Knowledge Graph Induction Enabling Recommending and Trend Analysis: A Corporate Research Community Use Case

Nandana Mihindukulasooriya[(✉)], Mike Sava, Gaetano Rossiello,
Md. Faisal Mahbub Chowdhury, Irene Yachbes, Aditya Gidh,
Jillian Duckwitz, Kovit Nisar, Michael Santos, and Alfio Gliozzo

IBM Research AI, Yorktown Heights, NY, USA
nandana@ibm.com

Abstract. A research division plays an important role of driving innovation in an organization. Drawing insights, following trends, keeping abreast of new research, and formulating strategies are increasingly becoming more challenging for both researchers and executives as the amount of information grows in both velocity and volume. In this paper we present a use case of how a corporate research community, IBM Research, utilizes Semantic Web technologies to induce a unified Knowledge Graph from both structured and textual data obtained by integrating various applications used by the community related to research projects, academic papers, datasets, achievements and recognition. In order to make the Knowledge Graph more accessible to application developers, we identified a set of common patterns for exploiting the induced knowledge and exposed them as APIs. Those patterns were born out of user research which identified the most valuable use cases or user pain points to be alleviated. We outline two distinct scenarios: recommendation and analytics for business use. We will discuss these scenarios in detail and provide an empirical evaluation on entity recommendation specifically. The methodology used and the lessons learned from this work can be applied to other organizations facing similar challenges.

Keywords: Knowledge graph · Knowledge induction ·
Recommending · Trend analysis

1 Introduction

Research and innovation is the heart of any organization that is focused on advancing technologies to meet the challenges of solving real world problems by bridging the business needs with scientific discoveries. In fast moving research areas such as artificial intelligence or quantum computing, there is a tremendous growth of research activities in both velocity and volume happening within and

outside the organization [5, 37]. It is challenging to understand the trends and draw insights, and doing so manually is becoming unfeasible. Nevertheless, such insights are of utmost important for the executives who make strategy decisions on the impact of current investments and decide on future directions [36] and for the researchers who are looking for effective collaborations to optimize the reuse of research assets. In addition, in large organizations involving thousands of people and various scientific disciplines, it is difficult to keep abreast of individual projects. Weekly updates are often overwhelming but essential to make sure that people are informed of progress, to prevent redundant work, enhance re-usability, and cross fertilize ideas and assets. However, those has to be personalized to each person's user's interests to keep the information overload minimal.

One major challenge in generating insights is that generally data is scattered across different applications in their own siloed spaces. If integrated manually, this requires a lot of effort and hinders their full potential use for downstream applications. Thus, it is useful for an organization to have a unified integrated view of the data. Furthermore, these applications capture both structured metadata and also a lot of unstructured textual data. It's challenging to analyze the useful insights hidden in large volumes of text and uncover the insights.

For example, in the IBM research community, there are different applications for managing research projects, academic papers, datasets, internal achievements and external recognition. Researchers are both the content providers who contribute to these applications as well as end users that gets the recomendations and insights. From the adoption point of view, it is important that they have to spend only a minimum amount of valuable time without duplication of effort in multiple apps for the same information and get high value and useful insights in order to increase the engagement.

Before jumping to the solutions, we have first conducted a user study to understand the most valuable user pain points to be alleviated. Through a set of in-depth interviews from a set of selected users in different stages of their career, recommendations and trend analytics were identified as two main use cases that most requested by the community, as discussed in Sect. 2.

The aforementioned scenario provided us an excellent use case to test the boundaries of Knowledge Graph Induction (KGI) framework which is presented in this paper. Specifically, we apply our technology to mitigate some of the challenges in a corporate research community: IBM Research. While we restrict our focus to a research community in this paper, KGI framework can be applied to any organization that has a large volume of structured and unstructured data to be integrated and analyzed.

We will discuss how we address the common challenges of extraction of knowledge from both structured and unstructured data, how to enrich the KG from information available in the vast amounts of unstructured text and how to use the enriched KG to power Knowledge Exploitation Patterns (KEP) for entity recommendation and trend analytics. We will also discuss how the external encyclopedic knowledge such as Wikidata [38] can be seamlessly integrated to internal knowledge enabling traversal following the Linked Data principles to get more context or provide more structure to the data using the taxonomic knowledge.

The main contributions of this paper are as follows:

– We introduce an end-to-end framework for Knowledge Graph Induction from both structured, semi-structured, and unstructured data. KGI is easily portable across domains and enables the reuse of high level abstractions, i.e. KEP, for recommending and trend analysis.
– We introduce the Wikidata Parser, a Knowledge Generation and Linking approach based on transformer based generative models, which achieves the state of the art performances on information extraction benchmarks.
– We demonstrate the effectiveness of the KGI framework in two different scenarios: IBM research internal community and ISWC 2002–2021 proceedings.
– We discuss how a research organization can benefit from building a KG from both structured and unstructured data motivated by the pain points identified in a user study.

The rest of this paper is structured as follows. Section 2 discusses the use cases identified after an extensive user study. Section 3 introduces the KGI framework including knowledge integration, Wikidata parser and evaluate the knowledge generation using an academic benchmark. Section 4 introduces KEP for Entity Recommendation, Trend Analysis, and Infobox Generation, providing and empirical evaluation of the recommending capabilities based on user evaluation. Section 5 presents a review of related work, while Sect. 6 concludes the paper highlighting directions for future work.

2   Application Use Cases

The Apps@Research team, an application design and development team inside IBM Research, designs, develops, and supports a portfolio of cloud-based web applications providing rich, intuitive, integrated experiences that serve the unique needs of the IBM Research community. These include collaborative tools for:

– proposing and reporting progress on research projects including tracking staff effort, milestones, and impact (Research Project Portal)
– tracking the status of papers submitted to conferences and journal throughout the cycle from submission to decision (Academic Paper Portal)
– cataloging datasets approved for use by the legal team and datasets published by our teams (Dataset catalog)
– nominating, reviewing and selecting projects to receive yearly internal accomplishment awards (Achievements Portal)
– tracking external recognition and awards won by IBM researchers (Recognition Portal)

The Apps@Research team engaged the IBM Research AI team to partner on ways to incorporate IBM Research's own artificial intelligence technologies to augment the user experience in these applications. The key motivations were to:

– Unlock the content potential of the Apps@Research applications, which reflects the work and expertise across each division and teams.

– Improve user experience by creating exceptional, well-curated, concise and personalized information.
– Leverage and offer a testbed for IBM Research's own AI technology

In order to inform prioritization for the product roadmap for one of the most pervasive applications, we undertook a foundational user research study in 2020 to better understand user needs. The study included over 100 interviews and 220 survey responses from users of our applications. From this study, one key pain point was identified: because the content in our tools describe detailed research project proposals and plans of thousands of research projects, the content is too dense to be easily digestible. Users struggle in discovering relevant content and are under the perception that other users will find their content either. In turn, many users could become frustrated and stop using the tools for their key intended purposes - collaboration, innovation, and sharing updates.

Our hypothesis was that if we were to find a way to help the content become more discoverable, personalized, and digestible, that users would be motivated to keep their content up-to-date and visit the tool more frequently to find synergies and sparking innovation across research projects.

After doing some preliminary technical discovery and feasibility study with the AI Research team, we performed a more detailed user study. We recruited 12 participants from a representative sample of researcher and strategists at different stages in their career. They had varying experience with AI technology concepts. We conducted 60 min structured interview sessions with users in which we asked open-ended questions and then engaged them in an interactive exercise in a mural application.

The purpose of the interactive exercise was to identify various possible use cases and to prioritize them. We gave the users a hypothetical "$100" and asked them how they would "spend" the money, dividing among the use case ideas (Hundred dollar prioritization [20]). The purpose of the exercise was to understand the quantitative value that participants would ascribe to various use cases.

Upon completion of the interviews, we then performed a design thinking exercise called affinity mapping, to group ideas and identify common themes and patterns. We also analyzed the "$100 prioritization" to help quantify the value of use cases to all the participants.
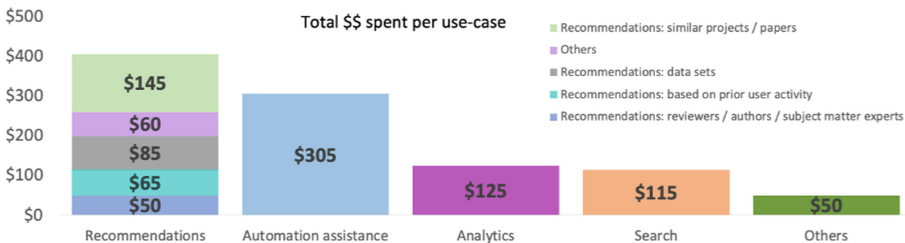

Fig. 1. User Interview $100 "spending" results

Figure 1 shows that most users identified "recommendations" as the most valuable use case. Recommendations would be automatically generated with information of which the user might otherwise not be aware. Recommendations would be personalized, based on users' previous activities such as papers/patents published, expertise, current projects, etc. - all of which could be derived from data in our KG. Figure 1 also shows that users wanted several types of recommendations such as related research projects, relevant papers, collaborators, experts to review their papers, etc.

The second most valuable use case would be "automation assistance". This would include help to pre-fill forms in the various tools in a smart way, saving the user time and anticipating their needs. This was a technical requirements and having an integrated view in KG would allow us to pre-fill a lot of information in different applications based on the context.

Next, users were interested in "analytics" - smart reports and dashboards that could be generated to provide business insights. The users have found that the data in our portals are dense and overwhelming and wanted to have high-level overview summaries so that can understand the common trends and dig more into the details.

Users were interested in improved Search and Filtering. Currently most of our applications' search is based on keywords and users were interested in more advanced semantic search capabilities. A KG would allow us to perform more complex structured searches.

Knowing that recommendations ranked highest as the most important use case, we analyzed further which types of content would be of greatest interest, so that we could prioritize developing those features first. We found that users ascribed the most value to being recommended projects and papers.

The insights gained from the user study led us to focus on the following two use case scenarios:

– **Recommending**: For researchers keeping abreast of colleagues' work (project status and publications) is very difficult in a large organization focusing on many technology areas. This is a hindrance to effective collaborations and reuse of research assets. There is a need for technologies and tools to make this process more seamless.
– **Trend Analysis**: For executives it is difficult to understand the breadth of the research portfolio, gain useful insights, and formulate a future strategy. There is a need to process large volumes of unstructured data and provide useful insights.

In the following sections, we will discuss our NLP and Semantic Web-driven approach for addressing these two main use case scenarios.

3   Knowledge Graph Induction

The overview of our KGI framework is illustrated in Fig. 2. It consists on three main conceptual blocks: data integration, whose main goal is to integrate heterogeneous semi-structured data from siloed applications using a domain ontology;
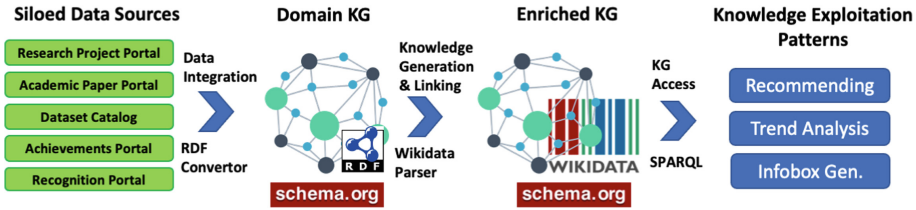
Fig. 2. The knowledge graph induction framework

knowledge extraction and linking, implemented by the *Wikidata Parser*, a component capable of generating RDF compliant knowledge by processing all textual content attached to entities in the domain KG; and *Knowledge Exploitation Patterns*, a set of abstractions over the induced KG that are domain-agnostic and generalized to use cases such as *recommending* and *analytics*.

3.1   Data Integration

Data related to IBM Research is scattered across multiple siloed applications. We used a knowledge representation approach based on Semantic Web standards and unified them into a single KG with links to both internal entities as well as relevant information extracted from background knowledge sources such as Wikidata.

Internal data pertains to items that are of particular interest to a research organization: research projects (science, strategy), people (eminence), academic publications and datasets (eminence), achievements and recognition (impact).

Each of the applications provides an API to extract data, which is then processed through a RDF conversion pipeline following a process similar to RML-based tools [10]. For this purpose, a Research KG ontology was built by reusing and extending the Schema.org with classes and relations that were more specific to our use case. The data schema of each of the five applications were aligned to the ontology by a knowledge engineer and the mappings were created.

The Schema.org ontology was selected as the base because it covered most of the concepts in our applications and is used by some of our collaborators. In addition, entities and relations from Wikidata are also reused. This enables us to easily integrate with third parties. The conversion process consists of (a) data extraction and (b) cleaning to normalize certain values, (c) mapping and RDF generation. Entity resolution is carried out to convert mentions to people, projects, and other entities to their canonical identifiers through a deterministic process. To this aim, we used unique identifiers such as emails and other internal conventions.
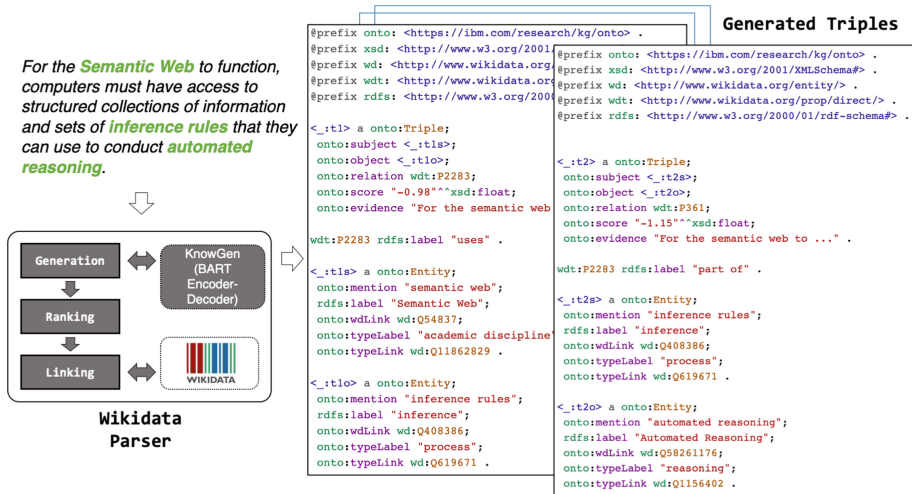
Fig. 3. The Wikidata Parser architecture and an example output.

3.2 Knowledge Generation and Linking: Wikidata Parser

A large part of our data is unstructured text. In order to incorporate them in the KG, we extract the textual values attached to each entity such as article content or a project description, chunk them into sentences and parse them using *Wikidata Parser*, a novel tool integrated in our KGI framework [9,13].

Wikidata Parser allows converting unstructured text into structured data represented as a set of ABox assertions compliant with the TBox of Wikidata. We address this problem as a sequence generation task, similar to machine translation or text summarization, where the input is an English sentence and the output is a set of facts. To this aim we leveraged large pre-trained sequence-to-sequence language models, such as BART [23] and train them from large dataset derived using distant supervision, by exploiting the alignments between Wikidata facts with the abstracts of Wikipedia pages.

Specifically, given a sentence, we fine-tune the language model to detect pairs of entity mentions and jointly generate a set of facts (i.e. <Subject (Subject Type), relation, Object (Object Type)>) representing entity labels, entity types and their relationships. The output of the system is then deterministically converted in RDF statements, as shown in Fig. 3.

Our experiments and analysis show that Wikidata Parser produces more accurate triples improving in both precision and recall if compared with the state-of-the-art generative information extraction methods [6,19,32].

Table 1 reports the F1 results of Wikidata Parser for each type of semantic annotations part of the triples generated from the abstracts, in terms of correct predictions of entity mentions, entity labels, entity types and their joint relations. For training and evaluation purposes, we extended a distantly supervised dataset

for relation extraction [6] with the full set of Wikidata-based annotations for each matched triple found in the abstracts of Wikipedia.

Table 1. Information extraction results. **MD** = Mention Detection. **TYPE** = Type Prediction. **EL** = Entity Label. **RN** = Relation Name. **REL** = Relation Prediction with Label Match. **P** = Precision. **R** = Recall. **F1** = Micro F1-score.

	MD-F1	TYPE-F1	EL-F1	RN-F1	REL-P	REL-R	REL-F1
Approach							
SOTA IE pipeline [19]	–	–	–	–	43.30	41.73	42.50
GenIE [19]	-	-	79.69	78.21	68.02	**69.87**	68.93
Wikidata Parser	84.27	79.65	**82.73**	**80.84**	**73.88**	67.85	**70.74**

For both subject and object, we generate the surface form mention, canonical label, type label, relation label. Whenever applicable, we link the entities and types to Wikidata entities. Relations are also linked to Wikidata. This information is then converted in RDF and represented using a reified statement meta model. In addition, the facts are associated to an evidence attribute, which contains the provenance (i.e. the sentence) from which the triple has been generated together with its confidence score. An example output is shown in Fig. 3. In addition, each triple is linked to the corresponding entity where the text was extracted.

3.3   Implementation Details

The KG implementation consists of several components. First and foremost is the actual deployment and hosting of the knowledge graph. Our knowledge graph is hosted on a Blazegraph triplestore inside a RedHat OpenShift Container platform which gives us all the advantages of a cloud deployment (scaling, flexibility, storage). We have a second component, a reverse proxy for Single Sign-On (SSO) authentication and authorization to the graph. Some of the data in our graph is confidential and therefore requires a need to know access to prevent traversing and querying the graph by unintended parties. The final set of components relate to the ETL (Extraction, Transform, and Load) process. Currently we build and load the graph on weekly basis. Our ETL process consists of extracting the data from all of the application APIs (both GraphQL and REST) as JSON documents, keeping an in-memory representation of the documents, and then converting these documents to RDF in Turtle format. The textual raw data of each entity is enriched with Wikidata parser as described in Sect. 3.2 with automated OpenShift cronjobs. Finally, RDF data coming from both structured and textual sources is integrated and loaded into the triple store on a scheduled basis.

The current ETL process will be vastly be improved in the future to address the evolution of data by limiting text processing only to detected changes in the KG. Some of this future work will require including a text fingerprinting service

to decide if the data has indeed changed (*i.e.* for computational cost, we only care about the free text changes and not usually the meta data).

4 Knowledge Exploitation Patterns

To make the KG easy to use and adapt across different domains, we identified a set of common usage patterns, *Knowledge Exploitation Patterns (KEP)*, and expose them as parameterized client API library to minimize the learning curve for the technology. These APIs generate the corresponding SPARQL queries and handle other cross-cutting concerns such as security or caching. Nevertheless, developers also can run queries directly in the SPARQL endpoint if needed. Currently, we provide APIs required for induced ontology exploration (type hierarchies, infoboxes), entity recommendation, and trend analysis. The idea behind the use of KEP is that certain functionalities can be abstracted out of the specific application domain by performing queries against the KG metamodel that is then used differently in downstream applications for the specific domain.

4.1 Entity Recommendations

Based on our use cases study described in Sect. 2, the automatic recommendations of items, such as publications, projects or collaborators, is one of the main desiderata for the members of our enterprise research community. Collaborative filtering [22] is arguably the most common approach for recommendation systems, especially in environments with a large user base where the state-of-the-art methods are based on advanced deep learning techniques. However, an enterprise research community might not have enough users to train large parametric models due to the sparsity of user log activities. For this reason, we adopt a hybrid content-based recommendation system method [12,24] by exploiting jointly the textual content, structured data and induced semantic annotations generated from our Wikidata Parser (see Sect. 3.2).

The idea is to convert our KG in an entity-feature Vector Space Model (VSM) model, where the rows are represented by the different type of entities in the KG, such as people, publications, projects and accomplishments, and the columns represent the feature space. In detail, let us consider $VSM^{n,m}$ a matrix using the standard tf-idf weighting schema, where each row $e_{i,*}$ is an entity vector created by concatenating different groups of features, described as follow:

Bag of Words. The textual content of entities, such as publications or projects, are tokenized and each token is considered as a single (sparse) feature. For entities representing people, where the textual context is not available, we exploit our KG to collect the textual content, e.g., from the publications or projects linked to the specific user by a multi-hop navigation in the graph.

Structured Data. This feature set represents relations derived from knowledge integration from our original data sources. For instance, the research division and topic of a project, the upper-line management for a person, and so on.

Entities. This feature set represent the entities extracted from Wikidata parser, grouped by their Wikidata type. For example, given the triples in Fig. 3, we create entity features such as SEMANTIC WEB:ACADEMIC DISCIPLINE, INFERENCE:PROCESS, and so on.

Frames. We also leverage the semantic relational information from the extracted triples. In order to alleviate the sparsity problem, we only concatenate the semantic annotations w.r.t. the domain, relation and range of each triple. For instance, <ACADEMIC DISCIPLINE, USES, PROCESS> for one of the generated triple in Fig. 3.

It is important to note that our feature set does not depend on the specific entity and relation set. Instead, this pattern is totally domain-agnostic and reusable and can be applied to any KG and entity type generated from our KG induction pipeline and integration process.

After the VSM is built, the recommendation inference for a user is implemented in a non-parametric manner by exploiting the cosine similarity between the user and the target item vectors, such as publications, projects or other users. In other words, the recommended items for a user are the nearest neighbor entities in the vector space ranked by their cosine similarity scores.
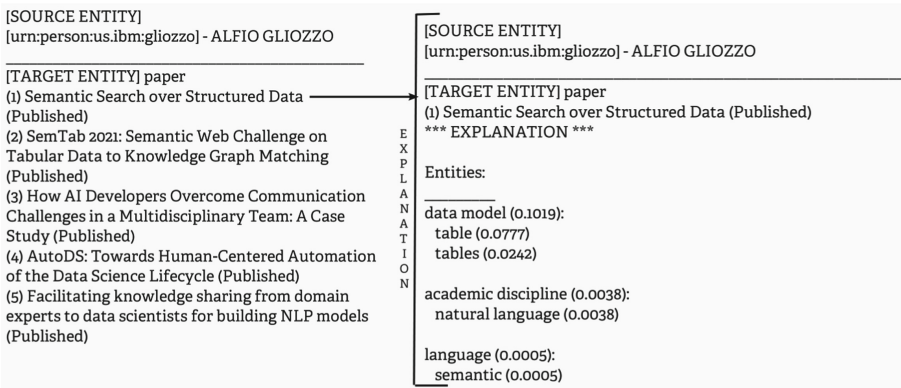
Fig. 4. An example of paper recommendations for a researcher. The figure on the left reports the list of recommended publications. The explanation for the top ranked item is shown in the figure on the right as a list of relevant entities grouped by their semantic types.

Figure 4 shows an example of a list of recommended publications for an researcher using the aforementioned KG-based VSM. The KG induced from text allows us to provide meaningful explanations for the user that justify the recommendation. The explanation is obtained by measuring and selecting the most relevant entities (i.e. those that contributed most to the similarity score), ranked by their combined tf-idf weights.

Table 2. User evaluation for scholary article, project and achievement recommendations for 30 users.

	Papers		Projects		Achievements	
Criteria	MAP	P @ 10	MAP	P @ 10	MAP	P @ 5
LOW	0.89	0.76	0.92	0.81	0.87	0.82
MEDIUM	0.51	0.34	0.65	0.45	0.51	0.36
HIGH	0.21	0.08	0.50	0.14	0.41	0.17

To evaluate the quality of the recommendations, we recruited 30 volunteer researchers from various disciplines. For each participant, we recommended 10 projects, 10 papers and 5 achievements. Each participant was asked to rate the recommendations on the following scale:

– NONE: No value to me
– LOW: Good to know but I am not going to read anytime soon
– MEDIUM: Relevant for my specific area of interest (must read)
– HIGH: Relevant to my current project(s) and work

We performed a quantitative analysis by evaluating Mean Average Precision (MAP) and Precision@K (P@K) metrics, which are popular choices to evaluate recommendation systems. Both MAP and P@K take in consideration only binary assessments, i.e. if the recommended item is relevant or non-relevant. In order to convert our graded rating into a binary assessment, we adopt three different criteria, namely HIGH (i.e. only HIGH category is regarded as positive), MEDIUM (i.e. HIGH and MED categories are positive), LOW (i.e. HIGH, MED and LOW are positive). As shown in Table 2, the performance of our recommendation system is consistent across the different type of recommended items. Moreover, the MAP is consistently higher than P@K, showing that the system tends to provide higher scores to those items considered relevant for the users.

We also performed an analysis focusing on irrelevant recommendations. One repeating pattern was the users who have recently moved to a different research area tends to have less accurate recommendation. This is can be explained by observing that their historical publication profile did not reflect their current information needs. Another commonly reported problem is that in many cases the researchers were aware of the recommended items already, in spite of the fact that we filtered out those items were they were explicitly listed as authors or contributors. The explanation for that is that there could be multiple relations between a person and an information object, besides being `authorOf`. For example, one researcher might have been the mentor of one of those authors, might have been part of a review committee and so on. In future work, we planned to address the above issues by applying more sophisticated machine learning-based recommendation techniques able to learn how to traverse the graph structure from the user provided feedback.

4.2   Trend Analysis

The KG induced from the unstructured text is used to implement KEPs for trend analysis. Once a corpus is completely processed by Wikidata parser, trend analytics provide an overview of the concepts found in the corpus simply by performing aggregation queries the induced KG.

Since we can not show examples of analytic from the IBM internal data due to privacy of strategic information, we have created a KG by processing ISWC papers from year 2002–2021 using DBLP RDF data[1]. For each paper, we collected the title and the abstract of the paper and parsed them using the Wikidata Parser to create an Induced KG. Examples in this section uses that KG. This also provides evidence that the approach that we have proposed can be easily adapted to other communities.

Figure 5 shows the most frequent types found in the ISWC 2002–2021 corpus with the number of unique entities found in the corpus and number of associated triples. Any type can be selected and expanded to see its subtypes in the corpus ordered by their cumulative frequency (direct children and all descendants). Figure 5 (right) illustrates the expansion of type `algorithm` which has 473 direct entities and 746 transitive entities. The subclass relations are both induced from text and extracted from Wikidata. Users can select any of 4739 types generated in the case of the ISWC corpus and generate a trend analysis for the given type.

Induced Types in the KG (Top 15 out of 4739)

	Type	# of Entities	# of Triples	Wikidata Link
1	software	1127	3920	http://www.wikidata.org/entity/Q7397
2	academic discipline	936	20313	http://www.wikidata.org/entity/Q11862829
3	free software	879	3941	http://www.wikidata.org/entity/Q341
4	concept	707	2702	http://www.wikidata.org/entity/Q151885
5	data structure	547	1929	http://www.wikidata.org/entity/Q175263
6	ontology	502	1448	http://www.wikidata.org/entity/Q44325
7	algorithm	473	1516	http://www.wikidata.org/entity/Q8366
8	software feature	472	1625	http://www.wikidata.org/entity/Q4485156
9	process	462	2025	http://www.wikidata.org/entity/Q619671
10	programming language	438	2572	http://www.wikidata.org/entity/Q9143
11	file format	432	5417	http://www.wikidata.org/entity/Q235557
12	computer science term	429	1447	http://www.wikidata.org/entity/Q66747126
13	website	423	1868	http://www.wikidata.org/entity/Q35127
14	data set	370	1857	http://www.wikidata.org/entity/Q1172284
15	database	311	1896	http://www.wikidata.org/entity/Q8513

```
Type hierarchy for algorithm
algorithm[Q8366] (746)
+- artificial intelligence[Q11660] (253)
|   +- machine learning[Q2539] (212)
|   |   +- artificial neural network[Q192776] (26)
|   |   |   +- feedforward neural network[Q5441227] (7)
|   |   |   |   +- convolutional neural network[Q17084460] (2)
|   |   |   +- recurrent neural network[Q1457734] (1)
|   |   +- statistical classification[Q1744628] (20)
|   |   +- supervised learning[Q334384] (5)
|   |   +- deep learning[Q197536] (3)
|   |   +- multi-task learning[Q6934509] (1)
|   |   +- unsupervised learning[Q1152135] (1)
|   +- natural language processing[Q30642] (22)
|   |   +- speech recognition[Q189436] (2)
|   |   +- natural language generation[Q1513879] (1)
|   +- heuristic[Q1981968] (4)
|   +- sentiment analysis[Q2271421] (4)
|   +- knowledge engineering[Q1540472] (1)
|   +- cognitive science[Q147638] (1)
+- search algorithm[Q755673] (102)
+- combinatorial algorithm[Q41883552] (32)
|   +- graph algorithm[Q30503704] (9)
|   +- sorting algorithm[Q181593] (5)
+- optimization algorithm[Q2835765] (20)
|   +- evolutionary algorithm[Q14489129] (1)
+- cipher[Q4681865] (14)
|   +- block cipher[Q543151] (10)
+- parallel algorithm[Q10879871] (13)
```
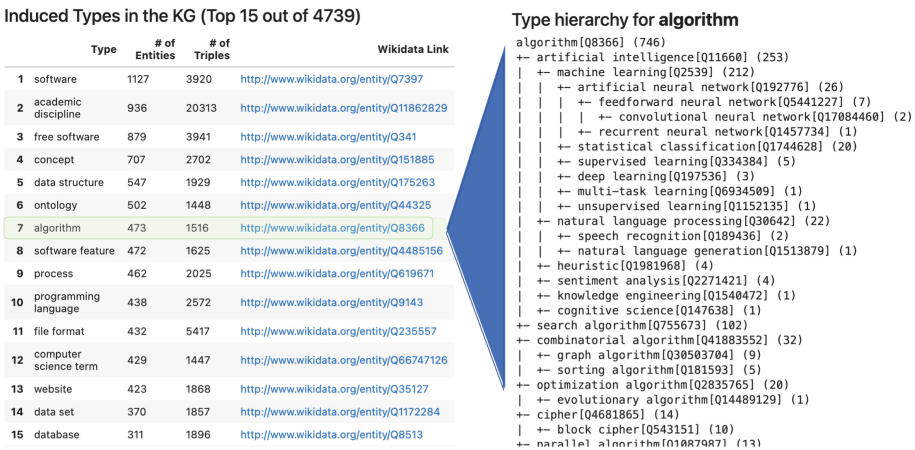
Fig. 5. A snippet from induced types from the ISWC corpus.

Figure 6 shows the trend analysis for entities belonging to the type `academic discipline`. The last column shows the total number of occurrences of each entity in all ISWC papers from 2002–2021. Individual cells show the distribution of the papers in different years as a percentage. Such trend analysis can highlight

some interesting facts. For instance, it shows that there was a high interest in "*Ontology*" and "Semantic Web" throughout from the beginning but the interest diversify more in later years. Similarly, we can see that there is a high interest in "*Linked Data*" from year 2009 which is at highest during the 2013–2017 period. In contrast, "*Semantic Web Services*" are of high interest during 2003–2009 period but the interest completely vanishes on later years. It is important to notice that the list of entities belonging to the type `academic discipline` or any other type is automatically generated. The analyst is just supposed to point to the right concept in the taxonomy to get her job done.

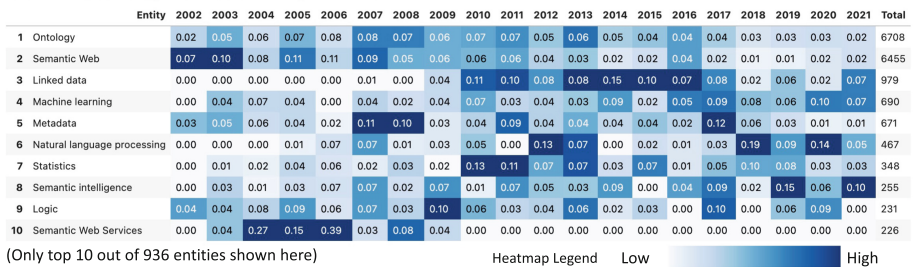**Trend analysis for type 'academic discipline' in from KG**

	Entity	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021	Total
1	Ontology	0.02	0.05	0.06	0.07	0.08	0.08	0.07	0.06	0.07	0.07	0.05	0.06	0.05	0.04	0.04	0.04	0.03	0.03	0.03	0.02	6708
2	Semantic Web	0.07	0.10	0.08	0.11	0.11	0.09	0.05	0.06	0.06	0.06	0.04	0.03	0.02	0.02	0.04	0.02	0.01	0.01	0.02	0.02	6455
3	Linked data	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.04	0.11	0.10	0.08	0.08	0.15	0.10	0.07	0.08	0.02	0.06	0.02	0.07	979
4	Machine learning	0.00	0.04	0.07	0.04	0.00	0.04	0.02	0.04	0.07	0.03	0.04	0.03	0.09	0.02	0.05	0.09	0.08	0.06	0.10	0.07	690
5	Metadata	0.03	0.05	0.06	0.04	0.02	0.11	0.10	0.03	0.04	0.09	0.04	0.04	0.04	0.04	0.02	0.12	0.06	0.03	0.01	0.01	671
6	Natural language processing	0.00	0.00	0.00	0.01	0.07	0.07	0.01	0.03	0.05	0.00	0.13	0.07	0.00	0.02	0.01	0.03	0.19	0.09	0.14	0.05	467
7	Statistics	0.00	0.01	0.02	0.04	0.06	0.02	0.03	0.02	0.13	0.11	0.07	0.07	0.03	0.07	0.01	0.05	0.10	0.08	0.03	0.03	348
8	Semantic intelligence	0.00	0.03	0.01	0.03	0.07	0.07	0.02	0.07	0.01	0.07	0.05	0.03	0.09	0.00	0.04	0.09	0.02	0.15	0.06	0.10	255
9	Logic	0.04	0.04	0.08	0.09	0.06	0.07	0.03	0.10	0.06	0.03	0.04	0.06	0.02	0.03	0.00	0.10	0.00	0.06	0.09	0.00	231
10	Semantic Web Services	0.00	0.04	0.27	0.15	0.39	0.03	0.08	0.04	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	226

(Only top 10 out of 936 entities shown here)   Heatmap Legend   Low ▢▢▢▢ High

Fig. 6. An example of trends analytics for entities of type academic discipline

4.3   Infobox Generation

Once an entity of interest is selected, for example, "*Linked Data*", the users can automatically generate an infobox, as shown in Fig. 7. We first induce a schema for each type, by counting the most frequent relations extracted by the parser for entities of that type. For example, for the type `academic discipline` the important relations are `part of`, `facet of`, `based on`, `studies` and so on. Then we collect the object filling those relations for a specific target entity (*Linked Data*, in the example). Those relations might come from induced triples or from Wikidata itself. Each of the relations in the infobox is also associated to its provenance (might be a textual occurrence or a pre-existing triple in Wikidata) as illustrated by Fig. 8.

5   Related Work

KGs are a common way to organize data from multiple sources providing a unified view and represent them in a semantically rich manner empowering a wide range of downstream applications [15,18,29]. More specifically, Scholarly KGs such as ORKG [16], MAG [39], OpenAIRE [25] are becoming popular way to represent research data. Such KGs are used for search [4,14], question answering [17], recommendation [26,27], analysis of research trends [36], performing

Infobox for **Linked Data** (academic discipline) - **Q515701**

WIKIDATA RELATION	VALUES INDUCED FROM TEXT	EXTRACTED FROM WIKIDATA
part of	Open Data, Semantic Web, Semantic metadata, Web of data, World Wide Web, +7 more	
facet of	metadata, Semantic technology, Ontology	
based on	Resource Description Framework, XML	
studies	Ontology, Graph, Information, Artificial intelligence	
has quality	Accessibility, Data integration, Context-aware web	
used by	E-commerce, Federal government	Semantic Web
designed by		World Wide Web Consortium
official website		http://linkeddata.org/
described at URL		https://www.w3.org/wiki/LinkedData

Fig. 7. An example of infobox for "*Linked Data*" an entity including both induced facts and integrated Wikidata facts.

Evidences for the fact: <Linked Data, part of, Open Data>

paper	sentence	score
https://doi.org/10.1007/978-3-030-30796-7_27	"An Assessment of Adoption and Quality of Linked Data in European Open Government Data."	1.0
https://doi.org/10.1007/978-3-319-25010-6_4	"Collecting, Integrating, Enriching and Republishing Open City Data as Linked Data."	0.1

Fig. 8. Evidences for the induced fact (Linked Data, part of, Open Data)

surveys [30], and understanding the dynamics between academia and industry [3].

The Semantic Web community has developed several methods and tools for building KGs. There are comprehensive survey articles on building KGs from relational databases [35], semi-structued data [33], and unstructured text [1,8, 11,28,34]. Rezayi et al. [31] propose an approach to augment a KG with key phrases generated from textual content of entities. In our work, we augment our KG with semantically rich triples generated from textual content of each entity. Furthermore, we integrate the induced knowledge with the relevant portion of background knowledge from Wikidata.

Trend analysis on KGs has been used for analysing research topics [21,36,41], patents [40], market trends [2]. Wikidata parser presented in our approach allows automatically create an induced knowledge graphs from text with a large number of Wikidata types (50K in 2022) enabling fine-grain analysis and seamless integration of background knowledge from Wikidata that can be used in the analysis.

Cai et al. [7] proposes an explainable recommender by generating the candidates using a KG and using an evolutionary algorithm. We use a simpler vector space model to produce recommendations between different types of entities.

6 Conclusions and Future Work

In this work, we presented an application of the Knowledge Graph Induction (KGI) technology to fulfill the requirements identified by a user study to enhancing cooperation in a research community. We have shown how the induced knowledge enables several downstream applications, such as recommending and trend analytics, providing evaluation for most of the component based on both quantitative and qualitative approaches. This year, we intend to deploy the recommending technology to all the member of the IBM research community, in the order of 6,000 people. We envision both in-app and "meet users where they are" experiences outside the apps. In all cases, we will provide feedback mechanisms (e.g. thumbs up/down, free text explanations) for users to share their view on the quality of the recommendations. The intention is to feed this back into a deep learning based recommender to learn how to better exploit the graph traversals.

In addition to trend analysis, we believe that KGI technology could also be leveraged for flexible and on-demand business analytics, providing powerful insights to accelerate business, for example:

Predicting Success. What are the characteristics of research projects that result in recognition and awards. How do we invest in new projects that exhibit these characteristics to better steer the IBM research agenda? Which papers should we support to have the best chance of publication at key conferences?

Business Development. Quickly identifying relevant research activity of interest to current or prospective clients or partners

Operations and Efficiency. Who is working on what projects and is time being used effectively? Is there duplicate activity? Where are the gaps? What are best opportunities for cross-collaboration?

Talent. Who are the rising stars? How do we find the right projects for them, or nominate them for external awards?

Portfolio. Tracing research projects and outcomes to Objects and Key Results.

We plan to develop KEP for the use cases above that can be generalized beyond the research community use case. We believe that the KEPs can be designed to cover variety of different use cases in many different organizations.

Moreover, we are planning to acquire KGs from different research communities (e.g. Semantic Web, NLP, Deep Learning communities) and make them available to the community. The goal is to act as a catalyzer for future research work in the research community beyond IBM.

References

1. Al-Aswadi, F.N., Chan, H.Y., Gan, K.H.: Automatic ontology construction from text: a review from shallow to deep learning trend. Artif. Intell. Rev. **53**(6), 3901–3928 (2019). https://doi.org/10.1007/s10462-019-09782-9

2. Albrecht, J., Belger, A., Blum, R., Zimmermann, R.: Business analytics on knowledge graphs for market trend analysis. In: LWDA, pp. 371–376 (2019)

3. Angioni, S., Salatino, A.A., Osborne, F., Recupero, D.R., Motta, E.: Integrating knowledge graphs for analysing academia and industry dynamics. In: Bellatreche, L., et al. (eds.) TPDL/ADBIS -2020. CCIS, vol. 1260, pp. 219–225. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-55814-7_18

4. Heidari, G., Ramadan, A., Stocker, M., Auer, S.: Leveraging a federation of knowledge graphs to improve faceted search in digital libraries. In: Berget, G., Hall, M.M., Brenn, D., Kumpulainen, S. (eds.) TPDL 2021. LNCS, vol. 12866, pp. 141–152. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-86324-1_18

5. Bornmann, L., Mutz, R.: Growth rates of modern science: a bibliometric analysis based on the number of publications and cited references. J. Am. Soc. Inf. Sci. **66**(11), 2215–2222 (2015)

6. Cabot, P.H., Navigli, R.: REBEL: relation extraction by end-to-end language generation. In: EMNLP (Findings), pp. 2370–2381. Association for Computational Linguistics (2021)

7. Cai, X., Xie, L., Tian, R., Cui, Z.: Explicable recommendation based on knowledge graph. Expert Syst. Appl. **15**, 117035 (2022)

8. Carlson, A., Betteridge, J., Kisiel, B., Settles, B., Hruschka Jr., E.R., Mitchell, T.M.: Toward an architecture for never-ending language learning. In: AAAI. AAAI Press (2010)

9. Chowdhury, M.F.M., Glass, M.R., Rossiello, G., Gliozzo, A., Mihindukulasooriya, N.: KGI: an integrated framework for knowledge intensive language tasks. CoRR abs/2204.03985 (2022)

10. Dimou, A., Vander Sande, M., Colpaert, P., Verborgh, R., Mannens, E., Van de Walle, R.: RML: a generic language for integrated RDF mappings of heterogeneous data. In: LDOW (2014)

11. Dong, X., et al.: Knowledge vault: a web-scale approach to probabilistic knowledge fusion. In: KDD, pp. 601–610. ACM (2014)

12. de Gemmis, M., Lops, P., Musto, C., Narducci, F., Semeraro, G.: Semantics-Aware content-based recommender systems. In: Ricci, F., Rokach, L., Shapira, B. (eds.) Recommender Systems Handbook, pp. 119–159. Springer, Boston, MA (2015). https://doi.org/10.1007/978-1-4899-7637-6_4

13. Glass, M.R., Rossiello, G., Chowdhury, M.F.M., Gliozzo, A.: Robust retrieval augmented generation for zero-shot slot filling. In: EMNLP (1), pp. 1939–1949. Association for Computational Linguistics (2021)

14. Heidari, G., Ramadan, A., Stocker, M., Auer, S.: Demonstration of faceted search on scholarly knowledge graphs. In: Companion Proceedings of the Web Conference 2021, pp. 685–686 (2021)

15. Hogan, A., et al.: Knowledge graphs. Synth. Lect. Data Semant. Knowl. **12**(2), 1–257 (2021)

16. Jaradeh, M.Y., et al.: Open research knowledge graph: next generation infrastructure for semantic scholarly knowledge. In: Proceedings of the 10th International Conference on Knowledge Capture, pp. 243–246 (2019)

17. Jaradeh, M.Y., Stocker, M., Auer, S.: Question answering on scholarly knowledge graphs. In: Hall, M., Merčun, T., Risse, T., Duchateau, F. (eds.) TPDL 2020. LNCS, vol. 12246, pp. 19–32. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-54956-5_2

18. Ji, S., Pan, S., Cambria, E., Marttinen, P., Philip, S.Y.: A survey on knowledge graphs: representation, acquisition, and applications. IEEE Trans. Neural Netw. Learn. Syst. **33**(2), 494–514 (2021)

19. Josifoski, M., Cao, N.D., Peyrard, M., West, R.: Genie: generative information extraction. CoRR abs/2112.08340 (2021)

20. Khan, J.A., Rehman, I.U., Khan, Y.H., Khan, I.J., Rashid, S.: Comparison of requirement prioritization techniques to find best prioritization technique. Int. J. Mod. Educ. Comput. Sci. **7**(11), 53–59 (2015)

21. Kim, Y., Ju, Y., Hong, S., Jeong, S.R.: Practical text mining for trend analysis: ontology to visualization in aerospace technology. KSII Trans. Internet Inf. Syst. (TIIS) **11**(8), 4133–4145 (2017)

22. Koren, Y., Bell, R.: Advances in collaborative filtering. In: Ricci, F., Rokach, L., Shapira, B. (eds.) Recommender Systems Handbook, pp. 77–118. Springer, Boston (2015). https://doi.org/10.1007/978-1-4899-7637-6_3

23. Lewis, M., et al.: BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In: ACL, pp. 7871–7880. Association for Computational Linguistics (2020)

24. Liu, J., Duan, L.: A survey on knowledge graph-based recommender systems. In: 2021 IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), vol. 5, pp. 2450–2453. IEEE (2021)

25. Manghi, P., Houssos, N., Mikulicic, M., Jörg, B.: The data model of the OpenAIRE scientific communication e-Infrastructure. In: Dodero, J.M., Palomo-Duarte, M., Karampiperis, P. (eds.) MTSR 2012. CCIS, vol. 343, pp. 168–180. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-35233-1_18

26. Manrique, R., Marino, O.: Knowledge graph-based weighting strategies for a scholarly paper recommendation scenario. In: KaRS@ RecSys, pp. 5–8 (2018)

27. Nayyeri, M., Vahdati, S., Zhou, X., Shariat Yazdi, H., Lehmann, J.: Embedding-based recommendations on scholarly knowledge graphs. In: Harth, A., et al. (eds.) ESWC 2020. LNCS, vol. 12123, pp. 255–270. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-49461-2_15

28. Niu, F., Zhang, C., Ré, C., Shavlik, J.W.: DeepDive: web-scale knowledge-base construction using statistical learning and inference. In: VLDS. CEUR Workshop Proceedings, vol. 884, pp. 25–28. CEUR-WS.org (2012)

29. Noy, N., Gao, Y., Jain, A., Narayanan, A., Patterson, A., Taylor, J.: Industry-scale knowledge graphs: lessons and challenges. Commun. ACM **62**(8), 36–43 (2019)

30. Oelen, A., Jaradeh, M.Y., Stocker, M., Auer, S.: Generate fair literature surveys with scholarly knowledge graphs. In: Proceedings of the ACM/IEEE Joint Conference on Digital Libraries in 2020, pp. 97–106 (2020)

31. Rezayi, S., Zhao, H., Kim, S., Rossi, R., Lipka, N., Li, S.: Edge: enriching knowledge graph embeddings with external text. In: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL), pp. 2767–2776 (2021)

32. Rossiello, G., et al.: Generative relation linking for question answering over knowledge bases. In: Hotho, A., et al. (eds.) ISWC 2021. LNCS, vol. 12922, pp. 321–337. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-88361-4_19

33. Ryen, V., Soylu, A., Roman, D.: Building semantic knowledge graphs from (semi-) structured data: a review. Future Internet **14**(5), 129 (2022)

34. de Sá Mesquita, F., Cannaviccio, M., Schmidek, J., Mirza, P., Barbosa, D.: KnowledgeNet: a benchmark dataset for knowledge base population. In: EMNLP/IJCNLP (1), pp. 749–758. Association for Computational Linguistics (2019)
35. Sahoo, S.S., et al.: A survey of current approaches for mapping of relational databases to RDF. W3C RDB2RDF Incubator Group Rep. **1**, 113–130 (2009)
36. Salatino, A.A., Mannocci, A., Osborne, F.: Detection, analysis, and prediction of research topics with scientific knowledge graphs. In: Predicting the Dynamics of Research Impact, pp. 225–252. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-86668-6_11
37. Savage, N.: The race to the top among the world's leaders in artificial intelligence. Nature **588**(7837), S102–S102 (2020)
38. Vrandečić, D., Krötzsch, M.: WikiData: a free collaborative knowledgebase. Commun. ACM **57**(10), 78–85 (2014)
39. Wang, K., Shen, Z., Huang, C., Wu, C.H., Dong, Y., Kanakia, A.: Microsoft academic graph: when experts are not enough. Quant. Sci. Stud. **1**(1), 396–413 (2020)
40. Weber, L., Böhme, T., Irmer, M.: Ontology-based content analysis of US patent applications from 2001–2010. Pharm. Patent Analyst **2**(1), 39–54 (2013)
41. Wohlgenannt, G., Belk, S., Karacsonyi, M., Schett, M.: Using an ontology learning system for trend analysis and detection. In: International Semantic Web Conference (Posters & Demos), pp. 37–40. Citeseer (2014)