

ESRA: Explainable Scientific Research Assistant

Pollawat Hongwimol^{1*}, Peeranuth Kehasukcharoen^{1*}, Pasit Laohawarutchai^{1*},
Piyawat Lertvittayakumjorn², Aik Beng Ng³, Zhangsheng Lai³,
Timothy Liu³, Peerapon Vateekul^{1†}

¹ Department of Computer Engineering, Faculty of Engineering,
Chulalongkorn University, Thailand

² Department of Computing, Imperial College London, United Kingdom

³ NVIDIA

{6030400021, 6030416021, 6031039321}@student.chula.ac.th
pl1515@ic.ac.uk, {aikbengn, zlai, timothy1}@nvidia.com
peerapon.v@chula.ac.th

Abstract

We introduce Explainable Scientific Research Assistant (ESRA), a literature discovery platform that augments search results with relevant details and explanations, aiding users in understanding more about their queries and the returned papers beyond existing literature search systems. Enabled by a knowledge graph we extracted from abstracts of 23k papers on the arXiv’s cs.CL category, ESRA provides three main features: explanation (for why a paper is returned to the user), list of facts (that are relevant to the query), and graph visualization (drawing connections between the query and each paper with surrounding related entities). The experimental results with humans involved show that ESRA can accelerate the users’ search process with paper explanations and helps them better explore the landscape of the topics of interest by exploiting the underlying knowledge graph. We provide the ESRA web application at <http://esra.cp.eng.chula.ac.th/>.¹

1 Introduction

Existing literature search platforms mostly present metadata of papers as search results, and this requires users to read the entire abstracts to understand the brief contents of the returned papers. The users then need to reflect on the knowledge of the papers themselves so as to decide which keywords they should search next. Therefore, it is time-consuming to gradually expand their understanding of the field using existing platforms.

Meanwhile, research on analyzing scientific literature has been getting more attention due to the extremely large number of new papers published every day (Williams et al., 2014; Khan et al., 2017).

Also, many of them are freely accessible online and the number is still rising (Munroe, 2013). These lead to several frameworks that aim for extracting knowledge (i.e., scientific concepts and their relations) from scientific documents and representing them as a Knowledge Graph (KG) (Luan et al., 2018; Eberts and Ulges, 2019). However, to the best of our knowledge, most of the existing literature platforms have not yet leveraged such extracted knowledge graphs, but only the graph of metadata and hierarchical topics (Ammar et al., 2018; Sinha et al., 2015). So, they are not aware of relations among scientific entities in the papers (e.g., methods, models, and materials) resulting in an inability to provide insightful knowledge beyond a list of papers and abstracts.

In this paper, we develop Explainable Scientific Research Assistant (ESRA) – a literature discovery platform that utilizes a knowledge graph and modern Natural Language Processing (NLP) models to augment user experience. ESRA has three main features built around our extracted knowledge graph as illustrated partly in Figure 1. First, “the explanation feature” explains how the query and each returned paper are related. Second, “the fact list feature” suggests top-related keywords with their relationships to the query supporting exploration of related scientific concepts. Third, “the graph visualization feature” provides a subgraph illustrating related knowledge around the query and the returned papers. These features aim to help researchers quickly discover and understand a collection of literature they are looking for.

The strengths of the main features are demonstrated through a use case in Figure 1. Suppose users want to know about “BERT”, they initially enter “BERT” as the search query. On the top of the result page, there is a fact list displayed along with the graph visualization, showing facts (keywords) related to BERT such as “BERT is a subtype of

* Equal contributions

† Corresponding author

¹A brief demo of ESRA is available at <https://youtu.be/2RC6d4IFgIw>

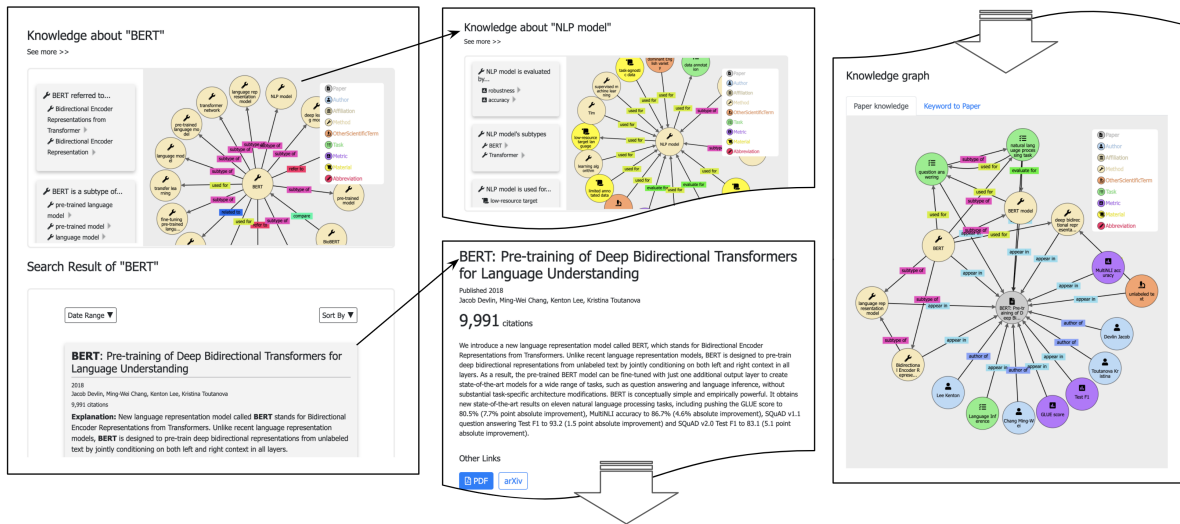


Figure 1: Searching scenario on the keyword “BERT” containing (left) the main result page, (middle top) routing to another keyword by clicking on the node, (middle bottom) meta data section of the paper page, (right) knowledge graph section of the paper page (continued from the middle bottom)

pre-trained language model” and “BERT is used for transfer learning”. Users can navigate to pages of the related keywords conveniently by clicking the node names as shown in Figure 1 (middle-top). From the middle to the bottom of that page, there is a list of returned papers containing their meta-data and explanations. For example, the explanation for the paper of RoBERTa (Liu et al., 2019) is “We present a replication study of BERT pre-training (Devlin et al., 2019) that carefully measures the impact of many key hyperparameters and training data size. We find that BERT was significantly undertrained, and can match or exceed the performance of every model published after it.” Users can click the paper title in order to redirect to the specific paper page which consists of all available metadata, knowledge graph visualization, references, and citations of the paper. With these features, users can quickly learn about the search query, check out related papers of their interest, and navigate to relevant concepts more conveniently.

2 Related Work

In this section, we present an overview of existing work related to ESRA along two topics, i.e., scientific knowledge extraction frameworks and scientific literature discovery platforms.

2.1 Scientific Knowledge Extraction Frameworks

In the past, research on information extraction (IE) for scientific texts focused mainly on citation re-

lations (Sim et al., 2012; Kas, 2011) and unsupervised extraction (Gábor et al., 2016). With the arrival of the SemEval shared tasks 2017 and 2018 (Augenstein et al., 2017; Gábor et al., 2018), the associated datasets enabled the research on supervised and semi-supervised learning for entity and relation extraction task for scientific papers. Since then, many research papers on supervised scientific IE have emerged. For example, SpERT (Eberts and Ulges, 2019) performs entity extraction and relation extraction jointly using pre-trained Transformers. DyGIE++ (Wadden et al., 2019) also jointly addresses the two tasks with the event extraction task. Besides, Luan et al. (2018) added the coreference resolution task into their IE framework, called SciIE, and created the SciERC dataset to support coreference resolution between cross-sentence entities for more detailed relations. Our framework combines SpERT and SciIE to cover both entity/relation extraction and coreference resolution, i.e., using SpERT for the former task and SciIE for the latter task.

2.2 Scientific Literature Discovery Platforms

There are various modern literature discovery platforms such as ACM Digital Library², IEEE Xplore³, Google Scholar⁴, Microsoft Academic⁵

²<https://dl.acm.org/>

³<https://ieeexplore.ieee.org/>

⁴<https://scholar.google.com/>

⁵<https://academic.microsoft.com/>

Feature / Platform	Microsoft Academic	Semantic Scholar	ORKG	AceMap	ESRA (ours)
Scientific Knowledge Graph	✗	✓	✗	✗	✓
Metadata Graph	✓	✓	✓	✓	✓
Explanation	✗	✗	✗	✗	✓
Fact List	✗	✗	✗	✗	✓
Graph Visualization	✗	✗	✓	✓	✓

Table 1: A feature comparison of existing graph-based literature platforms and our ESRA system

(Sinha et al., 2015), AceMap⁶ (Tan et al., 2016), ORKG⁷ (Jaradeh et al., 2019), and Semantic Scholar⁸ (Ammar et al., 2018). Most platforms are using the metadata of academic papers to rank and return results to their users. To the best of our knowledge, only Semantic Scholar uses scientific knowledge graph in their system. Table 1 compares prominent features of existing graph-based literature platforms to our ESRA system. We can see that the existing platforms focus on returning paper metadata as the search results without explaining why the papers are related to the query. In contrast, our ESRA system fills this gap by providing the explanations together with related scientific knowledge (via the fact list and the graph visualizations) to help the users better understand the query.

Besides the mentioned platforms, in the biomedical domain, there are many efforts to integrate knowledge bases into literature analysis systems. Similar to our fact list feature, Life-iNet (Ren et al., 2017) and BioTextQuest+ (Papanikolaou et al., 2014) are platforms that focus on exploring factual knowledge of a queried entity in the knowledge base and providing a list of supported documents. DeepLife (Ernst et al., 2016) and SetSearch+ (Shen et al., 2018) are entity-aware literature search engines that broaden results by expanding the query with related entities in the knowledge base. However, these platforms lack the ability to explain the relationship between the search query and the results. Our system uses the explanation and graph visualization feature to show the users how the query and the returned papers are related.

3 Explainable Scientific Research Assistant (ESRA)

Our goal is to create a scientific literature discovery platform that is explainable to users and helps

⁶<https://www.acemap.info/>

⁷<https://www.orkg.org/>

⁸<https://www.semanticscholar.org/>

them explore and expand knowledge more conveniently. This leads to the ESRA system with the following three main features, all of which leverage a knowledge graph we extracted from abstracts of the papers in our system.

Explanation: The explanation attached to each search result enables users to understand the reasons behind the recommendation of the system, i.e., why the paper is selected. The generated explanations for the same paper are dissimilar given different queries, making the explanations become specific to what the users want to know.

Fact list: For each query, ESRA displays related knowledge facts from the knowledge graph as a list for the users to explore. The goal of this feature is to aid users in having a better understanding of their search queries.

Graph visualization: Visualization gives users an understanding of the big picture of the relevant knowledge. In both the search result page and individual paper pages, the web application visualizes a subgraph of knowledge that is related to the search keyword and the papers, respectively.

To enable these three features, we implemented two main engines underlying ESRA as shown in Figure 2, including (1) a knowledge graph construction engine and (2) a web application engine. We will explain them and the overall system development in the next subsections.

3.1 Knowledge Graph Construction

Figure 2(a) shows the pipeline for extracting relations from scientific texts and constructing our knowledge graph. Given input texts (i.e., paper abstracts in our case), the pipeline works in three steps.

Step 1: Extraction The input abstracts are fed into an extractor which returns a list of extracted triples. The extractor consists of two models which

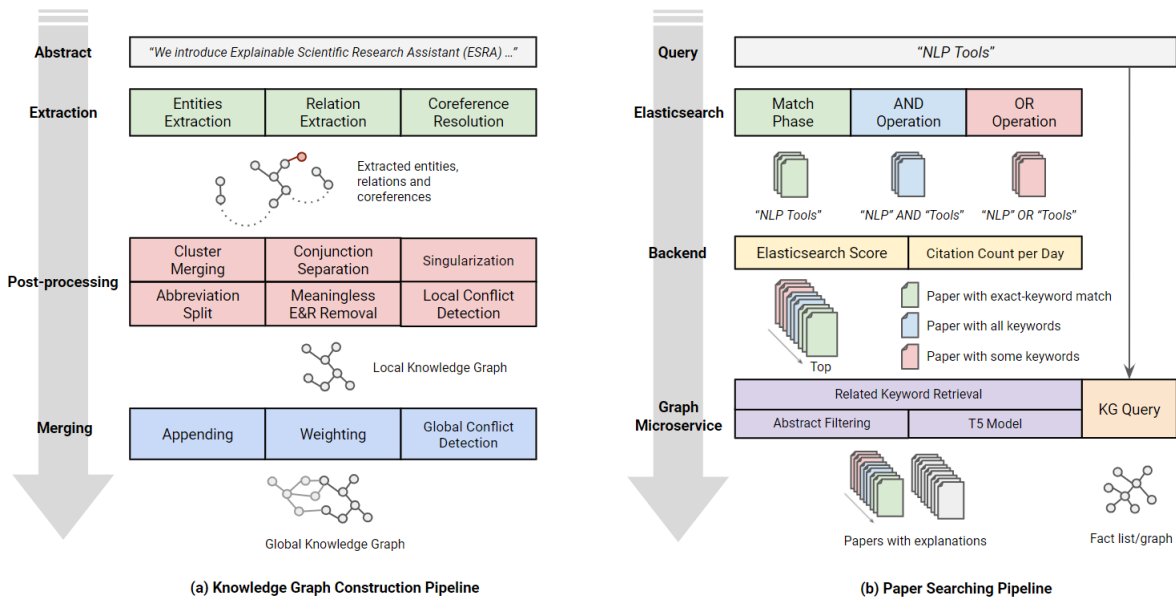


Figure 2: The two pipelines of ESRA: (a) Knowledge Graph Construction (section 3.1) and (b) Paper Searching for web application (section 3.2).

are SciIE (Luan et al., 2018) and SpERT (Eberts and Ulges, 2019). SciIE is a multi-task model that can perform named-entity recognition, relation extraction, and coreference resolution, whereas SpERT can only do the first two tasks but with better performance. Therefore, we combine the two models to be our extractor, using SciIE for coreference resolution and SpERT for entity and relation extraction, so as to achieve better performance across all the tasks.

Step 2: Post-processing The triples are then post-processed to clean duplicates and/or uninformative entities and relations to get the cleaned triples which form a local knowledge graph for each abstract. The post-processing includes (i) merging entities from the same coreference cluster, (ii) split entities with conjunction, (iii) converting plurals to singulars, (iv) relating abbreviations to the corresponding entities, (v) removing meaningless entities and relations and (vi) detecting conflicts against the knowledge graph ontology.

Step 3: Merging We insert the cleaned triples into the main knowledge graph and detect conflicts again to ensure that all comply with the ontology (e.g., no self-cycle or insensible relations). If the triple to be inserted already exists, its weight in the graph is then updated.

We use this pipeline to extract scientific knowledge from paper abstracts in the arXiv dataset

(Clement et al., 2019), particularly in the Computation and Language category (cs.CL). At the end, our knowledge graph contains 242k entities and 1.67M relations. It consists of eight entity types and eleven relation types, the statistics of which are displayed in Table 2 and 3, respectively. Most of the entity types (excluding Abbreviation, Author, and Paper) and relation types (excluding appear_in, cite, related_to, and refer_to) are adopted from the SciERC dataset (Luan et al., 2018).

Note that this pipeline is optimized for a scenario with AI-related texts because the extraction models were initially trained on the SciERC dataset containing only AI-related documents (Luan et al., 2018). To extend this pipeline to other domains, we need to use an extractor that can effectively recognize entities and relations tailored for those domains. For example, to work on the life science domain, we should use an extractor that recognizes concepts of *drugs* and *diseases* rather than *tasks* and *methods* (Ren et al., 2017).

3.2 Web Application: Search, Rank, Explain, and Visualize

As shown in Figure 2(b), after receiving an input query from a user, we perform query expansion by using entity names from our knowledge graph that are similar to the user query according to the similarity score given by sentence-BERT embeddings (Reimers and Gurevych, 2019). Then, the system

Query Keyword	Related Keywords
machine translation	phrase-based system, neural network
Abstract	
<p>[1] Neural machine translation is a recently proposed approach to machine translation. [2] Unlike the traditional statistical machine translation, the neural machine translation aims at building a single neural network that can be jointly tuned to maximize the translation performance. [3] The models proposed recently for neural machine translation often belong to a family of encoder-decoders and consists of an encoder that encodes a source sentence into a fixed-length vector from which a decoder generates a translation. [4] In this paper, we conjecture that (...). [5] With this new approach, we achieve a translation performance comparable to the existing state-of-the-art phrase-based system on the task of English-to-French translation. [6] Furthermore, qualitative analysis reveals that (...).</p>	
Filtered Abstract	
[1] [2] [2] [3] [5]	
Explanation	
<p>The neural machine translation aims at building a single neural network that can be jointly tuned to maximize the translation performance. The models proposed recently belong to a family of encoder-decoders and consists of an encoder that encodes a source sentence into a fixed-length vector.</p>	

Figure 3: The process of the explanation generation using the conditional text summarization technique. The query keywords are highlighted as yellow and the related keywords are highlighted as green. Some sentences without important keywords are ignored by replacing them with ‘(...)’.

passes the query to Elasticsearch⁹ for searching and ranking papers.

We retrieve the papers whose title or abstract contains an exact query, all of the keywords regardless of the orders, and some of the keywords. The results from each category will be sorted using a combination of (i) normalized Elasticsearch score and (ii) normalized citation count per day, before concatenated to be the final search results.

To provide a short explanation for why each paper is returned, we propose a technique called “Conditional text summarization”, as illustrated in Figure 3. We start by collecting the related keywords, i.e., the entities along the knowledge graph paths (of length 1 or 2) from the query to the paper. Then, to form the input of the summarization, the query and those keywords are used to select important sentences in the paper abstract with the sentences containing more than one keyword being repeated twice. After that, we use T5 (Xiong et al., 2017), a pre-trained sequence-to-sequence model, to summarize the filtered abstract to be the

⁹<https://www.elastic.co/elasticsearch/>

explanation. With this method, ESRA can generate different explanations for the same paper given different queries. For example, Table 4 shows the three different explanations for the BERT paper (Devlin et al., 2019) in response to the three queries – *BERT*, *Transformer*, and *SQuAD*.

For the fact list feature, we choose a group of facts from our knowledge graph that is connected to the user’s query nodes and show them along with the search results. In addition, ESRA provides visualizations of three subgraphs of our knowledge graph to the users. Firstly, the fact graph visualizes facts related to search keywords. In other words, it is the graphical view of the fact list. Secondly and thirdly, the paper graph and the keyword-to-paper graph visualize all nodes and relations that appear in the returned paper and relate the paper to the search keywords, respectively.

3.3 System Development

Users can interact with our platform, ESRA, via <http://esra.cp.eng.chula.ac.th/>. We developed the web application using React and Django frameworks for front-end and back-end services, respectively. The back-end also connects to (1) a knowledge graph manager which is responsible for searching and retrieving data from the graph database (Neo4j) and (2) a relational database (SQLite) that stores metadata. All the deep learning models used by ESRA are based on PyTorch.

4 Results and Evaluation

We evaluate ESRA in two ways. First, empirical evaluation concerns the effectiveness of knowledge graph extraction. Second, human evaluation targets the three main features of ESRA – explanation, fact list, and graph visualization.

4.1 Knowledge Graph Construction

According to section 3.1, our extractor combines SpERT (Eberts and Ulges, 2019) and SciIE (Luan et al., 2018) for achieving the three IE tasks in Table 5. Due to the lack of information extraction ground truth on the arXiv dataset, we decided to use the SciERC dataset (Luan et al., 2018) to evaluate the extractor instead. We compared our extractor to SpERT, SciIE, and DyGIE++ (Wadden et al., 2019). The results in Table 5 show that our extractor can retain the performance of SpERT on the first two tasks (entity and relation extraction), while it slightly sacrifices the performance of SciIE

Statistics	Quantity
#Method	57,762
#OtherScientificTerm	56,553
#Author*	34,449
#Task	34,365
#Material	27,766
#Paper*	23,111
#Metric	4,992
#Abbreviation**	3,066
Total	242,064

* Obtained from arXiv metadata

** Obtained during post-processing

Table 2: Entity statistics

Statistics	Quantity
#appear_in**	715,948
#cite*	438,161
#used_for	190,386
#author_of	172,388
#hyponym_of	40,800
#evaluate_for	38,110
#compare	15,366
#related_to**	15,136
#feature_of	14,970
#part_of	14,398
#refer_to**	11,472
Total	1,667,135

* Obtained from arXiv metadata

** Obtained during post-processing

Table 3: Relation statistics

for coreference resolution due to the difference between recognized named entities of both models (SpERT and SciIE).

4.2 Human Evaluation

We recruited 32 human participants who have been studying or working in the area of Computer Science and Engineering to evaluate ESRA. 14 out of the 32 participants identified that they specialize in NLP. Each participant was asked to evaluate the three main features of ESRA along three main dimensions – usefulness, understandability, and visual appeal – using a scale from 1 to 5 where the numbers mean strongly disappointed, disappointed, neutral, satisfied, and strongly satisfied, respectively. The results are reported in Table 6. The average score from all participants on each dimension falls within the range between 3.6 and 4.2, meaning that our system could reasonably sat-

BERT: A language representation model called BERT is designed to pre-train deep bidirectional representations from unlabeled text. The pre-trained model can be fine-tuned with just one additional output layer to create state-of-the-art models for a wide range of tasks.

Transformer: We introduce a new language representation model called BERT, which stands for Bidirectional Encoder Representations from Transformers.

SQuAD: It obtains new state-of-the-art results on eleven natural language processing tasks. It includes pushing the GLUE score to 80.5% (7.7% point absolute improvement), MultiNLI accuracy to 86.7% (4.6% absolute improvement) and SQuAD v1.1 question answering Test F1 to 93.2.

Table 4: Explanations for the BERT paper (Devlin et al., 2019) given three different queries: *BERT*, *Transformer*, and *SQuAD*.

Model	F1 (on SciERC)		
	NER	RE	CR
SciIE	64.20	39.30	48.20
DyGIE++	67.50	48.40	-
SpERT	70.33	50.84	-
SpERT+SciIE (Ours)	70.33	50.84	45.87

Table 5: Evaluation of knowledge extraction models on three tasks: named-entity recognition (NER), relation extraction (RE), and coreference resolution (CR).

isfy users with some room for further improvement. Apart from the satisfaction scores, we also collected users’ opinions on feature-specific questions and let them give us free-text comments where the results are discussed next.

Explanation: Overall, the participants responded that the generated explanations have an appropriate length (score 4.44 / 5) and they are easy to understand (4.25 / 5). Moreover, the explanations help the participants screen papers faster (4.22 / 5). However, the score for usefulness of this feature is relatively low (3.94 / 5) because usually the output from T5 is not much different from the abstract. We believe that adding more contents apart from the filtered abstract to the summarizer’s input would help mitigate this issue.

Fact list: The displayed facts are helpful for non-NLP-specialized users (4.07 / 5), probably because they can jump and explore related concepts in the list. However, NLP-specialized users gave a lower average score (3.78 / 5). Some comments suggested that the displayed facts are redundant. For example, “recall” and “recall value” should be

Dimension/Feature	Average score (1-5)		
	Expla- nation	Fact list	Graph viz.
Usefulness	3.94	3.91	3.88
Understandability	4.25	3.69	3.94
Visual appeal	3.81	3.81	3.81

Table 6: Human evaluation on the three main features

merged into one concept. This problem is a common weakness of automatic knowledge graph construction which could be alleviated by knowledge graph refinement (Paulheim, 2017).

Graph visualization: Some participants found that the graph visualization help them gather important points from the paper quickly such as evaluation metrics used in the paper. However, most of the comments noted that the graph is quite difficult to read, so they suggested the system show the full name of each graph node and adjust the layout for more readability.

5 Conclusion

Our literature discovery platform, ESRA, uses a scientific knowledge graph to enhance user’s experience. Based on the human evaluation, ESRA can help users screen through papers faster using the generated explanations and capture important facts about the query and the papers using the fact list and the graph visualization. In the future, we aim to expand the coverage of our knowledge graph by extracting facts from the full documents to enhance the quality of ESRA results.

References

Waleed Ammar, Dirk Groeneveld, Chandra Bhagavathula, Iz Beltagy, Miles Crawford, Doug Downey, Jason Dunkelberger, Ahmed Elgohary, Sergey Feldman, Vu Ha, Rodney Kinney, Sebastian Kohlmeier, Kyle Lo, Tyler Murray, Hsu-Han Ooi, Matthew Peters, Joanna Power, Sam Skjonsberg, Lucy Wang, Chris Wilhelm, Zheng Yuan, Madeleine van Zuylen, and Oren Etzioni. 2018. [Construction of the literature graph in semantic scholar](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 3 (Industry Papers)*, pages 84–91, New Orleans - Louisiana. Association for Computational Linguistics.

Isabelle Augenstein, Mrinal Das, Sebastian Riedel, Lakshmi Vikraman, and Andrew McCallum. 2017. [SemEval 2017 task 10: ScienceIE - extracting](#)

[keyphrases and relations from scientific publications](#). In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 546–555, Vancouver, Canada. Association for Computational Linguistics.

Colin B. Clement, Matthew Bierbaum, Kevin P. O’Keeffe, and Alexander A. Alemi. 2019. [On the use of arxiv as a dataset](#).

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

Markus Eberts and Adrian Ulges. 2019. [Span-based joint entity and relation extraction with transformer pre-training](#). *CoRR*, abs/1909.07755.

Patrick Ernst, Amy Siu, Dragan Milchevski, Johannes Hoffart, and Gerhard Weikum. 2016. [DeepLife: An entity-aware search, analytics and exploration platform for health and life sciences](#). In *Proceedings of ACL-2016 System Demonstrations*, pages 19–24, Berlin, Germany. Association for Computational Linguistics.

K. Gábor, Haïfa Zargayouna, I. Tellier, D. Buscaldi, and Thierry Charnois. 2016. [Unsupervised relation extraction in specialized corpora using sequence mining](#). In *IDA*.

Kata Gábor, Davide Buscaldi, Anne-Kathrin Schumann, Behrang QasemiZadeh, Haïfa Zargayouna, and Thierry Charnois. 2018. [SemEval-2018 task 7: Semantic relation extraction and classification in scientific papers](#). In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 679–688, New Orleans, Louisiana. Association for Computational Linguistics.

Mohamad Yaser Jaradeh, Allard Oelen, Kheir Eddine Farfar, Manuel Prinz, Jennifer D’Souza, Gábor Kismihók, Markus Stocker, and Sören Auer. 2019. [Open research knowledge graph: Next generation infrastructure for semantic scholarly knowledge](#). In *Proceedings of the 10th International Conference on Knowledge Capture, K-CAP ’19*, page 243–246, New York, NY, USA. Association for Computing Machinery.

M. Kas. 2011. [Structures and statistics of citation networks](#).

Samiya Khan, Xiufeng Liu, Kashish A Shakil, and Mansaf Alam. 2017. [A survey on scholarly data: From big data perspective](#). *Information Processing & Management*, 53(4):923–944.

- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.
- Yi Luan, Luheng He, Mari Ostendorf, and Hannaneh Hajishirzi. 2018. [Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3219–3232, Brussels, Belgium. Association for Computational Linguistics.
- Randall Munroe. 2013. [The rise of open access](#). *Science*, 342(6154):58–59.
- Nikolas Papanikolaou, G. Pavlopoulos, E. Pafilis, T. Theodosiou, Reinhard Schneider, V. Satagopam, C. Ouzounis, A. Eliopoulos, V. Promponas, and I. Iliopoulos. 2014. Biotextquest+: a knowledge integration platform for literature mining and concept discovery. *Bioinformatics*, 30 22:3249–56.
- Heiko Paulheim. 2017. Knowledge graph refinement: A survey of approaches and evaluation methods. *Semantic web*, 8(3):489–508.
- Nils Reimers and Iryna Gurevych. 2019. [Sentence-BERT: Sentence embeddings using Siamese BERT-networks](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.
- Xiang Ren, Jiaming Shen, Meng Qu, Xuan Wang, Zeqiu Wu, Qi Zhu, Meng Jiang, Fangbo Tao, Saurabh Sinha, David Liem, Peipei Ping, Richard Weinsilboum, and Jiawei Han. 2017. [Life-iNet: A structured network-based knowledge exploration and analytics system for life sciences](#). In *Proceedings of ACL 2017, System Demonstrations*, pages 55–60, Vancouver, Canada. Association for Computational Linguistics.
- J. Shen, Jinfeng Xiao, Yu Zhang, Carl Yang, Jingbo Shang, Jinda Han, Saurabh Sinha, P. Ping, R. Weinsilboum, Z. Lu, and Jiawei Han. 2018. Setsearch + : Entity-set-aware search and mining for scientific literature.
- Yanchuan Sim, Noah A. Smith, and David A. Smith. 2012. [Discovering factions in the computational linguistics community](#). In *Proceedings of the ACL-2012 Special Workshop on Rediscovering 50 Years of Discoveries*, pages 22–32, Jeju Island, Korea. Association for Computational Linguistics.
- Arnab Sinha, Zhihong Shen, Yang Song, Hao Ma, Darin Eide, Bo-June (Paul) Hsu, and Kuansan Wang. 2015. An overview of microsoft academic service (mas) and applications. In *Proceedings of the 24th International Conference on World Wide Web*, pages 243–246.
- Zhaowei Tan, Changfeng Liu, Yuning Mao, Yunqi Guo, Jiaming Shen, and Xinbing Wang. 2016. [Acemap: A novel approach towards displaying relationship among academic literatures](#). In *Proceedings of the 25th International Conference Companion on World Wide Web, WWW '16 Companion*, page 437–442, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.
- David Wadden, Ulme Wennberg, Yi Luan, and Hannaneh Hajishirzi. 2019. [Entity, relation, and event extraction with contextualized span representations](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5784–5789, Hong Kong, China. Association for Computational Linguistics.
- Kyle Williams, Jian Wu, Sagnik Ray Choudhury, Madian Khabsa, and C Lee Giles. 2014. Scholarly big data information extraction and integration in the citeseer χ digital library. In *2014 IEEE 30th international conference on data engineering workshops*, pages 68–73. IEEE.
- Chenyan Xiong, Russell Power, and Jamie Callan. 2017. [Explicit semantic ranking for academic search via knowledge graph embedding](#). In *WWW '17 Proceedings of the 26th International Conference on World Wide Web*, pages 1271–1279.