avg	Std	Avg	Std
3000	0.872	0.013	0.839	0.020	0.855	0.008	0.877	0.010	0.841	0.008	0.858	0.009
6000	0.885	0.008	0.836	0.015	0.860	0.010	0.885	0.010	0.843	0.004	0.863	0.003
9000	0.877	0.012	0.859	0.010	0.868	0.005	0.877	0.014	0.859	0.020	0.868	0.005
12000	0.882	0.007	0.845	0.016	0.863	0.006	0.881	0.012	0.854	0.013	0.867	0.005
15000	0.879	0.024	0.858	0.014	0.868	0.016	0.880	0.007	0.868	0.013	0.874	0.006
18000	0.894	0.012	0.852	0.010	0.872	0.007	0.878	0.005	0.865	0.006	0.871	0.001
21000	0.888	0.015	0.862	0.015	0.874	0.006	0.894	0.009	0.868	0.003	0.881	0.005
Train size	K-BERT						BERT-MLP					
	PRECISION		RECALL		F1-SCORE		PRECISION		RECALL		F1-SCORE	
	Avg	Std	Avg	Std	Avg	Std	Avg	Std	Avg	Std	Avg	Std
3000	0.892	0.010	0.850	0.012	**0.869**	0.009	0.879	0.013	0.824	0.015	0.850	0.010
6000	0.879	0.007	0.871	0.012	**0.871**	0.007	0.880	0.012	0.852	0.015	0.866	0.009
9000	0.877	0.005	0.864	0.005	0.870	0.006	0.902	0.011	0.860	0.032	**0.880**	0.019
12000	0.890	0.009	0.873	0.014	0.871	0.008	0.884	0.012	0.858	0.007	**0.880**	0.007
15000	0.891	0.008	0.874	0.016	0.879	0.005	0.914	0.005	0.875	0.016	**0.894**	0.008
18000	0.882	0.010	0.894	0.012	0.881	0.007	0.905	0.014	0.895	0.013	**0.900**	0.006
21000	0.884	0.005	0.881	0.005	0.880	0.004	0.905	0.016	0.910	0.011	**0.907**	0.007
Train size	BERT-PT											
	PRECISION		RECALL		F1-SCORE							
	Avg	Std	Avg	Std	Avg	Std						
3000	0.874	0.016	0.828	0.013	0.850	0.002						
6000	0.884	0.012	0.822	0.011	0.852	0.003						
9000	0.883	0.006	0.848	0.007	0.865	0.005						
12000	0.880	0.012	0.842	0.008	0.861	0.005						
15000	0.884	0.016	0.853	0.023	0.868	0.005						
18000	0.886	0.008	0.855	0.009	0.870	0.007						
21000	0.870	0.006	0.870	0.007	0.870	0.006						

6.2. Results

Fig. 6 reports the F1-score obtained by the different methods in function of the size of the training set.

BERT MLP outperformed all the other methods for the largest training sizes (9K, 12K, 15K, 18K, 21K). Overall, it demonstrated the most substantial improvement over the BERT baseline, enhancing the F1-score by 2.8 and 3.3 for the 18K and 21K training sets, respectively. The statistical analysis revealed that the difference between **BERT MLP** and all other methods is statistically significant for training sizes exceeding 12K ($p < 0.0001$). Furthermore, BERT-MLP appeared to benefit from larger training sizes, steadily extending its lead over BERT as the dataset size increased.

K BERT demonstrated superior performance for smaller training sizes (3k, 6k), notably achieving a 1.1 F1-score improvement over **BERT** with the 6k dataset. The difference between **K BERT** and all other methods is significant for training sizes of 3K and 6K ($p < 0.0001$). However, in contrast to **BERT MLP**, **K BERT** did not seem to benefit from larger training sizes and gradually lost its advantage over **BERT** as the dataset size grew.

BERT DTI showed marginal enhancements over the standard BERT, suggesting that even straightforward methods that add knowledge to the text can be beneficial. However, more advanced approaches appear to yield significantly better outcomes.

Finally, **BERT PT** showed no significant improvement over the BERT baseline. This is likely due to the limited size of the knowledge base used for pre-training. This result implies that for similar scenarios, standard knowledge injection techniques might be more beneficial than pretraining the model on domain-specific data.

In summary, both **BERT MLP** and **K BERT** emerge as suitable options for this task, though they display very distinct behaviours. K-BERT excels with smaller training sets but exhibits diminished performance as the size of the training set increases. Conversely, BERT-MLP achieves excellent results with medium and large training sets.

7. Limitations and future works

The study presented in this manuscript is subject to certain limitations which warrant discussion. These limitations will be a focus for improvement in future research.

First, we exclusively adopt BERT as the foundational model for all experiments. The choice of BERT was strategic, as it serves as a consistent benchmark and remains a leading tool for many tasks in the field. However, it is crucial to explore the potential outcomes of applying the same knowledge injection methods to a variety of other recent Large Language Models, such as LLaMA 2 (Touvron et al., 2023). Broadening the scope to include these models in future work would enhance our understanding of the effectiveness of knowledge injection techniques across different LLMs. *Second*, we focus specifically on a classification task, while LLM can be used for a variety of other tasks in the scholarly domain, such as question answering (Auer et al., 2023), citation prediction (Gosangi et al., 2021), and information extraction (Dessí et al., 2022a). *Third*, the research is confined to the scholarly domain. It is conceivable that knowledge injection techniques might yield varied results in other sectors characterised by different styles, such as news articles, social media posts, and online reviews.

In the following, we discuss some preliminary experiments that demonstrate the flexibility of the techniques we have analysed in this paper. These experiments will also serve as a foundation for future research.

As a first example, we are currently exploring the task of question-answering over research data. Specifically, we are studying the application of a fine-tuned LLM to convert scientific questions in natural language into SPARQL queries, i.e., the *de facto* language to query knowledge graphs. For these experiments, we are using the SciQA benckhmark.[15] Our approach involves harnessing prompt engineering techniques for translation in two distinct ways.

[15] https://huggingface.co/datasets/orkg/SciQA?row=9

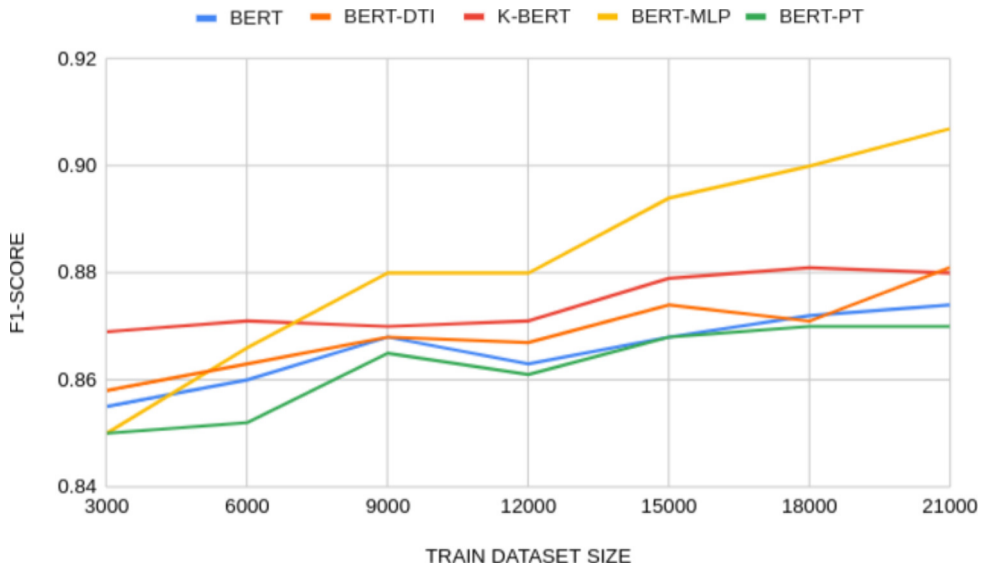


Fig. 6. Experimental evaluation of the five methods described in Section 5.

We first performed the translations using a one-shot learning approach. Subsequently, we enhanced the prompt by incorporating the knowledge stored in AIDA. This entails the identification of entities within the natural language text and then seeking triples in AIDA with the subject corresponding to any of the identified entities. As an example, let us consider the question *"Indicate the model that performed best in terms of Accuracy metric on the Kuzushiji-MNIST benchmark dataset?"*. From AIDA, it is possible to extract several statements related to the Kuzushiji-MNIST benchmark, among which there is the abstract of the associated paper, which can be given as input with the prompt.

Our initial experiments produced notable results. When using GPT 3.5, we successfully translated 287 out of 500 queries, meaning that the generated queries precisely matched the original ones. In further experiments, we have obtained encouraging results that can potentially beat this baseline. We are still investigating additional metrics and techniques. Our goal is to demonstrate the benefits of knowledge injection in tasks beyond classification.

With regard to the domain of application, we are currently assessing the generalisability of the same strategies presented in this paper also in the tourism domain. In particular, we conducted preliminary experiments of knowledge injection for two classification tasks in the tourism domain. The first task aims to predict the likelihood of an accommodation being booked at a given time. The second task is a review rating classification, which aims to predict whether an accommodation will be highly rated. To perform these tasks, we created training datasets of different sizes from 3000 to 12000, with a step of 3000, containing descriptions of tourist accommodations. We extracted these descriptions from the Tourism Knowledge Graph (Chessa et al., 2023) (TKG), which is based on AirBnb and describes lodging structures in London (UK). Additionally, we created fixed-size validation and test datasets, each containing 1800 descriptions. In these datasets, we described each accommodation according to a set of features. In particular, from TKG we extracted various numerical features such as number of bedrooms and beds, number of bathrooms, minimum night says, and so on. Then, we applied DBpedia Spotlight (Mendes et al., 2011) to the accommodation descriptions to identify the relevant touristic entities or amenities (e.g., air conditioning, wifi, parking space, etc.) available within the tourism ontology (Chessa et al., 2023) (TAO). DBpedia Spotlight is a cross-domain entity linking tool built upon the DBpedia public knowledge graph. We encoded all these features into a vector representation and employed it to test the BERT-MLP approach. Preliminary results show that the BERT-MLP knowledge injection technique outperformed BERT with an average increase of 12.5 points in F1 score (see Table 3).

These preliminary experiments provide further evidence of the generalisability and effectiveness of the knowledge injection strategies analysed in this paper.

In future research, we aim to extensively investigate new strategies across diverse domains and tasks. Our objective is to discern the optimal approaches tailored for specific scopes. This analysis will encompass the application of these techniques to pivotal areas, including but not limited to scientific research, tourism, and news analysis. Moreover, we intend to carry out a rigorous investigation regarding the ideal representation of knowledge for injection, also weighing the trade-offs between small and concise representations and more expansive, potentially noise-prone ones. This will involve an in-depth assessment of how these representations influence the performance of LLMs, enabling a deeper understanding of the interplay between knowledge formulation and model efficacy.

8. Conclusions

In this paper, we conducted a comprehensive examination of various knowledge injection methods tailored for transformer architectures in the context of scientific literature. This study offers a meticulous overview and a comparative evaluation of four predominant techniques, focusing on their effectiveness in the classification of scientific articles.

For this purpose, we have developed AIDA24k, a novel open-access benchmark encompassing 24,000 scientific papers sourced from the AIDA Knowledge Graph. This benchmark also incorporates a knowledge graph of 4629 research topics and 9258 statements, sourced from the Computer Science Ontology, intended to serve as supplementary knowledge for the classification tasks. The primary aim of this resource is to evaluate the overarching effectiveness of the various knowledge injection techniques.

The comparative evaluation on AIDA24k indicates that both BERT-MLP and K-BERT are the best choices for classifying scientific articles. However, BERT-MLP is better suited for larger training sets, while K-BERT excels with smaller ones. Furthermore, the findings underscore that even very simple knowledge injection techniques, like appending metadata to the input text, can produce positive results.

To reproduce and further extend our work, we created a repository on GitHub containing the AIDA24k benchmark and the full codebase of the five approaches.

Table 3
Preliminary experiment on the tourism domain — results on "Visit propensity" and "User review score" classification tasks: average values for F1-score, precision, and recall.

Visit propensity classification task						
Train size	BERT			BERT-MLP		
	PRECISION	RECALL	F1-SCORE	PRECISION	RECALL	F1-SCORE
3000	0.656	0.648	0.644	0.855	0.852	0.852
6000	0.668	0.681	0.673	0.859	0.857	0.857
9000	0.670	0.676	0.670	0.861	0.857	0.856
12000	0.676	0.685	0.679	0.863	0.861	0.860
User review score classification task						
Train size	BERT			BERT-MLP		
	PRECISION	RECALL	F1-SCORE	PRECISION	RECALL	F1-SCORE
3000	0.631	0.616	0.620	0.705	0.699	0.698
6000	0.639	0.646	0.639	0.708	0.697	0.694
9000	0.644	0.641	0.642	0.710	0.697	0.694
12000	0.648	0.647	0.645	0.715	0.702	0.698

CRediT authorship contribution statement

Andrea Cadeddu: Software, Investigation. **Alessandro Chessa:** Project administration, Supervision. **Vincenzo De Leo:** Software, Validation, Investigation, Writing – original draft. **Gianni Fenu:** Funding acquisition, Project administration. **Enrico Motta:** Project administration. **Francesco Osborne:** Conceptualization, Methodology, Formal analysis, Data curation, Writing – review & editing, Supervision. **Diego Reforgiato Recupero:** Conceptualization, Methodology, Formal analysis, Data curation, Writing – review & editing, Supervision. **Angelo Salatino:** Software, Validation, Resources, Data curation, Writing – review & editing, Visualization. **Luca Secchi:** Software, Validation, Investigation, Writing – original draft.

Declaration of competing interest

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

Data availability

I have shared the data on a public repository in the paper.

Acknowledgements

We acknowledge financial support under the National Recovery and Resilience Plan (NRRP), Mission 4 Component 2 Investment 1.5 - Call for tender No. 3277 published on December 30, 2021 by the Italian Ministry of University and Research (MIUR) funded by the European Union – NextGenerationEU. Project Code ECS0000038 – Project Title eINS Ecosystem of Innovation for Next Generation Sardinia – CUP F53C22000430001- Grant Assignment Decree No. 1056 adopted on June 23, 2022 by the Italian Ministry of University and Research (MIUR).

References

Aggarwal, T., Salatino, A., Osborne, F., Motta, E., 2022. R-classify: Extracting research papers' relevant concepts from a controlled vocabulary. Softw. Impacts 14, http://dx.doi.org/10.1016/j.simpa.2022.100444, Publisher: Elsevier.

Al-Moslmi, T., Ocaña, M.G., Opdahl, A.L., Veres, C., 2020. Named entity extraction for knowledge graphs: A literature overview. IEEE Access 8, 32862–32881.

Alkaissi, H., McFarlane, S.I., 2023. Artificial hallucinations in ChatGPT: implications in scientific writing. Cureus 15 (2).

Amizadeh, S., Palangi, H., Polozov, O., Huang, Y., Koishida, K., 2020. Neuro-symbolic visual reasoning: Disentangling "visual" from "reasoning". arXiv:2006.11524.

Angioni, S., Salatino, A., Osborne, F., Recupero, D.R., Motta, E., 2020. The AIDA dashboard: Analysing conferences with semantic technologies. In: 19th International Semantic Web Conference. ISWC 2020, URL http://oro.open.ac.uk/72293/.

Angioni, S., Salatino, A., Osborne, F., Recupero, D.R., Motta, E., 2021. AIDA: A knowledge graph about research dynamics in academia and industry. Quant. Sci. Stud. 2 (4), 1356–1398.

Auer, S., Barone, D.A., Bartz, C., Cortes, E.G., Jaradeh, M.Y., Karras, O., Koubarakis, M., Mouromtsev, D., Pliukhin, D., Radyush, D., et al., 2023. The SciQA scientific question answering benchmark for scholarly knowledge. Sci. Rep. 13 (1), 7240.

Barbieri, F., Camacho-Collados, J., Espinosa Anke, L., Neves, L., 2020. TweetEval: Unified benchmark and comparative evaluation for tweet classification. In: Findings of the Association for Computational Linguistics. EMNLP 2020, Association for Computational Linguistics, Online, pp. 1644–1650. http://dx.doi.org/10.18653/v1/2020.findings-emnlp.148, URL https://aclanthology.org/2020.findings-emnlp.148.

Beck, M., Rizvi, S.T.R., Dengel, A., Ahmed, S., 2020. From automatic keyword detection to ontology-based topic modeling. In: International Workshop on Document Analysis Systems. Springer, pp. 451–465. http://dx.doi.org/10.1007/978-3-030-57058-3_32.

Borges, M.V.M., dos Reis, J.C., 2019. Semantic-enhanced recommendation of video lectures. In: 2019 IEEE 19th International Conference on Advanced Learning Technologies, vol. 2161. ICALT, IEEE, pp. 42–46. http://dx.doi.org/10.1109/ICALT.2019.00013.

Bosselut, A., Rashkin, H., Sap, M., Malaviya, C., Celikyilmaz, A., Choi, Y., 2019. COMET: Commonsense transformers for automatic knowledge graph construction. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics, Florence, Italy, pp. 4762–4779. http://dx.doi.org/10.18653/v1/P19-1470, URL https://aclanthology.org/P19-1470.

Caselli, T., Basile, V., Mitrovic, J., Granitzer, M., 2020. HateBERT: Retraining BERT for abusive language detection in english. CoRR abs/2010.12472. URL https://arxiv.org/abs/2010.12472.

Chamorro-Padial, J., Rodríguez-Sánchez, R., 2023. Attention-survival score: A metric to choose better keywords and improve visibility of information. Algorithms 16 (4), http://dx.doi.org/10.3390/a16040196, URL https://www.mdpi.com/1999-4893/16/4/196.

Chari, S., Seneviratne, O., Ghalwash, M., Shirai, S., Gruen, D.M., Meyer, P., Chakraborty, P., McGuinness, D.L., 2023. Explanation Ontology: A general-purpose, semantic representation for supporting user-centered explanations. Semant Web Preprint (Preprint), 1–31. http://dx.doi.org/10.3233/SW-233282, Publisher: IOS Press.

Chatzopoulos, S., Vergoulis, T., Kanellos, I., Dalamagas, T., Tryfonopoulos, C., 2020. Artsim: improved estimation of current impact for recent articles. In: ADBIS, TPDL and EDA 2020 Common Workshops and Doctoral Consortium. Springer, pp. 323–334. http://dx.doi.org/10.1007/978-3-030-55814-7_27.

Chessa, A., Fenu, G., Motta, E., Osborne, F., Recupero, D.R., Salatino, A.A., Secchi, L., 2023. Data-driven methodology for knowledge graph generation within the tourism domain. IEEE Access 11, 67567–67599. http://dx.doi.org/10.1109/ACCESS.2023.3292153.

Dessí, D., Osborne, F., Recupero, D.R., Buscaldi, D., Motta, E., 2022a. SCICERO: A deep learning and NLP approach for generating scientific knowledge graphs in the computer science domain. Knowl.-Based Syst. 258, 109945.

Dessí, D., Osborne, F., Reforgiato Recupero, D., Buscaldi, D., Motta, E., 2022b. CS-kg: A large-scale knowledge graph of research entities and claims in computer science. In: The Semantic Web–ISWC 2022: 21st International Semantic Web Conference, Virtual Event, October 23–27, 2022, Proceedings. Springer, pp. 678–696.

Devlin, J., Chang, M.-W., Lee, K., Toutanova, K., 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). Association for Computational Linguistics, Minneapolis, Minnesota, pp. 4171–4186. http://dx.doi.org/10.18653/v1/N19-1423, URL https://www.aclweb.org/anthology/N19-1423.

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

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

Gangopadhyay, B., Hazra, S., Dasgupta, P., 2021. Semi-lexical languages: a formal basis for using domain knowledge to resolve ambiguities in deep-learning based computer vision. Pattern Recognit. Lett. 152, 143–149. http://dx.doi.org/10.1016/j.patrec.2021.10.004, URL https://www.sciencedirect.com/science/article/pii/S0167865521003615.

Gao, S., Alawad, M., Young, M.T., Gounley, J., Schaefferkoetter, N., Yoon, H.J., Wu, X.C., Durbin, E.B., Doherty, J., Stroup, A., et al., 2021. Limitations of transformers on clinical text classification. IEEE J. Biomed. Health Inform. 25 (9), 3596–3607.

Gosangi, R., Arora, R., Gheisarieha, M., Mahata, D., Zhang, H., 2021. On the use of context for predicting citation worthiness of sentences in scholarly articles. In: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Association for Computational Linguistics, Online, pp. 4539–4545. http://dx.doi.org/10.18653/v1/2021.naacl-main.359, URL https://aclanthology.org/2021.naacl-main.359.

Guu, K., Lee, K., Tung, Z., Pasupat, P., Chang, M.W., 2020. REALM: Retrieval-augmented language model pre-training. In: Proceedings of the 37th International Conference on Machine Learning. ICML '20, JMLR.org.

Han, K., 2023. Incorporating knowledge resources into natural language processing techniques to advance academic research and application development.

Hitzler, P., 2021. A review of the semantic web field. Commun. ACM 64 (2), 76–83.

Joshi, M., Lee, K., Luan, Y., Toutanova, K., 2021. Contextualized representations using textual encyclopedic knowledge. arXiv:2004.12006.

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

Ke, P., Ji, H., Liu, S., Zhu, X., Huang, M., 2020. SentiLARE: Sentiment-aware language representation learning with linguistic knowledge. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing. EMNLP, Association for Computational Linguistics, Online, pp. 6975–6988. http://dx.doi.org/10.18653/v1/2020.emnlp-main.567, URL https://aclanthology.org/2020.emnlp-main.567.

Kim, S.W., Gil, J.M., 2019. Research paper classification systems based on TF-IDF and LDA schemes. Hum. Centr. Comput. Inf. Sci. 9, 1–21.

Kumar, K., 2023. Geotechnical Parrot Tales (GPT): Overcoming GPT hallucinations with prompt engineering for geotechnical applications. arXiv preprint arXiv:2304.02138.

Kumar, V., Recupero, D.R., Helaoui, R., Riboni, D., 2022. K-LM: knowledge augmenting in language models within the scholarly domain. IEEE Access 10, 91802–91815. http://dx.doi.org/10.1109/ACCESS.2022.3201542.

Kung, T.H., Cheatham, M., Medenilla, A., Sillos, C., De Leon, L., Elepaño, C., Madriaga, M., Aggabao, R., Diaz-Candido, G., Maningo, J., et al., 2023. Performance of ChatGPT on USMLE: Potential for AI-assisted medical education using large language models. PLoS Dig. Health 2 (2), e0000198.

Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C.H., Kang, J., 2019. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. CoRR abs/1901.08746. URL http://arxiv.org/abs/1901.08746.

Leivaditi, S., Rossi, J., Kanoulas, E., 2020. A benchmark for lease contract review. CoRR abs/2010.10386. URL https://arxiv.org/abs/2010.10386.

Lerer, A., Wu, L., Shen, J., Lacroix, T., Wehrstedt, L., Bose, A., Peysakhovich, A., 2019. PyTorch-BigGraph: A large-scale graph embedding system. arXiv:1903.12287.

Li, W., Zhou, H., Dong, J., Zhang, Q., Li, Q., Baciu, G., Cao, J., Huang, X., 2023. Constructing low-redundant and high-accuracy knowledge graphs for education. In: Learning Technologies and Systems: 21st International Conference on Web-Based Learning, ICWL 2022, and 7th International Symposium on Emerging Technologies for Education, SETE 2022, Tenerife, Spain, November 21–23, 2022, Revised Selected Papers. Springer-Verlag, Berlin, Heidelberg, pp. 148–160. http://dx.doi.org/10.1007/978-3-031-33023-0_13.

Liu, P., Neubig, G., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., Neubig, G., 2023. Pre-train , prompt , and predict : A systematic survey of prompting methods in natural language processing. ACM Comput. Surv. 55, 1–46. http://dx.doi.org/10.1145/3560815, arXiv:2107.13586v1.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V., 2019a. RoBERTa: A robustly optimized BERT pretraining approach. CoRR abs/1907.11692. URL http://arxiv.org/abs/1907.11692.

Liu, W., Zhou, P., Zhao, Z., Wang, Z., Ju, Q., Deng, H., Wang, P., 2019b. K-BERT: Enabling language representation with knowledge graph. URL http://arxiv.org/abs/1909.07606.

Löffler, F., Wesp, V., Babalou, S., Kahn, P., Lachmann, R., Sateli, B., Witte, R., König-Ries, B., 2020. ScholarLensViz: A visualization framework for transparency in semantic user profiles. In: Taylor, K., Gonçalves, R., Lecue, F., Yan, J. (Eds.), Proceedings of the ISWC 2020 Demos and Industry Tracks: From Novel Ideas to Industrial Practice Co-Located with 19th International Semantic Web Conference. ISWC 2020, Globally Online, November 1-6, 2020 UTC.

Mardiah, M., Neyman, S.N., et al., 2023. Aggregate functions in categorical data skyline search (CDSS) for multi-keyword document search. Khazanah Informatika: Jurnal Ilmu Komputer dan Informatika 9 (1).

Meloni, A., Angioni, S., Salatino, A., Osborne, F., Reforgiato Recupero, D., Motta, E., 2023. Integrating conversational agents and knowledge graphs within the scholarly domain. IEEE Access 11, 22468–22489. http://dx.doi.org/10.1109/ACCESS.2023.3253388.

Mendes, P.N., Jakob, M., García-Silva, A., Bizer, C., 2011. Dbpedia spotlight: Shedding light on the web of documents. In: Proceedings of the 7th International Conference on Semantic Systems. I-Semantics '11, Association for Computing Machinery, New York, NY, USA, pp. 1–8. http://dx.doi.org/10.1145/2063518.2063519.

Moiseev, F., Dong, Z., Alfonseca, E., Jaggi, M., 2022a. SKILL: Structured knowledge infusion for large language models. In: Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Association for Computational Linguistics, Seattle, United States, pp. 1581–1588. http://dx.doi.org/10.18653/v1/2022.naacl-main.113, URL https://aclanthology.org/2022.naacl-main.113.

Moiseev, F., Dong, Z., Alfonseca, E., Jaggi, M., 2022b. SKILL: structured knowledge infusion for large language models. arXiv preprint arXiv:2205.08184.

Nayyeri, M., Cil, G.M., Vahdati, S., Osborne, F., Rahman, M., Angioni, S., Salatino, A., Recupero, D.R., Vassilyeva, N., Motta, E., et al., 2021. Trans4E: Link prediction on scholarly knowledge graphs. Neurocomputing 461, 530–542. http://dx.doi.org/10.1016/j.neucom.2021.02.100.

OpenAI, 2023. GPT-4 technical report. arXiv:2303.08774.

Osborne, F., Motta, E., 2015. Klink-2: Integrating multiple web sources to generate semantic topic networks. In: Arenas, M., Corcho, O., Simperl, E., Strohmaier, M., d'Aquin, M., Srinivas, K., Groth, P., Dumontier, M., Heflin, J., Thirunarayan, K., Thirunarayan, K., Staab, S. (Eds.), The Semantic Web. ISWC 2015, Springer International Publishing, Cham, pp. 408–424. http://dx.doi.org/10.1007/978-3-319-25007-6_24.

Osborne, F., Motta, E., Mulholland, P., 2013. Exploring scholarly data with rexplore. In: Alani, H., Kagal, L., Fokoue, A., Groth, P., Biemann, C., Parreira, J.X., Aroyo, L., Noy, N., Welty, C., Janowicz, K. (Eds.), The Semantic Web. ISWC 2013, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 460–477. http://dx.doi.org/10.1007/978-3-642-41335-3_29.

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

Peng, C., Xia, F., Naseriparsa, M., Osborne, F., 2023. Knowledge graphs: opportunities and challenges. Artif. Intell. Rev. 1–32.

Qin, Y., Lin, Y., Takanobu, R., Liu, Z., Li, P., Ji, H., Huang, M., Sun, M., Zhou, J., 2021. ERICA: Improving entity and relation understanding for pre-trained language models via contrastive learning. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). Association for Computational Linguistics, Online, pp. 3350–3363. http://dx.doi.org/10.18653/v1/2021.acl-long.260, URL https://aclanthology.org/2021.acl-long.260.

Reimers, N., Gurevych, I., 2019. Sentence-BERT: Sentence embeddings using siamese BERT-networks. pp. 3973–3983. http://dx.doi.org/10.18653/v1/D19-1410.

Rizvi, S.T.R., Ahmed, S., Dengel, A., 2023. ACE 2.0: A comprehensive tool for automatic extraction, analysis, and digital profiling of the researchers in Scientific Communities. Soc. Netw. Anal. Min. 13 (1), 81.

Rossanez, A., dos Reis, J.C., da Silva Torres, R., 2020. Representing scientific literature evolution via temporal knowledge graphs.

Salatino, A., Angioni, S., Osborne, F., Recupero, D.R., Motta, E., 2023. Diversity of expertise is key to scientific impact: a large-scale analysis in the field of computer science. http://dx.doi.org/10.55835/6442f3fd947802668eee976c, URL https://dapp.orvium.io/deposits/6442f3fd947802668eee976c/view.

Salatino, A.A., Osborne, F., Birukou, A., Motta, E., 2019a. Improving editorial workflow and metadata quality at springer nature. In: Ghidini, C., Hartig, O., Maleshkova, M., Svátek, V., Cruz, I., Hogan, A., Song, J., Lefrançois, M., Gandon, F. (Eds.), The Semantic Web. ISWC 2019, Springer International Publishing, Cham, pp. 507–525. http://dx.doi.org/10.1007/978-3-030-30796-7_31.

Salatino, A.A., Osborne, F., Motta, E., 2018a. AUGUR: Forecasting the emergence of new research topics. In: Proceedings of the 18th ACM/IEEE on Joint Conference on Digital Libraries. JCDL '18, Association for Computing Machinery, New York, NY, USA, pp. 303–312. http://dx.doi.org/10.1145/3197026.3197052.

Salatino, A., Osborne, F., Motta, E., 2022. CSO classifier 3.0: a scalable unsupervised method for classifying documents in terms of research topics. Int. J. Dig. Lib. 1–20.

Salatino, A.A., Osborne, F., Thanapalasingam, T., Motta, E., 2019b. The CSO classifier: Ontology-driven detection of research topics in scholarly articles. In: Doucet, A., Isaac, A., Golub, K., Aalberg, T., Jatowt, A. (Eds.), Digital Libraries for Open Knowledge. Springer International Publishing, Cham, pp. 296–311. http://dx.doi.org/10.1007/978-3-030-30760-8_26.

Salatino, A.A., Thanapalasingam, T., Mannocci, A., Osborne, F., Motta, E., 2018b. The computer science ontology: A large-scale taxonomy of research areas. In: Vrandečić, D., Bontcheva, K., Suárez-Figueroa, M.C., Presutti, V., Celino, I., Sabou, M., Kaffee, L.-A., Simperl, E. (Eds.), The Semantic Web. ISWC 2018, Springer International Publishing, Cham, pp. 187–205. http://dx.doi.org/10.1007/978-3-030-00668-6_12.

Su, Y., Han, X., Zhang, Z., Lin, Y., Li, P., Liu, Z., Zhou, J., Sun, M., 2021. CokeBERT: Contextual knowledge selection and embedding towards enhanced pre-trained language models. AI Open 2, 127–134. http://dx.doi.org/10.1016/j.aiopen.2021.06.004, URL https://www.sciencedirect.com/science/article/pii/S2666651021000188.

Sun, T., Shao, Y., Qiu, X., Guo, Q., Hu, Y., Huang, X., Zhang, Z., 2020. CoLAKE: Contextualized language and knowledge embedding. In: Proceedings of the 28th International Conference on Computational Linguistics. International Committee on Computational Linguistics, Barcelona, Spain (Online), pp. 3660–3670. http://dx.doi.org/10.18653/v1/2020.coling-main.327, URL https://aclanthology.org/2020.coling-main.327.

Thanapalasingam, T., Osborne, F., Birukou, A., Motta, E., 2018. Ontology-based recommendation of editorial products. In: Vrandečić, D., Bontcheva, K., Suárez-Figueroa, M.C., Presutti, V., Celino, I., Sabou, M., Kaffee, L.-A., Simperl, E. (Eds.), The Semantic Web. ISWC 2018, Springer International Publishing, Cham, pp. 341–358. http://dx.doi.org/10.1007/978-3-030-00668-6_21.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al., 2023. Llama 2: Open foundation and fine-tuned chat models. arXiv preprint arXiv:2307.09288.

Vergoulis, T., Chatzopoulos, S., Dalamagas, T., Tryfonopoulos, C., 2020. VeTo: Expert set expansion in academia. In: Hall, M., Merčun, T., Risse, T., Duchateau, F. (Eds.), Digital Libraries for Open Knowledge. Springer International Publishing, Cham, pp. 48–61. http://dx.doi.org/10.1007/978-3-030-54956-5_4.

Wang, X., Gao, T., Zhu, Z., Zhang, Z., Liu, Z., Li, J., Tang, J., 2021a. KEPLER: A unified model for knowledge embedding and pre-trained language representation. Trans. Assoc. Comput. Linguist. 9, 176–194. http://dx.doi.org/10.1162/tacl_a_00360, URL https://aclanthology.org/2021.tacl-1.11.

Wang, R., Tang, D., Duan, N., zhongyu wei, Huang, X., Ji, J., Cao, G., Jiang, D., Zhou, M., 2021b. K-adapter: Infusing knowledge into pre-trained models with adapters. URL https://openreview.net/forum?id=CLnj31GZ4cI.

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

Yamada, I., Asai, A., Shindo, H., Takeda, H., Matsumoto, Y., 2020. LUKE: Deep contextualized entity representations with entity-aware self-attention. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing. EMNLP, Association for Computational Linguistics, Online, pp. 6442–6454. http://dx.doi.org/10.18653/v1/2020.emnlp-main.523, URL https://aclanthology.org/2020.emnlp-main.523.

Yang, J., Xiao, G., Shen, Y., Jiang, W., Hu, X., Zhang, Y., Peng, J., 2021. A survey of knowledge enhanced pre-trained models. pp. 1–19, URL http://arxiv.org/abs/2110.00269.

Zhang, X., Chandrasegaran, S., Ma, K.-L., 2021. ConceptScope: Organizing and visualizing knowledge in documents based on domain ontology. In: Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems. pp. 1–13.

Zhang, Z., Han, X., Liu, Z., Jiang, X., Sun, M., Liu, Q., 2019. ERNIE: Enhanced language representation with informative entities. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics, Florence, Italy, pp. 1441–1451. http://dx.doi.org/10.18653/v1/P19-1139, URL https://aclanthology.org/P19-1139.