Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence (IJCAI-24)
Demonstrations Track

KnowledgeHub: An End-to-End Tool for Assisted Scientific Discovery

Shinnosuke Tanaka[1], **James Barry**[1], **Vishnudev Kuruvanthodi**[1], **Movina Moses**[2], **Maxwell J. Giammona**[2], **Nathan Herr**[3*], **Mohab Elkaref**[1] and **Geeth De Mel**[1]
[1]IBM Research Europe
[2]IBM Research
[3]University College London
{shinnosuke.tanaka, james.barry, vishnudev.k, movina.moses, maxwell.giammona, mohab.elkaref}@ibm.com, uceenhe@ucl.ac.uk, geeth.demel@uk.ibm.com

Abstract

This paper describes the KnowledgeHub tool, a scientific literature Information Extraction (IE) and Question Answering (QA) pipeline. This is achieved by supporting the ingestion of PDF documents that are converted to text and structured representations. An ontology can then be constructed where a user defines the types of entities and relationships they want to capture. A browser-based annotation tool enables annotating the contents of the PDF documents according to the ontology. Named Entity Recognition (NER) and Relation Classification (RC) models can be trained on the resulting annotations and can be used to annotate the unannotated portion of the documents. A knowledge graph is constructed from these entity and relation triples which can be queried to obtain insights from the data. Furthermore, we integrate a suite of Large Language Models (LLMs) that can be used for QA and summarisation that is grounded in the included documents via a retrieval component. KnowledgeHub is a unique tool that supports annotation, IE and QA, which gives the user full insight into the knowledge discovery pipeline.

1 Introduction

Accelerating scientific discovery has long been the goal of researchers and subject matter experts (SMEs) alike. The growing amount of data contained within the scientific literature means that automated solutions are becoming increasingly necessary to efficiently extract information to develop new discoveries. Advances in Artificial Intelligence (AI) and Natural Language Processing (NLP) research have led to larger and more capable models such as BERT [Devlin *et al.*, 2019] that can be used as feature extractors for token-classification tasks that have been the cornerstone for Information Extraction (IE) tasks such as Named Entity Recognition (NER) and Relation Classification (RC) between entities. Additionally, developments in Large Language Models (LLMs) that predict tokens in an autoregressive manner [Radford *et al.*, 2019; Brown *et al.*, 2020; Touvron *et al.*, 2023] and innovative methods like Retrieval Augmented Generation (RAG) [Lewis *et al.*, 2020] have led to systems that can leverage vast amounts of internal and external knowledge to enhance the suitability and factual correctness of LLM responses.

Several annotation tools have been created that focus on PDF layout annotation, linguistic annotation or IE through NER and RC. There also exist tools that perform QA over the documents. In the context of the above, tools such as PAWLS [Neumann *et al.*, 2021] enable users to annotate PDF document regions with labelled bounding boxes where the tool will then predict layout regions on other files. Linguistic annotation tools such as PDFAnno [Shindo *et al.*, 2018], AnnIE [Friedrich *et al.*, 2022] and Autodive [Du *et al.*, 2023] focus on annotating data for IE tasks such as NER and RC. Similarly, BatteryDataExtractor [Huang and Cole, 2022] is a tool that focuses on IE for the battery domain with an additional QA component but does not support annotation.

This paper introduces KnowledgeHub[1], a novel tool that covers the fundamental aspects of the knowledge discovery process: including linguistic annotation, IE with NER and RC models, and QA that is grounded in the source literature. To achieve this, we implement a pipeline where a user submits a collection of PDF documents for their field of study, which are then converted to text and structured representations. A user-defined ontology can then be supplied which defines the types of entities and relationships to consider. A browser-based annotation tool enables annotating the contents of the PDF documents according to the ontology. NER and RC models can then be trained on the resulting annotations where the trained models can be used to automatically annotate the portion of unannotated documents. A knowledge graph is constructed from these entity and relation triples which can be queried to gain certain insights. Furthermore, we include an RAG based QA system. Out of the systems introduced so far, KnowledgeHub is the only tool that supports annotation, IE, and QA (see Table 1 for an overview of different tools).

2 System Description

This section describes the KnowledgeHub pipeline. The overall system is shown in Figure 1. At a high level, it is an application that consists of a frontend built using JavaScript, React

*Work done while at IBM.

[1]A video describing KnowledgeHub is available at this **link**

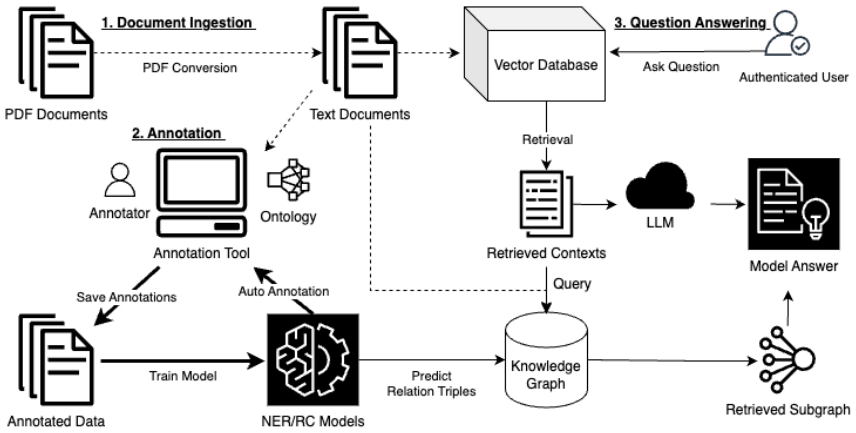

Figure 1: KnowledgeHub pipeline overview.

and the Carbon Design System library[2]. The frontend is connected to a backend using a Python Flask[3] web app. The backend consists of SQLite databases for storing information necessary for running the application. There is a Neo4j[4] graph database as well as a vector store which are used for building/querying knowledge graphs and performing RAG. All the components can be run locally except for the LLM in the RAG pipeline, which requires API calls. We also support a version that is hosted on OpenShift.

Firstly, a user defines a project and specifies the group members and the privileges each member should have. For that particular project, the remaining pipeline is discussed in the following sections.

Tool Name	IE	QA
AnnIE	NER, POS	-
Autodive*	NER	-
BatteryDataExtractor	NER, POS, AD	Extractive
PAWLS*	-	-
PDFAnno*	-	-
KnowledgeHub	NER, RC	LLM+RAG+KG

Table 1: Comparison with other tools. *Tool Name*: Superscript(*) represents support of direct PDF annotation. AD - Abbreviation Detection; Extractive - model predicts the answer span; LLM - model predicts answer with autoregressive LLM.

2.1 Document Ingestion

The user uploads an individual PDF file, or a zip file containing multiple PDF files. The PDF content is converted to structured XML using the GROBID tool[5], which predicts layout sections such as the title, headings, paragraphs, footnotes, references, tables and figures. We extract the text content inside the paragraph objects predicted by GROBID, which is then segmented, tokenised and annotated for Part-of-Speech (POS) information using the Stanza library [Qi *et al.*, 2020].

For our setting, each document is composed of paragraphs, and the text within the paragraphs is segmented into sentences. We create a Neo4j graph database where nodes are created at the document, paragraph, and sentence level. The sentence nodes are linked to their origin paragraph node, which in turn is linked to its origin document node. Metadata such as the title, authors, and year of publish can be stored for each document. As we will discuss in Section 2.2, NER and RC models are used to predict named entities and relations, where we also create nodes for the entity tokens and link them with the predicted relation.

The extracted text is also stored in a vector database, where we use a Chroma[6] database. We access an embedding model from the LangChain[7] library which offers several embedding models. Specifically, we use the `all-mpnet-base-v2` model released by the Sentence-Transformers library [Reimers and Gurevych, 2019] but note that this model can be easily replaced by a user to suit their needs by changing the model name or embedding provider.

2.2 Annotation

An ontology can be created manually in the browser interface or by importing external ontologies such as EMMO[8]. When importing an ontology, the user selects the entities and their associated relations. Similarly, our browser-based ontology creation tool enables i) choosing the entities, and ii) specifying the relation between certain entities. This produces a configuration file which lists the possible relation triples: `(entity1, relation, entity2)`. The user can map these entities and relations onto the contents of the PDF in a web browser using the BRAT annotation tool [Stenetorp *et al.*, 2012].

We implement our own models for NER and RC. These models are written in PyTorch [Paszke *et al.*, 2019] and involve placing a linear layer on top of a BERT-style model, where a user can specify an encoding model from the HuggingFace library [Wolf *et al.*, 2020]. The NER model in-

[2]https://carbondesignsystem.com/
[3]https://github.com/pallets/flask
[4]https://neo4j.com/
[5]https://github.com/kermitt2/grobid
[6]https://github.com/chroma-core/chroma
[7]https://github.com/langchain-ai/langchain
[8]https://github.com/emmo-repo/EMMO

volves a two-stage process: first, a span-based model predicts entity span regions including nested structures [Yu *et al.*, 2020], then an entity classifier model predicts the entity tag for the selected span [Elkaref *et al.*, 2023]. The RC model predicts a relation type (including no relation) between all pairs of entities in the sentence. The predictions of the NER/RC models are used to create connected entity nodes in the KG. A strength of using custom models is that we do not rely on external pipelines such as spaCy and we can train on any data where we have annotations.

We support two modes of auto-annotation: the first is based on regular expressions to label the target text based on entity names and their types defined by the user. The second is machine learning annotation, where the annotations from BRAT are saved to JSON and are then used to train NER and RC models. The trained models are then applied to the unlabelled data. This significantly reduces the user burden compared to manual annotation.

2.3 Question Answering

RAG is a method used to guide the generation process of an LLM by providing it with context-appropriate information, based on retrieving contexts that are most relevant to a user query, e.g. based on the cosine distance between an encoded query and the encoded documents [Lewis *et al.*, 2021]. KnowledgeHub lets users select a project and an LLM, e.g. a Llama [Touvron *et al.*, 2023] model and then ask a question. The three most relevant paragraphs from the project documents are retrieved. The LLM is then prompted through the IBM Generative AI Python SDK [IBM, 2024] to generate a summarised answer from these paragraphs, as well as individual answers from each paragraph. We also return a Neo4j subgraph showing all entity and relation objects from the three most relevant paragraphs. We leave integrating the graph structure into the LLM prompt as future work.

3 Use-case: Knowledge Discovery for the Battery Domain

In this section, we show how KnowledgeHub can be used for an example project related to the battery domain. The user starts by identifying and ingesting PDF documents related to their topic and creates their ontology based on BattINFO[9]. The user annotates a document d^1 with 150 entity types on 1,988 spans. They then train an NER model by fine-tuning `BatteryBERT-cased`[10] on d^1, and auto annotate a new document d^2 with 73 types on 1,464 spans. This out-of-domain (OOD) auto annotation yields a micro F1 score of 54.8% as shown in Table 2. This eliminates the need for users to annotate unknown documents from scratch, reducing the cost of annotations by more than half. After revising the annotations of d^2, the user trains a new NER model on d^1 and d^2, and auto annotates another new document d^3 with 96 types on 1,467 spans. In-domain (ID) results are 52.8%, 54.9% and 61.9% on d^1, $d^{1,2}$ and $d^{1,2,3}$, respectively, showing that repeating this process increases the performance of the model.

[9]https://github.com/BIG-MAP/BattINFO

[10]https://huggingface.co/batterydata/batterybert-cased

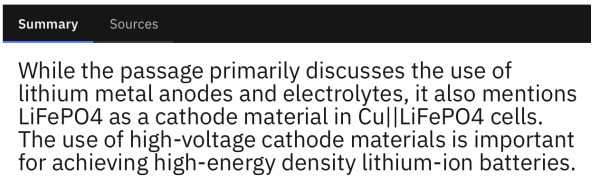

Figure 2: Part of summary of the model answers over retrieved contexts.

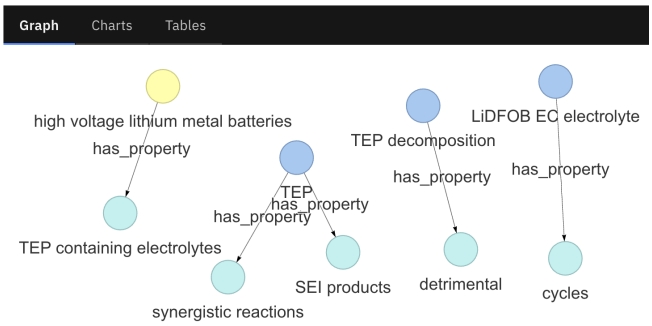

Figure 3: Part of the entities/relations contained in the retrieved contexts.

The user can also perform QA over their documents, where we show the summary using the instruction-tuned Mistral model[11] for an example question, *"What are promising cathode materials for high-voltage lithium-ion batteries?"* in Figure 2. The KG showing the entities and relations in the retrieved contexts is shown in Figure 3.

Setting	Train	Dev	F1
OOD	d^1	d^2	54.8
ID	d^1_{train}	d^1_{dev}	52.8
ID	d^1_{train}, d^2_{train}	d^1_{dev}, d^2_{dev}	54.9
ID	$d^1_{train}, \ldots, d^3_{train}$	$d^1_{dev}, \ldots d^3_{dev}$	61.9

Table 2: Performance of the auto annotation in Out Of Domain (OOD) and In Domain (ID) settings on documents d^1, d^2 and d^3.

4 Conclusion

In this paper we have presented KnowledgeHub, a tool for assisted scientific discovery by supporting IE tasks such as NER and RC. We also include a KG and an RAG component for grounded summarisation and QA, enhancing the factual correctness of LLM responses. We have demonstrated the usefulness of KnowledgeHub through an example where a researcher uses the tool for assisting their research for a project relating to batteries.

In future work, we would like to explore more ways of combining the graph information and the retrieved contexts. We would also like to implement QA based on non-text items in the PDF such as tables and figures. We wish to support direct annotation on the PDF content and improve functionality to support inter-annotator agreement.

[11]https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.2

Ethical Statement

We did not identify any ethical issues.

Acknowledgments

This work was supported by the Hartree National Centre for Digital Innovation (HNCDI), a collaboration between STFC and IBM.

References

[Brown *et al.*, 2020] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020.

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

[Du *et al.*, 2023] Yi Du, Ludi Wang, Mengyi Huang, Dongze Song, Wenjuan Cui, and Yuanchun Zhou. Autodive: An integrated onsite scientific literature annotation tool. In Danushka Bollegala, Ruihong Huang, and Alan Ritter, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 76–85, Toronto, Canada, July 2023. Association for Computational Linguistics.

[Elkaref *et al.*, 2023] Mohab Elkaref, Nathan Herr, Shinnosuke Tanaka, and Geeth De Mel. NLPeople at SemEval-2023 task 2: A staged approach for multilingual named entity recognition. In Atul Kr. Ojha, A. Seza Doğruöz, Giovanni Da San Martino, Harish Tayyar Madabushi, Ritesh Kumar, and Elisa Sartori, editors, *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 1148–1153, Toronto, Canada, July 2023. Association for Computational Linguistics.

[Friedrich *et al.*, 2022] Niklas Friedrich, Kiril Gashteovski, Mingying Yu, Bhushan Kotnis, Carolin Lawrence, Mathias Niepert, and Goran Glavaš. AnnIE: An annotation platform for constructing complete open information extraction benchmark. In Valerio Basile, Zornitsa Kozareva, and Sanja Stajner, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 44–60, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[Huang and Cole, 2022] Shu Huang and Jacqueline M. Cole. Batterydataextractor: battery-aware text-mining software embedded with bert models. *Chem. Sci.*, 13:11487–11495, 2022.

[IBM, 2024] IBM. Ibm generative ai python sdk (tech preview). https://github.com/IBM/ibm-generative-ai, 2024. Accessed: 2024-02-13.

[Lewis *et al.*, 2020] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA, 2020. Curran Associates Inc.

[Lewis *et al.*, 2021] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks, 2021.

[Neumann *et al.*, 2021] Mark Neumann, Zejiang Shen, and Sam Skjonsberg. PAWLS: PDF annotation with labels and structure. In Heng Ji, Jong C. Park, and Rui Xia, editors, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 258–264, Online, August 2021. Association for Computational Linguistics.

[Paszke *et al.*, 2019] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. *PyTorch: an imperative style, high-performance deep learning library*. Curran Associates Inc., Red Hook, NY, USA, 2019.

[Qi *et al.*, 2020] Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. Stanza: A python natural language processing toolkit for many human languages. In Asli Celikyilmaz and Tsung-Hsien Wen, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 101–108, Online, July 2020. Association for Computational Linguistics.

[Radford *et al.*, 2019] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[Reimers and Gurevych, 2019] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019.

[Shindo *et al.*, 2018] Hiroyuki Shindo, Yohei Munesada, and Yuji Matsumoto. PDFAnno: a web-based linguistic annotation tool for PDF documents. In Nicoletta Calzolari, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Koiti Hasida, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asuncion Moreno, Jan Odijk, Stelios Piperidis, and Takenobu Tokunaga, editors, *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, May 2018. European Language Resources Association (ELRA).

[Stenetorp *et al.*, 2012] Pontus Stenetorp, Sampo Pyysalo, Goran Topić, Tomoko Ohta, Sophia Ananiadou, and Jun'ichi Tsujii. brat: a web-based tool for NLP-assisted text annotation. In Frédérique Segond, editor, *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 102–107, Avignon, France, April 2012. Association for Computational Linguistics.

[Touvron *et al.*, 2023] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023.

[Wolf *et al.*, 2020] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. Transformers: State-of-the-art natural language processing. In Qun Liu and David Schlangen, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, October 2020. Association for Computational Linguistics.

[Yu *et al.*, 2020] Juntao Yu, Bernd Bohnet, and Massimo Poesio. Named entity recognition as dependency parsing. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6470–6476, Online, July 2020. Association for Computational Linguistics.