IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 12 June 2025, accepted 8 July 2025, date of publication 17 July 2025, date of current version 28 July 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3590135

RESEARCH ARTICLE

From Code to Concept: A Semantic Approach to AI Innovation Discovery in Open Source Software Repositories

INNA NOVALIJA[1], DUMITRU ROMAN[2,3,4], FEDERICO BELOTTI[5], VLADIMIR ALEXIEV[6], LUIS REI[1], ROBERTO AVOGADRO[2], BABAK KHALILVANDIAN[5], BOYAN BECHEV[6], CATALINA ALEXANDRA CHINIE[4], IULIA CIUREA[4], JANEZ BRANK[1], COSMIN UDROIU[7], AHMET SOYLU[8], AND MATTEO PALMONARI[5]

[1]Jožef Stefan Institute, 1000 Ljubljana, Slovenia
[2]SINTEF AS, 7034 Trondheim, Norway
[3]Oslo Metropolitan University (OsloMet), 0176 Oslo, Norway
[4]Bucharest University of Economic Studies, 010374 Bucharest, Romania
[5]University of Milano–Bicocca, 20126 Milan, Italy
[6]Graphwise/Ontotext, 1124 Sofia, Bulgaria
[7]CS GROUP–ROMANIA, 200692 Craiova, Romania
[8]Kristiania University of Applied Sciences, 0107 Oslo, Norway

Corresponding author: Inna Novalija (inna.koval@ijs.si)

This work was supported in part by the enRichMyData Project under Grant HE 101070284, in part by UPCAST Project under Grant HE 101093216, in part by DataPACT Project under Grant HE 101189771, in part by CauseFinder Project under Grant PNRR 760049, and in part by ELIAS Project under Grant HE 101120237.

ABSTRACT Artificial Intelligence (AI) is a transformative force driving innovation, yet tracking AI-related advancements remains challenging due to the rapid pace of development and unstructured data from platforms like GitHub. This paper proposes an AI-driven approach to innovation detection, leveraging GitHub as a data source to systematically identify and link AI projects to organizations. Key contributions include a domain-specific taxonomy comprising 7,490 AI topics, a modular pipeline for semantic annotation and entity linking, and a trend detection framework based on Singular Spectrum Analysis (SSA). A knowledge graph is constructed to represent relationships among AI topics, projects, and companies, thereby enabling structured innovation tracking. The approach addresses challenges such as data sparsity and noise, demonstrating strengths in semantic annotation and topic categorization. Results highlight the potential for accurately detecting AI innovations and linking them to organizational entities, offering valuable insights for researchers, companies, and policymakers. This work contributes a scalable, automated approach for AI innovation tracking, with future directions focusing on refining entity linking and expanding the knowledge graph to capture emerging trends.

INDEX TERMS Artificial intelligence, data mining, text mining, time series analysis.

I. INTRODUCTION

Recent developments underscore the unprecedented pace and breadth of advancements in Artificial Intelligence (AI). According to the 2025 Stanford AI Index Report [1], AI systems are achieving remarkable gains on complex benchmarks. These capabilities are no longer confined to research labs - AI is increasingly embedded in everyday life, from FDA-approved medical devices to commercial autonomous vehicle fleets. Artificial Intelligence continues to reshape industries by accelerating innovation and redefining the technological landscape [2]. In this rapidly evolving context, the ability to monitor and analyze AI-related innovations over time is essential for

The associate editor coordinating the review of this manuscript and approving it for publication was Pinjia Zhang.

understanding technological progress, anticipating trends, and supporting strategic decision-making. However, the speed of AI development and the sheer volume of publicly available data, particularly on platforms like GitHub, present major challenges in detecting and attributing innovations in a systematic way [3]. This paper addresses these challenges by proposing an AI based method for identifying emerging innovations, using GitHub as a rich and dynamic source of real-world technological activity.

To address these challenges, this research proposes a robust approach for identifying and linking AI-related innovations to their originating organizations, with GitHub repositories as the primary data source.

A comprehensive knowledge graph is constructed [4], [5] to map relationships between AI topics, projects, and companies, enabling structured and scalable innovation tracking.

This work builds on prior research in AI innovation tracking, text mining, and knowledge graph construction, while addressing key limitations such as data sparsity, noise, and the lack of structured metadata in open-source platforms.

This paper advances the state of the art in AI innovation tracking through three key contributions that uniquely leverage software repositories as primary signals. **First**, a comprehensive taxonomy of 7,490 AI topics is constructed to support the semantic annotation of software artifacts. Unlike prior taxonomies focused on academic literature, this taxonomy is specifically engineered for high-resolution annotation of real-world AI development signals within GitHub repositories. **Second**, an integrated pipeline is developed to perform large-scale data acquisition, semantic enrichment, entity disambiguation, and trend analysis. In contrast to existing methods centered on academic graphs and idea generation, the proposed system links AI concepts and institutions directly to source code and developer metadata, enabling detection of practical, implementation-level innovations as they emerge. **Third**, a multi-scale evaluation of topic annotation, company linking and temporal trends is conducted, uncovering both short- and long-term AI innovation dynamics grounded in open-source code. By focusing on software repositories, the approach captures fine-grained innovation signals earlier in the development lifecycle than traditional bibliometric techniques. Positioned at the intersection of AI innovation and open-source development, this work introduces a novel perspective that enhances innovation monitoring for researchers, policymakers, and industry stakeholders navigating the rapidly evolving AI landscape.

The rest of the paper is structured as follows: Section II provides an overview of the background and related work, including existing methods for AI innovation tracking and the role of knowledge graphs in representing AI trends. Section III details the methodology for building the innovation detection pipeline, including data collection, topic extraction, and company identification.

Section IV presents the evaluation and results, focusing on the accuracy and performance of the proposed approach.

Section V presents the practical aspects of pipeline implementation in TAO framework [6]. Finally, Section VI concludes with a summary of findings, key insights, and future directions for AI innovation tracking.

II. BACKGROUND AND RELATED WORK

This section provides an overview of existing approaches and tools relevant to tracking innovation in artificial intelligence (AI).

The discussion begins with traditional methods for monitoring technological progress, such as bibliometric analysis, emphasizing their limitations in the context of fast-evolving AI ecosystems. Subsequently, the emergence of knowledge graphs is examined as a powerful means for representing, organizing, and reasoning about AI trends. The analysis then turns to topic extraction and entity linking, as well as the use of open-source software repositories as alternative indicators of innovation, concluding with a comparison to related work.

A. AI INNOVATION TRACKING

Tracking innovation is a complex process that requires integrating quantitative data with qualitative insight to obtain a comprehensive picture of emerging technologies. To date, most innovation tracking methods have focused on analyzing academic literature and intellectual property records. One common approach is bibliometric analysis, which explores academic publications through citation networks, co-authorship patterns, and keyword frequency to identify emerging research areas and technological shifts [7]. A related method is patent analysis, which evaluates patent filings and classification codes [8], and may reveal trends ahead of formal academic publications. Complementary approaches use text mining and natural language processing tools on news sources and web data to gauge sentiment and track topic prevalence, offering early hints of new areas of focus [9]. Delphi studies and expert surveys are often used to complement and validate quantitative findings [10]. While these methods have demonstrated long-standing value, they may be less effective in capturing the fast-paced and decentralized nature of certain AI developments. A considerable portion of innovation now occurs outside traditional publication and patent systems — for example, in open-source projects hosted on platforms like GitHub. These informal environments often provide early signals of practical advances and technology adoption trends that are not immediately reflected in academic literature or intellectual property records.

Moreover, AI progress is frequently algorithmic or data-driven — involving iterative improvements, novel training strategies, or unique datasets — which are not always easily patented or captured by conventional innovation metrics. In response to these limitations, AI-specific tracking methods have become increasingly common. These include monitoring model performance on standardized benchmarks

such as ImageNet [11] or GLUE [12], which offer measurable indicators of progress within specific subfields [13]. Another emerging approach, and the focus of this paper, involves the use of knowledge graphs to structure information about AI concepts, tasks, and entities, enabling a more dynamic view of the evolving AI landscape [14].

B. KNOWLEDGE GRAPH FOR AI TRENDS

Knowledge graphs offer a structured and semantically rich framework for representing and analyzing AI innovation. They model entities such as technologies, researchers, and organizations, along with their relationships, using knowledge triples that encode factual information in a machine-readable format. In addition to supporting data interoperability and dynamic context representation [15], they enable reasoning, discovery of hidden connections, and the integration of symbolic and data-driven AI approaches [16], [17], [18].

Table 1 summarizes prominent knowledge graphs that support the study of AI innovation across domains ranging from lexical resources to benchmark tracking and scientific knowledge management.

Using knowledge graphs in association with Large Language Models (LLMs) has attracted increased interest, due to its potential to enhance the results of LLMs [29]. These have been proven to support the generation of new ideas and innovation, but still face limitations such as failing to consider semanting embeddings in citations, analysing links between citations in a linear way and still requiring human evaluation [30].

To address the current challenges and limitations of knowledge graphs in identifying innovation potential and generating new ideas within the AI field, Gao et al. [30] have proposed the Graph of AI Ideas (GoAI), used in a GoAI agent, which analyses trends and generates new ideas. It uses a two-steps method for the identification of new ideas: graph exploration (relations between papers and citation semantics) and idea generation, using paths defined in the first step and assessing trends. The authors have also introduced GoAI-CoT-Reviewer, an evaluation model designed to assess and refine automatically generated ideas, which uses a reasoning process similar to human reviewers [30].

The approach proposed by Massri et al. [14] was developed to maintain a knowledge graph for AI innovation, using a wide dataset spanning from industry research to social media and technical forums and integrating various phases of the AI innovation lifecycle.

C. AI TOPIC AND COMPANY IDENTIFICATION

GitHub repositories and user profiles are rich with content related to new technologies and innovations. Having a map of these advancements can be very beneficial, e.g., for sponsors. However, human-made content is often filled with noise and lacks structure. This paper aims to enhance methods to link GitHub repositories to the AI topics and organizations their users are a part of. An automated method enables linking users to entities in the Wikidata Knowledge Graph and creating a comprehensive map.

Several subtasks were defined for this study, including the development of a test set for evaluating topic extraction and the creation of a separate test set for assessing entity linking. Particular emphasis was placed on annotation precision, enabling a comparative evaluation of the performance of each component.

The topic extraction test set was used to assess the following tool: I) Wikifier [31], semantic annotation service for 100 Languages. The entity linking test set was used to assess the following approaches: I) Alligator [32], [33], an ML-based algorithm that uses two neural networks to score candidate relevance; II) TableLlama [34], a 7B Llama-2 LLM fine-tuned on table-related data; and III) GPT-4o-mini, OpenAI's cost-efficient LLM that balances performance and cost [35].

D. COMPARISON TO RELATED WORK

Recent efforts, such as GoAI [30] and Massri et al. [14], have applied knowledge graphs to track progress in AI research and support idea generation by analyzing academic literature, citation networks, and publication metadata.

In contrast to literature-centric approaches, this work offers a novel perspective by leveraging open-source software repositories as a signal for detecting emerging AI innovations. Whereas prior methods rely on data sources that undergo formal peer review or legal validation, such as scientific publications or patents, GitHub repositories, particularly their code and documentation, capture innovations at the implementation stage, often well before they are formally published or patented.

While semantic enrichment and structured knowledge representation are similarly employed, the proposed methodology applies these techniques to software artifacts by linking GitHub repositories with AI topics and organizational entities. This application introduces distinct challenges - such as data noise, entity disambiguation, and concept mapping - that are less pronounced in traditional analyses of scientific literature.

General-purpose knowledge graphs such as DBpedia and ConceptNet offer broad semantic coverage across many domains [16], but their generality limits their usefulness for detecting domain-specific innovation signals.

In contrast, the proposed approach is explicitly tailored to capture the dynamics of AI innovation across the development lifecycle, from code to deployment.

This focus enables a more targeted and actionable view of technological progress, bridging a gap between software practice and innovation tracking that remains largely unexplored.

Mining GitHub data for research purposes presents both opportunities and limitations. As highlighted by Kalliamvakou et al. [3], GitHub repositories often contain

TABLE 1. Knowledge graphs with impact on AI innovation.

Knowledge Graph	Description	Key Applications
Google Knowledge Graph	Introduced in 2012 to enhance Google's search engine	Search enhancement, question answering; content understanding and ranking [19]
DBpedia [20]	Structured content extracted from Wikipedia and other Wikimedia projects [21]	Semantic search and information retrieval, enhancing explainable AI models [16]
WordNet [22]	Lexical database of English grouping words based on semantic similarity	Enables more nuanced semantic connections [16]
Wikidata [23]	Free and open knowledge base for structured data	Cross-domain knowledge representation
Wikiprompts [24]	A collaborative knowledge graph of prompts and prompt workflows [15]	Prompt composition and chaining; bot-executed workflows for generative AI outputs
Intelligence Task Ontology (ITO) [25]	Curated resource on AI tasks, benchmark results, and performance metrics [13]	Network-based analysis of tasks; integration with external data; research evaluation
ConceptNet [26]	Free semantic network enhancing machine understanding of words	Semantic space, multilingual knowledge, analogies, story understanding [27]
Open Research Knowledge Graph [28]	Structured scientific knowledge	Scientific knowledge management

noisy, incomplete, or unrepresentative metadata, and repository popularity does not always correlate with quality or innovation. These challenges underscore the need for robust preprocessing and semantic enrichment.

III. METHODOLOGY: BUILDING THE INNOVATION KNOWLEDGE DETECTION PIPELINE

Figure 1 illustrates the end-to-end architecture of the AI innovation detection pipeline. The process begins with the development of an *AI innovation taxonomy* that provides a structured vocabulary for categorizing emerging AI-related topics. This is followed by a *data collection and processing* phase, where metadata, textual content, and source code are extracted from GitHub repositories.

For the purposes of this study, the term metadata refers broadly to all textual fields associated with a repository and its owner, including the repository description,`README` file, and user profile information such as biography, company affiliation, and website. Two key enrichment steps are applied: semantic annotation using the Wikifier tool [31], and organization linking via the Alligator framework [32], [33]. The enriched data are then aligned with external knowledge bases such as Wikidata, Crunchbase, and a purpose-built AI innovation taxonomy, providing contextual grounding for entities and concepts. This integrated representation enables the derivation of structured signals from multiple data modalities.

Finally, the pipeline performs signal transformation, time series construction, and trend detection to reveal temporal patterns in AI innovation. The output consists of time-resolved indicators that support the identification and analysis of trends in the AI domain.

A. AI INNOVATION TAXONOMY

The AI innovation topic taxonomy[1] [36], used in this research, is a hierarchically structured controlled vocabulary with linked data, designed to support semantic annotation and topic extraction across diverse AI-related resources. It comprises 7,490 distinct concepts, systematically organized to reflect the breadth of the Artificial Intelligence domain.

Each concept is enriched with metadata including unique identifiers, preferred and alternate labels, broader categories, and descriptive definitions. This structure enables precise semantic tagging, effective filtering, and consistent categorization of AI topics.

A selection of example concepts is shown in Table 2, illustrating the diversity and granularity of the taxonomy. The taxonomy construction was based on a refinement of the methodology introduced in [37].

1) LEVERAGING WIKIPEDIA AS THE CORE SOURCE

Wikipedia has been selected as the primary source for developing the AI taxonomy, owing to its comprehensive coverage of important AI-related topics, encompassing historical, contemporary, and emerging concepts. Articles on Wikipedia are frequently updated and curated by a broad community, ensuring relevance and accuracy.

[1] https://zenodo.org/records/15113095

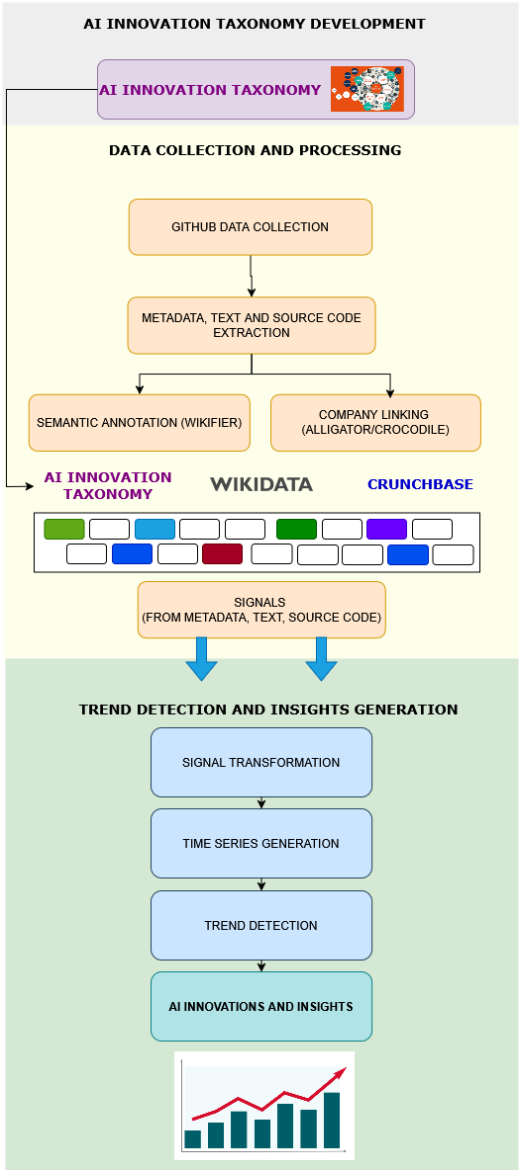


FIGURE 1. Pipeline architecture.

2) UTILIZING WIKIPEDIA CATEGORIES TO FORM A HIERARCHY

Wikipedia categories provide a structured way to classify and organize articles. They are a collaboratively developed hierarchy that can be used for navigation of AI-related topics.

These categories are utilized to identify and structure relevant AI concepts. The structure of Wikipedia categories forms a complex graph rather than a simple tree, as categories may have multiple parent paths and contain cycles. More importantly, there is significant semantic drift as the distance from the roots increases, leading to a huge hierarchy and loss of relevance.

3) CATEGORY-BASED ARTICLE DISCOVERY

Wikipedia's category system is employed to efficiently discover relevant articles.

TABLE 2. Sample topics from the AI taxonomy. Other metadata fields (e.g., descriptions, aliases, types, broader) are also included in the full taxonomy.

Topic	Wikidata ID
Natural language processing	Q30642
Artificial intelligence	Q11660
Machine translation	Q79798
Knowledge representation and reasoning	Q3478658
Computational linguistics	Q182557
Data mining	Q172491
Data science	Q2374463
Text mining	Q676880
Textual entailment	Q6588467

The category tree for Artificial Intelligence serves as a starting point, displaying the first-level categories along with subcategories and the number of articles within each.

4) CATEGORY PRUNING AND REFINEMENT

A systematic approach is adopted to prune and refine categories, ensuring the development of a well-structured taxonomy:

- **Selecting Root Categories**: The process begins with a set of root categories, including *Artificial Intelligence*, *Machine Learning*, *Data Science*, and several other closely related domains.
- **Building a Minimal Spanning Tree**: A minimal spanning tree is constructed by computing the shortest paths from the root categories to all reachable categories in the taxonomy.
- **Interactive Pruning**: An interactive tool is used to iteratively remove irrelevant branches, while monitoring the number of retained categories and articles to maintain a focused taxonomy.
- **Final Concept Selection**: Articles from the refined set of categories are collected and added to the tree structure.
- **Replacing Categories with Main Articles**: If a category has a designated "Main article," it is replaced with that article to improve clarity and organization.

5) INTERACTIVE TOOL FOR TAXONOMY CURATION

To facilitate the pruning and structuring process, an interactive tool has been developed to visually represent the AI category hierarchy (Figure 2). The tool supports the following functionalities:

- Navigating through the AI category tree
- Expanding and collapsing branches to evaluate subcategories
- Viewing all categories at a selected depth level
- Cutting irrelevant categories in real-time
- Restoring branches that were mistakenly removed
- Monitoring the live distribution of categories and articles during pruning

The bottom graph in Figure 2 displays the total number of categories (blue line) and their distribution by depth

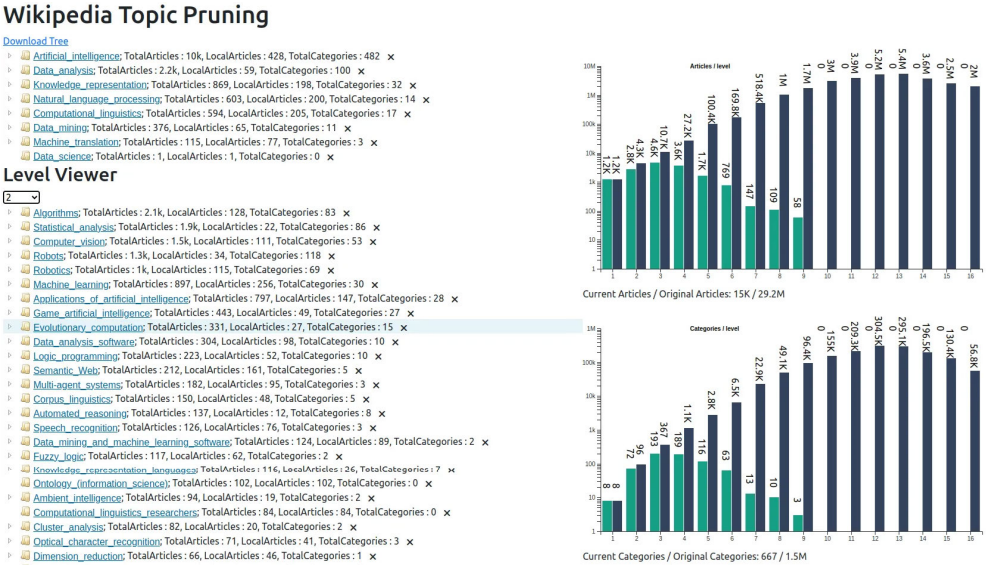


FIGURE 2. Interactive tool for AI taxonomy curation.

```
Artificial intelligence
→ Robotics
 → Robot control
 → Robot locomotion
  → Robots by method of locomotion
   → Flying robots
    → Airborne military robots†
     → Unmanned military aircraft
      → Drone warfare
       → 2020 Nagorno-Karabakh war
        → War crimes in the 2020 Nagorno-Karabakh war
```

LISTING. 1. Semantic drift example from the Wikipedia category hierarchy. The bolded category marked with † indicates the cut point.

(distance from the root categories). After 16 levels, nearly all 1.5 million Wikipedia categories are reachable. The green line shows the reduction resulting from pruning: after 359 cuts -primarily concentrated at levels 4, 5, and 6 - only 667 categories remain, representing just 0.044% of the original set. The maximum remaining depth is 9 levels, and the 58 surviving categories at this depth were all verified to be relevant, showing no signs of semantic drift.

The top graph shows the corresponding distribution of Wikipedia articles. After 16 levels, all 29.2 million articles are reachable. Following pruning, only 15,000 articles remain, corresponding to 0.051% of the initial number.

An example of semantic drift is provided in Listing 1, where a category was cut, along with all its descendants that lacked alternative paths to the root nodes.

6) MITIGATING SEMANTIC ANNOTATION CHALLENGES WITH AI INNOVATION TAXONOMY

General-purpose ontologies often lack the domain-specific granularity needed for nuanced AI topic classification. Moreover, they frequently include broad or ambiguous concepts that introduce noise into annotation tasks, thereby reducing interpretability and accuracy for AI-specific applications.

For instance, consider the following text excerpt from a GitHub repository:

> *In this python project, I am trying to build a Classification Machine Learning model to predict banknotes are genuine or forged. In real world this could mainly be any of the followings. Fraud detection; Counterfeit detection; Quality control; Authentication of banknotes. There are several valuable Business Impacts and Potential Benefits which we can define here. 1. Reducing financial losses; 2. Improve customer trust; 3. Enhancing operational efficiency; 4. Meeting any regulatory requirements. For this project stakeholders possibly be: Any banking system; Financial institutions; Law enforcement agencies; Any regulatory bodies. This project is based on Bank Notes Authentication UCI dataset. I'm using the Kaggle's version of it. I will be using: Machine Learning Algorithms*

for classification of banknotes; Various Python libraries to visualize different insights. Descriptive statistics will be used to derive valuable insights from the data. Following Machine Learning algorithms will be evaluated and the best performing model will be selected: 1. Logistic Regression; 2. Random Forest; 3. KNN Classifier; 4. Support Vector Machine Classifier. I have used pyforest library bundle for this project.

The text above can be annotated with a number of Wikipedia concepts, including *Kaggle*, *Machine learning*, *Python (programming language)*, *Banknote*, *Support vector machine*, *Quality control*, *Computer*, *Business*, *Bank*, *Law*, *Counterfeit*, etc.

While these are valid entities, several of them are not specific to the AI domain and thus dilute the semantic signal when tracking innovation. The AI Innovation Taxonomy facilitates focused analysis within the AI domain and reduces erroneous annotations caused by ambiguous concepts. Furthermore, the taxonomy allows disambiguation of context-sensitive terms such as *Transformer*, where domain knowledge is essential for accurate entity selection. For example, *Transformer (electrical device)* and *Transformer (deep learning architecture)* are distinct concepts with different Wikipedia and Wikidata entries.

Moreover, the taxonomy structures concepts hierarchically, enabling users and systems to distinguish between foundational techniques and application domains, and thereby supporting clearer innovation profiling.

7) OTHER RELEVANT CLASSIFICATIONS

In addition to Wikipedia, the following taxonomies were also researched, described, and extracted:

ACM Computing Classification System (CCS) [38], Academia Industry Dynamics' Focus Area Taxonomy (AIDA FAT) [39], ArnetMiner KGs [40], ANZSRC Fields Of Research (FOR) [41], arXiv Areas [42], Chinese NSFC [43], Computer Science Ontology (CSO) [44], Economics JEL [45], EU CORDIS EuroSciVoc [46], Medical Subject Headings (MSC) [47], Mathematical Subject Classification [48], Ontology of Data Science (OntoDM) Classes [49], OpenAlex Topics [50], SciGraph Subjects [51], SemanticScholar Fields Of Science (FOS) [52], StackExchange Tags [53].

The following elements are particularly important for the construction of the taxonomy:

- The Collaborative Patent Classification (CPC) [54]: this is the best way to seed an AI Application Areas taxonomy, since it is strongly oriented towards commercial applications;
- Papers With Code and Linked Papers With Code [55] offer a differentiated classification of AI research (e.g., challenges, tasks, methods, datasets), enabling the identification of a high concentration of machine learning papers, many of which include associated source code;
- Crunchbase Categories [56] enable the identification and classification of AI-focused companies and startups.

B. DATA COLLECTION AND INTEGRATION

Metadata and source code were collected from AI-related GitHub repositories through targeted crawling. The resulting data were cleaned, normalized, and structured into a standardized schema with well-defined fields, including id, `name`, `visibility`, `bio`, `website`, `company`, `location`, `created`, and `description`.

The final dataset comprises 561,932 repositories and their associated user profiles. Each entry includes a unique identifier, repository metadata (e.g., name, description,`README`), and user-level attributes such as biography, website, and company affiliation. Among these, the `bio`, `company`, and `website` fields are particularly relevant for linking repositories to organizations.

However, these user-specific fields exhibit significant sparsity: `bio` is missing in 24% of entries, while both `website` and `company` are missing in over 50%. This is likely due to the optional nature of profile completion on GitHub. Despite these limitations, the `company` and `bio` fields provide the most consistent signals for organizational linkage, even though the majority of users in the dataset appear to be individuals rather than institutions. Analyzing the word count distribution in the `bio` and `company` fields reveals notable patterns. The `company` values are highly skewed, with over 90% of entries containing fewer than five words. Longer entries often contain irrelevant content, verbose explanations, or multiple affiliations, which introduce noise and ambiguity. In contrast, the `bio` field tends to include more descriptive content about the individual, occasionally offering indirect clues about organizational affiliations.

To explore the semantic content of these fields, word cloud visualizations were generated based on term frequency. The `bio` word cloud, shown in Figure 3, prominently features terms such as "data science" and "machine learning", indicating a strong concentration of users with technical backgrounds, particularly in the AI domain.

The word cloud for the `company` field, shown in Figure 4, reveals two dominant themes. First, a substantial number of entries reference academic institutions, including specific mentions of universities, indicating that a significant portion of users are affiliated with higher education. Second, the presence of large technology companies reflects the expected overlap with the AI research and development ecosystem.

Among the most frequent entities in the `company` field are "Microsoft", "Amazon", "Google", "Student", and "Carnegie Mellon University". This further reinforces the dual presence of academic and industry affiliations, while also highlighting the prevalence of student contributors. Although some entries are ambiguous or unstructured, these dominant terms provide useful cues for downstream linking and classification tasks.

1) TOPIC ANNOTATION AND EXTRACTION

A critical component of the pipeline involves the extraction and categorization of AI-related topics from GitHub repositories.

This process enables the identification of thematic signals and facilitates meaningful trend analysis. The methodology consists of three main stages, designed to ensure both scalability and domain specificity:

- **AI-Specific Semantic Annotation:** In the first phase, textual metadata associated with GitHub repositories - including repository descriptions, `README` files, and user biographies - are processed using the Wikifier tool [31]. Wikifier performs entity linking by mapping free - form text to structured knowledge bases such as Wikipedia and DBpedia. It is optimized for large-scale semantic annotation of textual content, making it suitable for processing rich metadata at scale. Terms or phrases that can be linked are mapped to corresponding entities, enabling a structured semantic representation of otherwise unstructured content.
  A PageRank-based annotation threshold of 0.01 has been applied to ensure robustness.
- **Alignment with AI Taxonomy:**
  To focus the annotation results on AI-specific content, a domain-constrained filtering mechanism is applied. All Wikifier annotations are matched against a curated taxonomy of AI topics, as described above.
  The alignment ensures that only entities directly related to AI are retained, significantly improving the thematic precision of the annotations.
- **Topic Extraction and Structuring:** Following annotation and filtering, the retained AI-specific entities are used to assign each repository a set of topic labels. These labels represent the presence and frequency of AI concepts mentioned in the metadata. This structured topic information is then aggregated to generate higher-level semantic signals, which are later used for trend detection and innovation profiling. Importantly, this process supports temporal analysis by allowing researchers to trace the emergence, evolution, and co-occurrence of AI themes over time.

The examples in Appendix A-A illustrate semantic annotations applied to GitHub `README` texts. Each example consists of a `README` excerpt along with the corresponding Wikifier-generated annotations, which were further filtered using the AI Innovation Taxonomy to retain only AI-related concepts.

Moreover, the software requirements from GitHub repositories are used as an additional source for temporal analysis (see Appendix A-B).

Overall, the annotation and extraction pipeline enables high fidelity mapping of unstructured software metadata to a semantically rich and domain-specific representation. This not only enhances explainability and interpretability of downstream models but also facilitates robust knowledge discovery in the context of AI innovation.

TABLE 3. Data distribution of the Entity-Linking GitHub test set.

Category	Count	Percentage
NIL	462	50.4%
Not NIL	454	49.6%
Total	916	100%

2) AI COMPANY IDENTIFICATION

A test set was constructed from the GitHub dataset by first removing rows with missing `Company` values, yielding 221,617 rows with 56,429 unique companies. The `Company` field was cleaned using regex to remove emails, convert text to lowercase, eliminate special characters and extra spaces, and trim whitespace. To avoid duplicates, the data were grouped by cleaned company names, selecting the first entry from each group. Each row was manually annotated - using the `Company` field as the primary source and the `Bio` field for context—by assigning a Wikidata QID or marking it as NIL when no match was found. Figures 3 and 4 illustrate the word clouds generated for the `Bio` and `Company` sections, respectively. The manual annotation process involved searching Wikidata, verifying details via GitHub and LinkedIn profiles, cross-checking with company websites, consulting Crunchbase, and mapping GitHub contributors to companies.

A total of 916 unique companies were manually annotated, resulting in a balanced distribution between *Not In Lexicon* (NIL) entities - those not present in the underlying Knowledge Base (KB) - and non-NIL entities.

Table 3 presents the dataset's distribution. Section A-C contains examples from the dataset with specific challenges for company identification, such as multiple relevant organizations, granularity issues, entities with similar names, entities with multiple QIDs, irrelevant values in company column.

C. TEMPORAL ANALYSIS OF AI INNOVATIONS

1) CONCEPT TREND DETECTION

To analyze long-term patterns in the evolution of AI-related research and development, trend detection has been conducted at the concept level using metadata from GitHub repositories.

`README` files were annotated with semantic labels derived from the AI Innovation Taxonomy described in Section III-A. Each annotation was timestamped based on the corresponding repository's creation date and used to generate monthly occurrence counts for each taxonomy concept. These counts were aggregated into time series.

To improve statistical robustness and reduce noise, only concepts that occurred at least 100 times across the corpus have been retained, resulting in 558 concept-specific time series spanning the past 10 years.

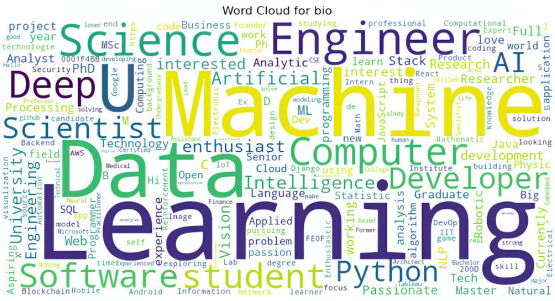


FIGURE 3. Word cloud for the `bio` field.

2) REQUIREMENTS TREND DETECTION

In parallel, temporal trends in software requirements have been analyzed based on information extracted from source code files (e.g., `requirements.txt`, `setup.py`) within the same repositories.

These files were parsed to identify software packages, frameworks, and libraries commonly used in AI development. As with the concept analysis, temporal frequencies were aggregated into monthly time series. Singular Spectrum Analysis (SSA) [57] has been applied to these time series to reveal adoption patterns of tools and technologies, providing insights into practical implementation trends and emerging toolchains within the AI community.

3) SINGULAR SPECTRUM ANALYSIS (SSA)

The concept and requirements time series have been constructed for the period from 2013 until 2024. This period was chosen to ensure a sufficiently long historical window to capture the evolution of AI-related innovation, while aligning with the growth trajectory of GitHub as a major platform for open-source development. Starting in 2013 enables the observation of early signals associated with the modern AI boom, including the emergence of deep learning topics.

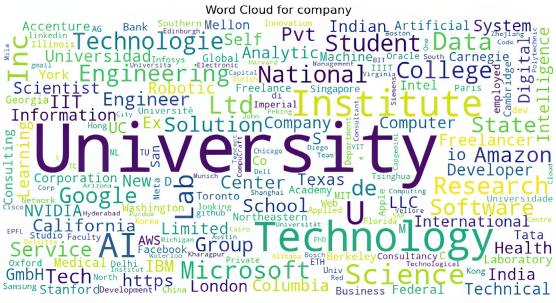


FIGURE 4. Word cloud for the `Company` field.

To extract underlying trends and reduce noise, Singular Spectrum Analysis (SSA) has been employed - a non-parametric, model-free technique for decomposing time series into interpretable components such as trend, periodicity, and noise [57]. SSA is particularly well-suited for irregular or noisy data, as it requires no prior assumptions about the data structure. The analysis was performed using sliding windows of 3, 6, and 12 months to capture various temporal resolutions. By decomposing and reconstructing the smoothed series, the dynamics of AI-related concepts were highlighted, enabling the identification of both emergent and declining innovation themes within the domain.

For each concept and window size, the following metrics have been computed:

- **SSA Score**: A measure reflecting the importance of a concept's trend component.
- **Top Trending Concepts**: Concepts ranked in the top 5% based on SSA scores were flagged as trending.

4) TREND DETECTION LIMITATIONS

Singular Spectrum Analysis (SSA) for trend detection has been chosen as it enables decomposition without assuming a specific model structure, which is particularly well-suited for exploratory analysis over noisy and irregular time series commonly encountered in open-source metadata. Unlike classical statistical models such as ARIMA [58], which require predefined order parameters, or Prophet [59], which assumes seasonality and trend decomposition with strong priors, SSA allows flexible decomposition without assuming any particular structure. This makes it a compelling choice for detecting latent innovation dynamics where domain knowledge is limited or trends may exhibit irregular evolution.

IV. EVALUATION AND RESULTS

A. AI TOPIC EXTRACTION EVALUATION

To evaluate the quality of topic annotations generated via the Wikifier tool, an experiment has been conducted using a manually curated gold standard dataset comprising `README` texts from 203 GitHub repositories. This dataset includes repositories annotated with relevant AI topics, serving as a reference for assessing the precision of automated annotations. The experimental setup involved computing precision scores at various cut-off ranks: Top 1, Top 3, and Top 10. Two scenarios have been compared: (i) utilizing all annotations regardless of confidence scores, and (ii) filtering annotations based on a PageRank threshold of 0.01, corresponding to higher-confidence semantic entities.

$$\text{Precision Top 1} = \frac{1}{N}\sum_{i=1}^{N}\frac{\left|\text{Relevant}_i \cap \text{Top1}_i\right|}{1}, \tag{1}$$

$$\text{Precision Top 3} = \frac{1}{N}\sum_{i=1}^{N}\frac{\left|\text{Relevant}_i \cap \text{Top3}_i\right|}{3}, \tag{2}$$

$$\text{Precision Top 10} = \frac{1}{N}\sum_{i=1}^{N}\frac{\left|\text{Relevant}_i \cap \text{Top10}_i\right|}{10}. \tag{3}$$

The gold standard has been constructed by randomly selecting and annotating 203 repositories from the GitHub dataset, with contributions from two expert annotators. The process involved several consecutive steps:

1) **Wikifier Annotation:** Each repository's `README` text has been annotated using the Wikifier tool to extract semantic concepts.

TABLE 4. Precision metrics at different levels for all annotations and filtered annotations based on PageRank threshold.

Metric	Precision
Precision Top 1 (All)	0.921
Precision Top 3 (All)	0.842
Precision Top 10 (All)	0.801
Precision Top 1 (PageRank > 0.01)	0.975
Precision Top 3 (PageRank > 0.01)	0.965
Precision Top 10 (PageRank > 0.01)	0.960

2) **AI Taxonomy Filtering:** The AI Innovation Taxonomy has been applied to the Wikifier annotations to retain only AI-relevant topics. This resulted in a set of suggested AI topics for each repository.
3) **PageRank Sorting:** The annotated terms have been sorted by PageRank value. This step is essential for computing top-1 and top-3 precision, as well as maintaining a relevance threshold.
4) **Expert Verification:** Annotators have manually verified the relevance of AI topics for each repository. They have reviewed the automatically suggested terms and assigned the most appropriate ones based on their expert judgment.

Inter-annotator agreement has been substantial: in 61% of cases, both annotators fully agreed on the correct annotations, while partial agreement was observed in 17% of the cases. The gold standard has been intentionally designed to capture a broader set of plausible annotations. In alignment with the exploratory nature of the research, an annotation was considered correct if at least one annotator marked it as relevant.

The results are summarized in Table 4. The baseline precision scores for all annotations are 0.921 (Top 1), 0.842 (Top 3), and 0.801 (Top 10). After applying the PageRank threshold, annotation precision improved significantly across all levels: 0.975 (Top 1), 0.965 (Top 3), and 0.960 (Top 10). These findings indicate that filtering semantic annotations by PageRank can substantially improve the relevance of detected topics.

B. AI COMPANIES LINKING EVALUATION

The proposed entity-linking approaches - namely, Alligator [32], [33], TableLlama [34], and GPT-4o-mini [35] - are evaluated by addressing two main research questions:

1) How effectively does the algorithm identify NIL entities?
2) How accurately does the algorithm predict the correct candidate?

To address the first research question, performance metrics are reported as follows:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{4}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{5}$$

TABLE 5. NIL detection performance. For each metric, the highest score is highlighted in bold, while the second-highest score is displayed in underlined text.

Approach	Precision	Recall	F1
Alligator (greedy)	**0.715**	0.381	0.497
Alligator nil@0.2	0.585	**0.974**	<u>0.731</u>
TableLlama-7B	**0.715**	0.403	0.515
GPT-4o-mini	<u>0.685</u>	<u>0.922</u>	**0.786**
Wikidata	0.611	0.831	0.705

TABLE 6. Not-NIL and overall accuracy. The highest score is highlighted in bold, while the second-highest score is displayed in underlined text.

Approach	Not-NIL Accuracy	Overall Accuracy
Alligator (greedy)	0.434	0.407
Alligator nil@0.2	0.247	<u>0.614</u>
TableLlama-7B	**0.555**	0.476
GPT-4o-mini	<u>0.496</u>	**0.712**
Wikidata	0.284	0.560

$$\text{F1 Score} = \frac{2\text{TP}}{2\text{TP} + \text{FP} + \text{FN}}, \tag{6}$$

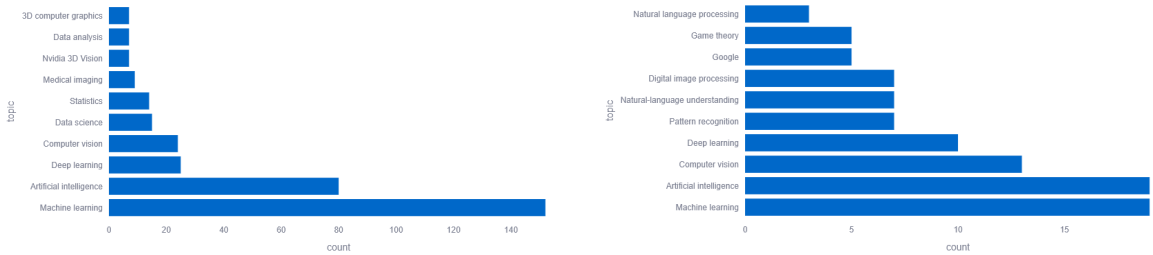
The evaluation follows a "NIL vs All" approach, where the metrics are defined as follows:

- **TP (True Positives):** Cases where the target is NIL and the prediction is also NIL.
- **FP (False Positives):** Cases where the target is a QID but the prediction is NIL.
- **FN (False Negatives):** Cases where the target is NIL but the prediction is a QID.
- **TN (True Negatives):** Cases where both the target and the prediction are QIDs (regardless of whether the QIDs match exactly).

In response to the second question, the evaluation reports accuracy computed exclusively over the non-NIL entity subset.

LamAPI [60] has been employed as the Entity Retrieval (ER) system. For each mention requiring annotation, up to 50 relevant candidate entities are retrieved. To isolate the evaluation of disambiguation performance, the retrieval system is configured to always include the correct candidate among the retrieved set. This approach ensures that the assessment remains independent of the information retrieval system. Each instance in which LamAPI fails to return candidates is considered a NIL prediction. For the Alligator system, two configurations have been evaluated: the *greedy* model, which consistently selects the candidate with the highest score, and an alternative that predicts NIL when the top candidate's score is below 0.2. This latter configuration is referred to as Alligator nil@0.2 in Tables 5 and 6. Additionally, the performance of the standard Wikidata Lookup Service[2] is reported as a baseline.

[2]https://www.wikidata.org/w/api.php?action=help&modules=wbsearchentities



(a) Innovation Topics at Imperial College London

(b) Innovation Topics at University of Amsterdam

FIGURE 5. Top innovation topics extracted from GitHub repositories by institution.

The results in Table 5 and Table 6 highlight the trade-offs between Precision, Recall, F1, and Accuracy across the evaluated approaches. The Alligator nil@0.2 variant achieves high recall (0.974) at the expense of precision, while the Alligator greedy variant excels in precision (0.715) but has lower recall. TableLlama-7B provides balanced performance, whereas GPT-4o-mini attains the highest F1 score (0.786) and robust non-NIL performance. Although TableLlama-7B achieves the highest non-NIL accuracy (0.555), GPT-4o-mini leads in overall accuracy (0.712). The Wikidata baseline, despite consistent performance, falls short compared to the advanced models. These findings suggest that while GPT-4o-mini is best overall, Alligator demonstrates superior computational efficiency while maintaining a good performance, making it the optimal choice in resource-constrained environments demanding scalability and rapid inference. In fact, it can be deployed on standard CPU infrastructure with minimal latency [61]. For these reasons, Alligator is the model employed in this work for company linking.

Below we provide an extensive error analysis for entity linking task.

- **GPT-4o-mini**. In instances where the model should have returned a NIL output, it instead generates a spurious link to an entity that is only superficially similar, thereby introducing erroneous associations. In most of these cases, the mention is either (a) a novel or very small organization, or (b) clearly not an entity at all. Additionally, in the majority of errors (62%), the correct candidate is not among the retrieved ones, with GPT returning either NIL or links to a different substring-match. In the presence of acronyms, GPT often links to the "most common" entity for that acronym: *CINVESTAV*, which stands for *"Center for Research and Advanced Studies (CINVESTAV) in Mexico"*, is linked to *"CINVESTAV Monterrey"*. This behavior arises from the absence of disambiguating context in the table - such as explicit mention of the entity's location - which limits the model's ability to resolve references accurately. Approximately 24% of the errors are due to ranking mistakes that reflect a *popularity bias*, whereby the model tends to select a more general or frequently observed sibling or parent entity. In practice, this often results in incorrect linking to high-frequency tokens; for instance, the mention *"Carleton University & Gov of Canada"* may be erroneously resolved to *"Gov"* rather than the intended compound or more specific target entity.
- **TableLlama**. The predominant source of errors for TableLlama (59%) stems from its tendency to avoid predicting NIL when appropriate. Instead, it frequently selects entities based on superficial substring overlap, often resulting in generic or semantically unrelated knowledge base (KB) entries. This behavior may be attributed to the absence of NIL examples during the model's extensive fine-tuning, leading to an implicit bias against abstention. In the case of GPT, 34% of the errors are due to the absence of the correct entity in the candidate set, preventing accurate resolution. Additionally, TableLlama exhibits a systematic preference for selecting broader or more popular sibling or parent entities. For instance, the specific mention *University of Texas at Arlington* is linked to *University of Texas at Arlington* with the description *public research university located in Arlington, Texas, USA [type] public research university, public educational institution*, rather than the ground truth *University of Texas at Arlington Department of Marketing [description] business school [type] business school*. A similar overgeneralization pattern is also observed in GPT's handling of acronyms.
- **Gemma-2-9B**. For the other large language models (LLMs), a significant proportion of errors (62%) arises from over-linking NIL entities. Specifically, Gemma2 tends to identify candidates based on overlapping substrings, resulting in incorrect linkages even when the correct resolution is NIL. Additionally, in 22% of the cases, the correct entity is absent from the retrieved candidate list, precluding accurate linking. Similar to TableLlama, Gemma2 also exhibits a tendency to select broader or more frequently observed sibling or parent entities in place of precise matches, contributing to 16% of its total errors.
- **Alligator-nil@0.2**. Alligator's default threshold of 0.2 for NIL prediction renders it overly conservative in linking valid knowledge base (KB) entities with lower popularity, accounting for approximately 90% of its

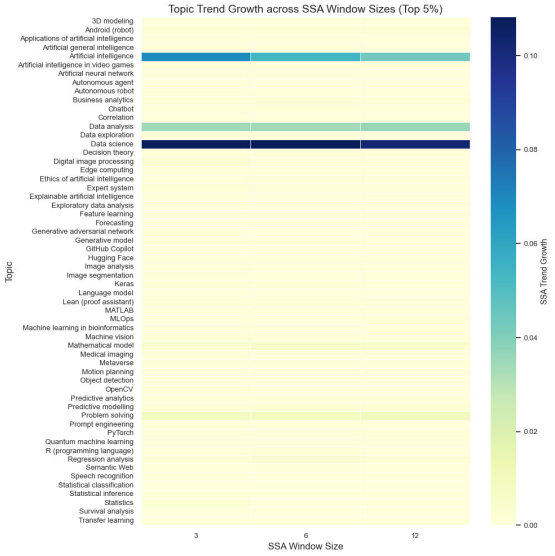


FIGURE 6. Trending topics analysis (by SSA, top 5%).

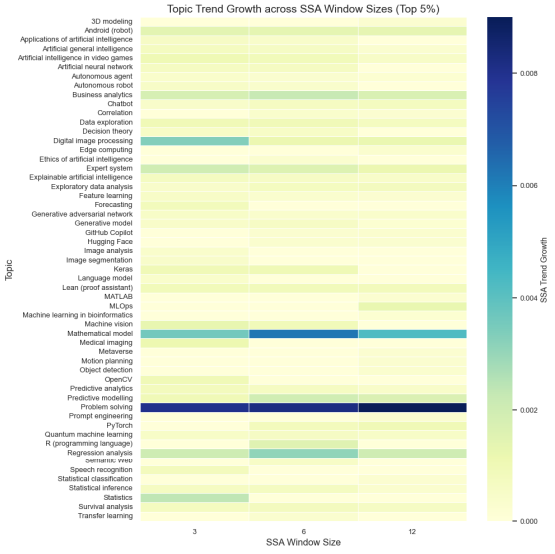


FIGURE 7. Trending topics analysis, filtered (by SSA, top 5%).

errors. As a result, entities with limited prominence - such as smaller colleges, recently established companies, or specific departments - typically fall below the threshold and are not linked, despite being valid targets. Multi-part affiliations involving semicolons, slashes, or ampersands often exacerbate this issue, either suppressing the match score below the threshold (false negatives) or promoting erroneous substring-based overlinking (false positives). In 3.4% of cases, mentions not present in the KB share sufficient lexical overlap with moderately popular KB entries to exceed the threshold spuriously, leading to incorrect linkage. Furthermore, since Alligator's features are primarily derived from the KB and basic syntactic patterns, its performance degrades in the presence of contextual ambiguity, typographical errors, acronyms, or multi-component mentions, all of which contribute to increased error rates.

Figures 5a and 5b illustrate the top innovation topics identified in GitHub repositories linked to two prominent academic institutions - Imperial College London and the University of Amsterdam. The resulting topic distributions demonstrate the effectiveness of the entity linking method in associating repositories with their institutional origins and reveal the thematic research orientations characteristic of each university. For example, Imperial College exhibits a strong emphasis on *Machine learning* and *Artificial intelligence*, with fewer projects on specialized subfields like *3D computer graphics* or *Medical imaging*. In contrast, the University of Amsterdam shows a broader thematic spread, with significant activity in *Machine learning*, *Computer vision*, *Deep learning*.

These results reinforce the potential of the proposed approach to surface meaningful, institution-specific innovation signals from open-source code repositories.

C. TREND DETECTION EVALUATION

Temporal dynamics of AI-related topics and technical requirements have been assessed using Singular Spectrum Analysis (SSA) across multiple window sizes (3, 6, and 12 months). SSA has been selected due to its effectiveness in decomposing complex time series into interpretable trend components.

The window size in SSA determines the length of the sliding window used to create the trajectory matrix from time series. The evaluation focused on identifying both the strength and predictability of trends in annotated AI concepts and extracted software dependencies.

The analysis yielded several key insights:

- **Strong Conceptual Signals:** Several AI concepts demonstrated high SSA trend scores, indicating their increasing prominence and sustained relevance within the GitHub ecosystem. Notable examples include *Artificial Intelligence*, *Data Analysis*, and *Data Science*, which appeared consistently in the top 5% across multiple SSA window sizes (Figure 6). These topics consistently exhibit high SSA trend scores, suggesting long-term and sustained interest in the AI research and development landscape.
- **Specific Conceptual Signals:** Figure 7 presents the top % of trending AI topics based on Singular Spectrum Analysis (SSA) (excluding concepts with strongest signals, such as *Artificial Intelligence*, *Data Science* and *Data Analysis*). By analyzing trend growth across multiple SSA window sizes (3, 6, and 12 months), the visualization captures both short-term fluctuations and long-term developments. Topics such as *Problem Solving*, *Mathematical Model* show consistently high trend scores, indicating sustained growth and increasing relevance in AI development. The variation in trend

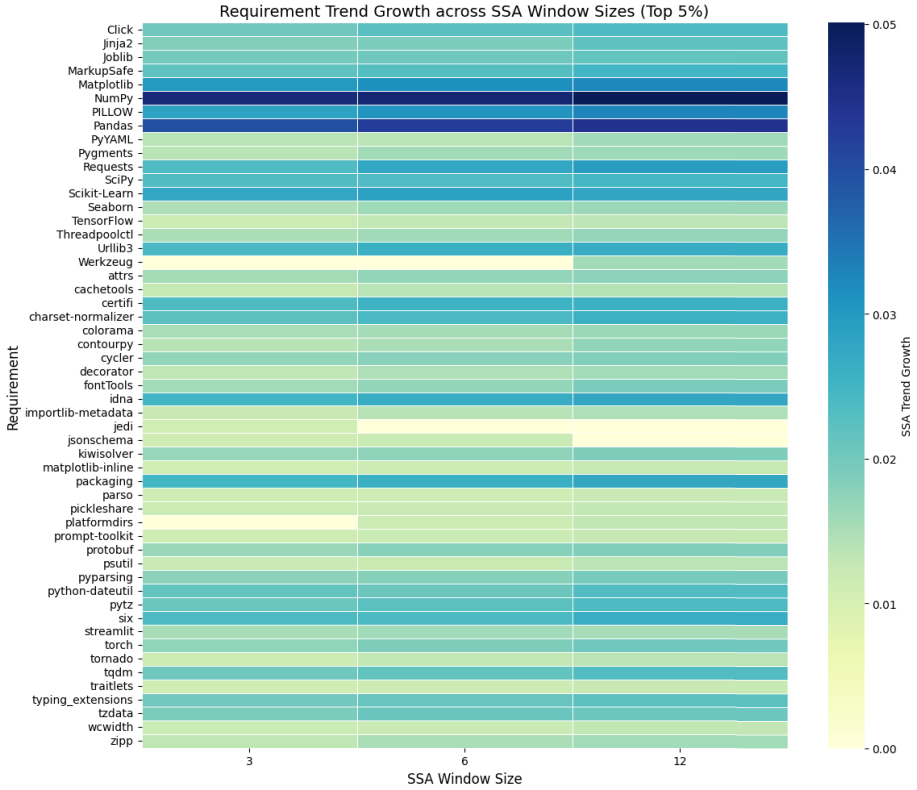


FIGURE 8. Trending requirements analysis (by SSA).

intensity across window sizes highlights temporal dynamics, with larger windows emphasizing enduring trends and shorter windows capturing recent shifts.

- **Multi-Scale Analysis:** The application of SSA using multiple window sizes enabled the differentiation between short-term fluctuations and long-term developments. Concepts that demonstrated consistent trends across these scales were interpreted as reflecting sustained community interest and stable adoption trajectories.
- **Requirement-Level Trends:** In addition to conceptual trends, the analysis of software requirements (Figure 8) revealed adoption patterns of widely used Python libraries and frameworks such as `pandas`, `scikit-learn`, `numpy`, and `matplotlib`. These results reflect the tool preferences of AI developers and signal shifts in implementation practices over time.

This multi-scale trend analysis approach offers a nuanced understanding of how AI innovation themes and tool usage evolve over time. It highlights not only dominant topics but also emerging concepts and tools with growing relevance across short to long temporal horizons.

V. PIPELINE IMPLEMENTATION

To operationalize the proposed semantic approach, the complete processing pipeline has been implemented using the TAO (Tool Augmentation by user enhancements and Orchestration) framework [6]. TAO offers an extensible and user-friendly environment for the design, execution, and management of complex data workflows. Its modular architecture allows the composition and distribution of processing pipelines, enabling independent definition and integration of additional modules by end users.

A processing module may be a standalone executable, a script, or an external service. TAO thus facilitates the integration and orchestration of heterogeneous components and libraries for data enrichment purposes.

To implement the pipeline, the following steps were carried out:

- **Preparation of resources:** This included setting up execution nodes, processing components, data sources, and data inputs. Docker images and corresponding descriptors were created for the tools used in the pipeline (e.g., *Wikifier*) (Figure 9)
- **Definition of the InnoGraph workflow:** The workflow is designed as a directed acyclic graph (DAG) composed of processing operations applied to inputs and producing at least one output. It is described using formal flow diagrams, where modules (processing components) are visually created via drag-and-drop interfaces. Modules are connected through directed lines, linking **input ports** (blue dots) to **output ports** (yellow dots). Each port specifies the expected format (e.g., file, folder), and compatibility checks are enforced when establishing

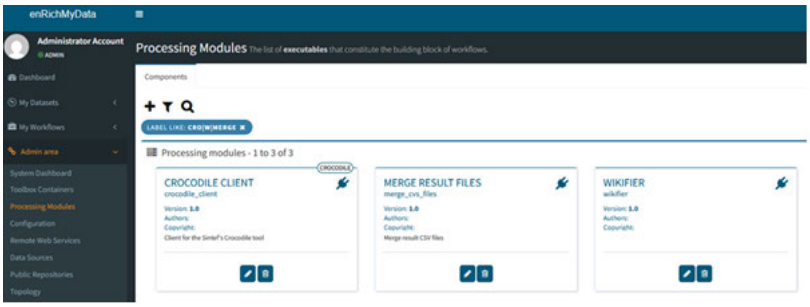


FIGURE 9. TAO interface: Processing modules.

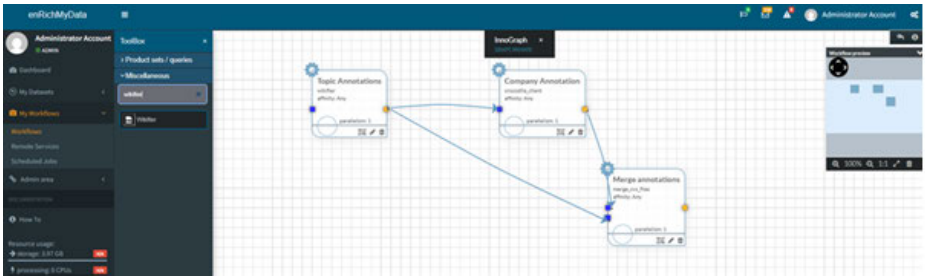


FIGURE 10. TAO interface: Workflow definition.

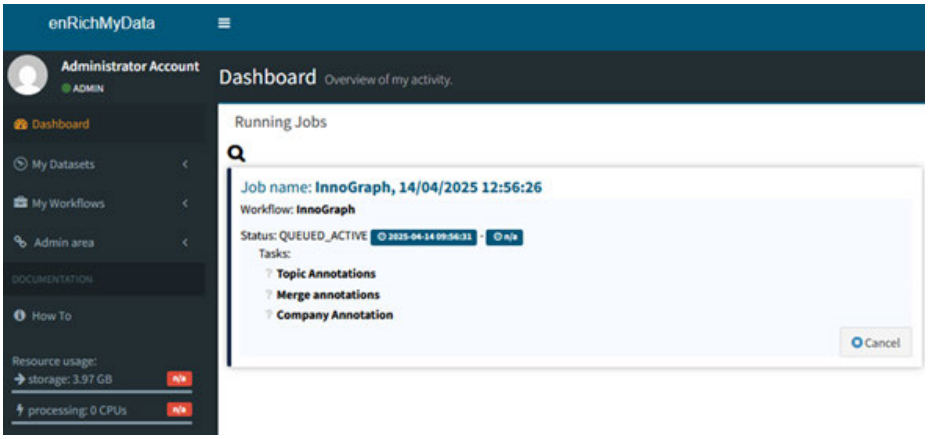


FIGURE 11. TAO interface: Dashboard demo.

connections. Module parameters can be edited via a pencil icon. Workflows can be created from scratch, modified, deleted, or cloned from existing templates. (Figure 10 in appendix B)

- **Execution of the workflow:** The TAO framework executes workflows using distributed resource managers with support for Kubernetes, Torque, Slurm, and local or SSH execution modes. TAO also provides capabilities for dynamic provisioning of virtual machines (VMs) within OpenStack environments. Through OpenStack's API-driven architecture, TAO can automatically create, configure, and launch VMs tailored to specific resource needs, including memory, storage, CPU, and network settings. (Figure 11 in appendix B)
- **Retrieval and visualization of results:** After execution, the results are retrieved and can be visualized or exported for further use.

Figure 9 illustrates a simplified view of the pipeline in TAO, showing how modules for annotation, enrichment, and organization linking are composed into an executable workflow.

One of the key advantages of implementing the innovation detection pipeline within the TAO (Tool Augmentation and Orchestration) framework lies in its modular architecture and visual workflow composition, which significantly enhances reproducibility, reusability, and domain portability. Each processing step - such as data ingestion, semantic annotation, entity linking - is encapsulated in independent, containerized

TABLE 7. Glossary of abbreviations used in the paper.

Abbr.	Term	Description
AI	Artificial Intelligence	A field of computer science focused on creating systems capable of intelligent behavior.
API	Application Programming Interface	A set of protocols and tools for building and integrating software applications.
CPU	Central Processing Unit	The main processor responsible for executing most instructions in a computer.
DAG	Directed Acyclic Graph	A finite graph with directed edges and no cycles, used to structure workflows.
ER	Entity Retrieval	The process of retrieving potential KB entities for a given text mention.
GPU	Graphics Processing Unit	A processor optimized for large-scale parallel computation, useful in ML tasks.
KB	Knowledge Base	A centralized, organized collection of information in a digital format
LLM	Large Language Model	A deep learning model trained on large corpora, capable of understanding and generating natural language.
ML	Machine Learning	A subfield of AI focused on learning patterns from data to make predictions.
NIL	Not In Lexicon	Indicates that an entity is not present in the underlying knowledge base during entity linking.
NLP	Natural Language Processing	A branch of AI focused on enabling computers to understand human language.
PR	PageRank	A link analysis algorithm used to prioritize high-confidence semantic annotations.
QID	Wikidata Q Identifier	A unique ID assigned to entities in Wikidata (e.g., Q11660 for AI).
SSA	Singular Spectrum Analysis	A model-free time series technique for trend extraction, noise reduction, and pattern discovery.
TAO	Tool Augmentation and Orchestration	A framework for building and executing modular data processing workflows.
VM	Virtual Machine	A software emulation of a physical computer for isolated execution.

modules that can be reused or replaced without modifying the entire workflow. This makes the system highly adaptable to other domains with similar needs for concept extraction and innovation tracking.

For example, adapting the pipeline to domains such as biotech or climate tech would primarily involve replacing the AI-specific taxonomy and annotation filters with domain-specific ontologies (e.g., MeSH [47] for biomedical topics). Since TAO supports drag-and-drop configuration of directed acyclic workflows, non-specialist users can rewire modules or integrate new tools (e.g., domain-specific NLP annotators) via Docker containers with minimal effort. Additionally, TAO's ability to track configurations, inputs, and outputs ensures end-to-end reproducibility across different runs or datasets.

VI. CONCLUSION

This study presents a novel, AI-driven methodology for detecting innovation within open-source code repositories, with a specific emphasis on GitHub. By integrating semantic annotation, an AI-specific innovation taxonomy, and entity linking, the approach enables the systematic extraction of structured insights into emerging AI trends. The implemented pipeline supports high-fidelity topic annotation via the Wikifier tool, incorporates mechanisms for company linking, extracts technical requirements and conceptual signals, and enriches textual data through the linkage of AI topics, projects, and organizations.

Although the methodology is tailored for detecting AI innovation in GitHub repositories, the underlying pipeline is potentially adaptable to other technology domains. Adaptation would require domain-specific modifications, particularly in taxonomy construction, semantic annotation, and concept filtering. For instance, applying the framework to other domains would necessitate the use of curated vocabularies and knowledge bases that accurately capture the terminology and innovation signals specific to those fields. While the modular implementation within the TAO framework supports such adaptation, empirical validation in non-AI domains remains an open avenue for future investigation.

In contrast to previous approaches that primarily rely on academic publications and citation networks, this work introduces a novel signal source: open-source software repositories, encompassing both metadata and source code for analysis. A comprehensive, multimodal pipeline is presented to capture implementation-level innovation signals. Furthermore, these signals are linked to organizations and temporal markers, enabling institutional and temporal analyses that are not supported by existing tools.

The concept trend detection, enabled by Singular Spectrum Analysis (SSA), reveals both short- and long-term trends in AI development.

The evolving AI ecosystem is effectively captured by detecting links between technologies, institutions, and contributors. This structured representation allows not only retrospective analysis but also the potential for forward-looking innovation forecasting. The study discovers dominant and volatile AI topics, reflecting the dynamic nature of AI research. Requirements-level trend detection further highlights the popularity and adoption timelines of foundational AI tools.

The implications of this work span multiple stakeholder groups. For researchers, it offers a scalable way to explore the

state of the art and detect emerging topics. For companies, the approach provides actionable intelligence to monitor competitors and align strategic investments. Policymakers can leverage the insights to guide funding and regulatory frameworks for AI development.

Nonetheless, several challenges remain. Entity linking to organizations is hindered by data sparsity, ambiguity, and the limited coverage of non-Western companies in current knowledge graphs. Moreover, fine-tuning of entity linking models like Alligator yielded limited improvements, signaling the need for domain-specific training data and refined features.

Future work will focus on enhancing entity linking through hybrid symbolic-neural methods, multilingual support, and improved temporal reasoning within the knowledge graph.

In parallel, trend detection will be advanced by exploring additional techniques, including temporal word embeddings, attention-based models, and change-point detection - to better capture emerging signals. The integration of external data sources (e.g., arXiv, technical blogs) will further enrich the analysis.

To evaluate robustness and responsiveness, SSA will be systematically compared with alternatives such as ARIMA, Prophet, and time-aware neural models across diverse data modalities. This multimodal benchmarking will inform the development of a more comprehensive and explainable innovation monitoring framework, supporting lifecycle analysis from early research to implementation and adoption.

In summary, this work lays the foundation for a more transparent, automated, and explainable framework for tracking AI innovation - one that is capable of evolving in parallel with the rapidly advancing technologies it aims to map.

APPENDIX A
EXAMPLES

A. TOPIC EXTRACTION EXAMPLES

Example 1

- **idd:** 65
- **Text:** Machine learning method examples with R
  Used datasets: iris, Glass, Shuttle, Groceries.
  For more datasets: `library(help=mlbench)`.
  Methods listed: Kmeans, Neural network, Random Forest, SOM, Naive Bayes, Associative rules, Decision tree.
- **Annotations:**
  - K-means clustering (0.0238)
  - Machine learning (0.0224)
  - Decision tree (0.0175)
  - Neural network (0.0156)
  - Random forest (0.0148)
  - Naive Bayes classifier (0.0114)

Example 2

- **idd:** 135
- **Text:** Project about recoloring images and looking at color bias for a course in Deep Learning at Chalmers HT22.
  Dataset: https://www.kaggle.com/datasets/arnaud58/landscape-pictures?resource=download
- **Annotations:**
  - Deep learning (0.0716)
  - Machine learning (0.0147)
  - Color (0.0111)

Example 3

- **idd:** 94
- **Text:** Machine-Learning-Recipes — using Scikit-learn and Tensorflow. This repository contains basic implementations of popular classifiers used in machine learning.
- **Annotations:**
  - Machine learning (0.0629)
  - Scikit-learn (0.0249)
  - Statistical classification (0.0161)

B. REQUIREMENTS EXAMPLES

Example 1

idd: 315

Repository: `Bird-Species-Classification`

Description: Machine learning model that predicts the species of a bird from its image.

Requirements:

- TensorFlow (2.5.1)
- Keras-Applications (1.0.8)
- Keras-Preprocessing (1.1.0)
- Flask (1.1.1), Gunicorn (20.0.4)
- NumPy (1.16.4), h5py (2.10.0), protobuf (3.11.2)
- Additional: Jinja2, soupsieve, termcolor, pytube, requests, astor, grpcio

Example 2

idd: 417

Repository: `TensorANFIS`

Description: Simple implementation of an Adaptive Neuro-Based Fuzzy Inference System (ANFIS) in TensorFlow. Includes example code for predicting the Mackey Glass time series.

Requirements:

- Python (3.5.5)
- TensorFlow (1.15.2)
- NumPy (1.15.2)
- Matplotlib (3.0.0)

Example 3

idd: 511

Repository: `WANN (Weighting Adversarial Neural Network)`

Description: Supervised domain adaptation method for regression, reweighting source losses to correct for distribution shifts. Evaluated against several baselines (e.g., KMM, DANN, ADDA).

Requirements:

- TensorFlow (>=2.0)

- scikit-learn, numpy, pandas, matplotlib, nltk
- `adapt` (domain adaptation library)

Datasets: CityCam, Sentiment Analysis

C. COMPANY IDENTIFICATION EXAMPLES

1) MULTIPLE RELEVANT ORGANIZATIONS

Example 1

- **Idd:** 201477
- **Company:** @facebook, @YonseiUniversity, @UCRiverside
- **Bio:** "Hello:) I'm a CS student at UC Riverside. I used to work on Instagram Reels Recommendation Team."

Example 2

- **Idd:** 47104
- **Company:** A member of intern @OpenGVLab, Shanghai AI Laboratory
- **Bio:** "PhD student. A member of intern @OpenGVLab, Shanghai AI Laboratory."

Example 3

- **Idd:** 500879
- **Company:** Data Scientist @ Amex, Ex-ML Intern @ Apple, BNP Paribas
- **Bio:** "Computer Science + Business Intelligence Analytics Graduate @ Univ. of Texas at Dallas."

2) GRANULARITY ISSUES

Example 1

- **Idd:** 80323
- **Company:** @GirardeauLab
- **Bio:** "i-Bio PhD fellow at Sorbonne Université. Electrophysiological analysis and machine learning."

Example 2

- **Idd:** 6405
- **Company:** Epson Europe B.V
- **Bio:** "Software Engineer with a focus on Optimisation/ Machine Learning/ Neural Networks/ AI/ Applied Mathematics/ Games."

Example 3

- **Idd:** 163789
- **Company:** @Liquid-Reply
- **Bio:** "Software Engineer with a mix of Industrial Engineering and Software Development."

3) ENTITIES WITH SIMILAR NAMES

Example 1

- **Idd:** 74424
- **Company:** deeper
- **Bio:** "I am an electronic engineer. Now I am interested in deep learning. I have done a lot on deep learning, caffe, and tensorflow."

Example 2

- **Idd:** 493311
- **Company:** Netsmart
- **Bio:** "Senior Software Engineer at Netsmart. Graduate with a focus on Artificial Intelligence."

4) ENTITIES WITH MULTIPLE QIDs

Example 1 - This example has been fixed in Wikidata, one QID redirects to the other now

- **Idd:** 233220
- **Company:** @iDTLabssl, @Code4SierraLeone, @CodeForAfrica
- **Bio:** N/A
- **QIDs:** Q117184139, Q105006683

5) IRRELEVANT VALUES IN THE COMPANY COLUMN

Example 1 - HR refers to his name

- **Idd:** 170282
- **Company:** HR Graphic
- **Bio:** "Front-end developer, Html, CSS, JavaScript, Bootstrap, and React expertise."
- **Website:** linkedin.com/in/hafiz-muhammmad-rayyan/

Example 2

- **Idd:** 379034
- **Company:** Open for hire
- **Bio:** "Flask | SQLAlchemy | Keras | C++ | Backend developer with experience in building music streaming services."

APPENDIX B
PIPELINE IN TAO

See Figures 10 and 11.

APPENDIX C
GLOSSARY OF ABBREVIATIONS

See Table 7.

ACKNOWLEDGMENT

The authors would like to thank the members of enRichMyData, DataPACT, UPCAST, CauseFinder, and ELIAS projects, for their valuable contributions, collaboration, and support throughout the development of this work. Their insights and expertise in data enrichment, infrastructure, and integration strategies played an important role in shaping the research direction and technical foundation of this project.

REFERENCES

[1] Stanford Inst. for Human-Centered Artif. Intell. (2025). *AI Index Report 2025*. Accessed: Apr. 15, 2025. [Online]. Available: https://hai.stanford.edu/ai-index/2025-ai-index-report

[2] I. M. Cockburn, R. Henderson, and S. Stern, "The impact of artificial intelligence on innovation," Nat. Bur. Econ. Res., Cambridge, MA, USA, Tech. Rep. 24449, 2018. [Online]. Available: https://www.nber.org/papers/w24449

[3] E. Kalliamvakou, G. Gousios, K. Blincoe, L. Singer, D. M. German, and D. Damian, "The promises and perils of mining GitHub," in *Proc. 11th Work. Conf. Mining Softw. Repositories*, May 2014, pp. 92–101, doi: 10.1145/2597073.2597074.

[4] A. Hogan, E. Blomqvist, M. Cochez, C. d'Amato, G. D. Melo, C. Gutiérrez, S. Kirrane, J. E. L. Gayo, R. Navigli, S. Neumaier, A.-C. N. Ngomo, A. Polleres, S. M. Rashid, A. Rula, L. Schmelzeisen, J. Sequeda, S. Staab, and A. Zimmermann, "Knowledge graphs," *ACM Comput. Surveys*, vol. 54, no. 4, pp. 1–37, Jul. 2021.

[5] V. Ryen, A. Soylu, and D. Roman, "Building semantic knowledge graphs from (Semi-)Structured data: A review," *Future Internet*, vol. 14, no. 5, p. 129, Apr. 2022.

[6] E. Ras, J. Swietlik, P. Plichart, and T. Latour, "Tao—A versatile and open platform for technology-based assessment," in *Sustaining TEL: From Innovation To Learning and Practice*. Berlin, Germany: Springer, 2010, pp. 644–649.

[7] T. U. Daim, G. Rueda, H. Martin, and P. Gerdsri, "Forecasting emerging technologies: Use of bibliometrics and patent analysis," *Technol. Forecasting Social Change*, vol. 73, no. 8, pp. 981–1012, Oct. 2006.

[8] H. Ernst, "Patent information for strategic technology management," *World Pat. Inf.*, vol. 25, no. 3, pp. 233–242, Sep. 2003.

[9] X. Li, Q. Xie, T. Daim, and L. Huang, "Forecasting technology trends using text mining of the gaps between science and technology: The case of perovskite solar cell technology," *Technol. Forecasting Social Change*, vol. 146, pp. 432–449, Sep. 2019.

[10] R. Popper, "How are foresight methods selected?" *Foresight*, vol. 10, no. 6, pp. 62–89, Oct. 2008.

[11] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.

[12] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. Bowman, "GLUE: A multi-task benchmark and analysis platform for natural language understanding," in *Proc. EMNLP Workshop BlackboxNLP, Analyzing Interpreting Neural Netw.*, Brussels, Belgium, 2018, pp. 353–355. [Online]. Available: https://aclanthology.org/W18-5446/

[13] K. Blagec, A. Barbosa-Silva, S. Ott, and M. Samwald, "A curated, ontology-based, large-scale knowledge graph of artificial intelligence tasks and benchmarks," *Scientific Data*, vol. 9, no. 1, p. 322, Jun. 2022.

[14] M. B. Massri, B. Spahiu, M. Grobelnik, V. Alexiev, M. Palmonari, and D. Roman, "Towards InnoGraph: A knowledge graph for AI innovation," in *Proc. ACM Web Conf. Companion*, New York, NY, USA, Apr. 2023, pp. 843–849, doi: 10.1145/3543873.3587614.

[15] A. Meroño-Peñuela, E. Simperl, A. Kurteva, and I. Reklos, "KG.GOV: Knowledge graphs as the backbone of data governance in AI," *J. Web Semantics*, vol. 85, May 2025, Art. no. 100847.

[16] I. Tiddi and S. Schlobach, "Knowledge graphs as tools for explainable machine learning: A survey," *Artif. Intell.*, vol. 302, Jan. 2022, Art. no. 103627. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0004370221001788

[17] L. Li, W. Xu, J. Guo, R. Zhao, X. Li, Y. Yuan, B. Zhang, Y. Jiang, Y. Xin, R. Dang, D. Zhao, Y. Rong, T. Feng, and L. Bing, "Chain of ideas: Revolutionizing research via novel idea development with LLM agents," 2024, *arXiv:2410.13185*.

[18] C. Peng, F. Xia, M. Naseriparsa, and F. Osborne, "Knowledge graphs: Opportunities and challenges," *Artif. Intell. Rev.*, vol. 56, no. 11, pp. 13071–13102, Nov. 2023.

[19] X. Dong, E. Gabrilovich, G. Heitz, W. Horn, N. Lao, K. Murphy, T. Strohmann, S. Sun, and W. Zhang, "Knowledge vault: A Web-scale approach to probabilistic knowledge fusion," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2014, pp. 601–610.

[20] DBpedia. (2024). *DBpedia Project*. Accessed: Apr. 15, 2025. [Online]. Available: https://www.dbpedia.org

[21] J. Lehmann, R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P. N. Mendes, S. Hellmann, M. Morsey, P. van Kleef, S. Auer, and C. Bizer, "DBpedia—A large-scale, multilingual knowledge base extracted from Wikipedia," *Semantic Web*, vol. 6, no. 2, pp. 167–195, 2015.

[22] Princeton Univ. (2024). *WordNet: A Lexical Database for English*. Accessed: Apr. 15, 2025. [Online]. Available: https://wordnet.princeton.edu

[23] Wikidata. (2024). *Wikidata: A Free and Open Knowledge Base*. Accessed: Apr. 15, 2025. [Online]. Available: https://www.wikidata.org/wiki/Wikidata

[24] Wikiprompts. (2024). *Wikiprompts Knowledge Graph*. Accessed: Apr. 15, 2025. [Online]. Available: https://wikiprompts.wikibase.cloud/wiki/Main_Page

[25] OpenBioLink. (2024). *Intelligence Task Ontology (ITO)*. Accessed: Apr. 15, 2025. [Online]. Available: https://github.com/OpenBioLink/ITO#

[26] ConceptNet. (2024). *ConceptNet Semantic Network*. Accessed: Apr. 15, 2025. [Online]. Available: https://conceptnet.io

[27] R. Speer and J. Lowry-Duda, "ConceptNet at SemEval-2017 task 2: Extending word embeddings with multilingual relational knowledge," in *Proc. 11th Int. Workshop Semantic Eval.*, 2017, pp. 85–89, doi: 10.18653/v1/s17-2008.

[28] (2024). *Open Research Knowledge Graph (ORKG)*. Accessed: Apr. 15, 2025. [Online]. Available: https://orkg.org

[29] S. Pan, L. Luo, Y. Wang, C. Chen, J. Wang, and X. Wu, "Unifying large language models and knowledge graphs: A roadmap," *IEEE Trans. Knowl. Data Eng.*, vol. 36, no. 7, pp. 3580–3599, Jul. 2024, doi: 10.1109/TKDE.2024.3352100.

[30] X. Gao, Z. Zhang, M. Xie, T. Liu, and Y. Fu, "Graph of AI ideas: Leveraging knowledge graphs and LLMs for AI research idea generation," 2025, *arXiv:2503.08549*.

[31] P. Schönhofen, "Annotating documents by Wikipedia concepts," in *Proc. IEEE/WIC/ACM Int. Conf. Web Intell. Intell. Agent Technol.*, Ljubljana, Slovenia, Dec. 2008, pp. 461–467.

[32] R. Avogadro, M. Ciavotta, F. De Paoli, M. Palmonari, and D. Roman, "Estimating link confidence for human-in-the-loop table annotation," in *Proc. IEEE Int. Conf. Web Intell. Intell. Agent Technol. (WI-IAT)*, Oct. 2023, pp. 142–149.

[33] R. Avogadro, F. D'Adda, and M. Cremaschi, "Feature/vector entity retrieval and disambiguation techniques to create a supervised and unsupervised semantic table interpretation approach," *Knowl.-Based Syst.*, vol. 304, Nov. 2024, Art. no. 112447.

[34] T. Zhang, X. Yue, Y. Li, and H. Sun, "TableLlama: Towards open large generalist models for tables," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Human Lang. Technol.*, 2024, pp. 6024–6044. [Online]. Available: https://aclanthology.org/2024.naacl-long.335/

[35] OpenAI. (2024). *GPT-4o Mini: Advancing Cost-Efficient Intelligence*. Accessed: Apr. 7, 2025. [Online]. Available: https://openai.com/index/gpt-4o-mini-advancing-costv-efficient-intelligence/

[36] V. Alexiev, B. Bechev, and A. Osytsin. (2023). *The InnoGraph Artificial Intelligence Taxonomy: A Key to Unlocking AI-Related Entities and Content*. Ontotext Corp. [Online]. Available: https://www.ontotext.com/knowledgehub/white_paper/the-innograph-artificial-intelligence-taxonomy/

[37] A. Tagarev, L. Tolosi, and V. Alexiev, "Domain-specific modeling: A food and drink gazetteer," in *Transactions on Computational Collective Intelligence XXVI*, vol. 10190, N. T. Nguyen, R. Kowalczyk, A. M. Pinto, and J. Cardoso, Eds., Cham, Switzerland: Springer, Jul. 2017, pp. 186–209. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-319-59268-8_9

[38] (2012). *ACM Computing Classification System*. Accessed: Apr. 15, 2025. [Online]. Available: https://www.acm.org/publications/class-2012

[39] S. Angioni, A. A. Salatino, F. Osborne, D. R. Recupero, and E. Motta, "AIDA: A knowledge graph about research dynamics in academia and industry," *Quantum Sci. Stud.*, vol. 2, no. 4, pp. 1356–1398, Jan. 2021.

[40] J. Tang, J. Zhang, L. Yao, J. Li, L. Zhang, and Z. Su, "ArnetMiner: Extraction and mining of academic social networks," in *Proc. 14th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2008, pp. 990–998, doi: 10.1145/1401890.1402008.

[41] A. B. Statistics. (2020). *Australian and New Zealand Standard Research Classification (ANZSRC)*. Accessed: Apr. 15, 2025. [Online]. Available: https://www.abs.gov.au/statistics/classifications/australian-and-new-zealand-standard-research-classification-anzsrc/latest-release

[42] (2025). *ArXiv Subject Classifications*. Accessed: Apr. 15, 2025. [Online]. Available: https://arxiv.org/category_taxonomy

[43] (2025). *National Natural Science Foundation of China (NSFC) Classification*. Accessed: Apr. 15, 2025. [Online]. Available: https://www.nsfc.gov.cn/english/site_1/index.html

[44] A. A. Salatino, T. Thanapalasingam, A. Mannocci, A. Birukou, F. Osborne, and E. Motta, "The computer science ontology: A comprehensive automatically-generated taxonomy of research areas," *Data Intell.*, vol. 2, no. 3, pp. 379–416, Dec. 2019.

[45] Amer. Econ. Assoc. (2025). *JEL Classification Codes Guide*. Accessed: Apr. 15, 2025. [Online]. Available: https://www.aeaweb.org/econlit/jelCodes.php

[46] Publications Office Eur. Union. (2025). *EuroSciVoc: The European Science Vocabulary*. Accessed: Apr. 15, 2025. [Online]. Available: https://op.europa.eu/en/web/eu-vocabularies/euroscivoc

[47] U.S. Nat. Library Med. (2025). *Medical Subject Headings (MeSH)*. Accessed: Apr. 15, 2025. [Online]. Available: https://www.nlm.nih.gov/mesh/meshhome.html

[48] A. M. Soc. (2020). *Mathematics Subject Classification 2020*. Accessed: Apr. 15, 2025. [Online]. Available: https://msc2020.org/

[49] P. Panov, L. N. Soldatova, and S. Deroski, "OntoDM: An ontology of data mining," in *Proc. 12th Int. Conf. Discovery Sci.* Cham, Switzerland: Springer, 2008, pp. 257–270.
[50] J. Priem, H. Piwowar, and R. Orr, "OpenAlex: A fully-open index of scholarly works, authors, venues, institutions, and concepts," 2022, *arXiv:2205.01833*.
[51] Springer Nature. (2025). *SciGraph: Springer Nature's Linked Open Data Platform*. Accessed: Apr. 15, 2025. [Online]. Available: https://communities.springernature.com/users/82895-sn-scigraph
[52] (2025). *Semantic Scholar*. Accessed: Apr. 15, 2025. [Online]. Available: https://www.semanticscholar.org
[53] (2025). *StackExchange Tags*. Accessed: Apr. 15, 2025. [Online]. Available: https://stackexchange.com
[54] Eur. Pat. Office United States Pat. Trademark Office. (2025). *Cooperative Patent Classification (CPC)*. Accessed: Apr. 15, 2025. [Online]. Available: https://www.cooperativepatentclassification.org
[55] M. Färber and D. Lamprecht, "Linked papers with code: The latest in machine learning as an RDF knowledge graph," 2023, *arXiv:2310.20475*.
[56] (2025). *Crunchbase Categories*. Accessed: Apr. 15, 2025. [Online]. Available: https://www.crunchbase.com/discover/categories
[57] N. Golyandina, V. Nekrutkin, and A. A. Zhigljavsky, *Analysis of Time Series Structure: SSA and Related Techniques*. Boca Raton, FL, USA: CRC Press, 2001.
[58] G. E. P. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time Series Analysis: Forecasting and Control*, 5th ed., Hoboken, NJ, USA: Wiley, 2015.
[59] K. Sharma, R. Bhalla, and G. Ganesan, "Time series forecasting using fb-prophet," in *Proc. ACM*, vol. 3445, R. Zwiggelaar, G. Ganesan, Q. S. Cheng, K.-L. Du, and S. R. Satti, Eds., 2022, pp. 59–65. [Online]. Available: http://dblp.uni-trier.de/db/conf/acm2/acm2022.html#SharmaBG22
[60] R. Avogadro, M. Cremaschi, F. Dadda, F. De Paoli, and M. Palmonari, "LamAPI: A comprehensive tool for string-based entity retrieval with type-base filters," in *Proc. 17th ISWC Workshop Ontol. Matching (OM)*, 2022, pp. 25–36.
[61] F. Belotti, F. Dadda, M. Cremaschi, R. Avogadro, and M. Palmonari, "Evaluating LLMs on entity disambiguation in tables," 2024, *arXiv:2408.06423*.

INNA NOVALIJA received the M.A. degree in economics and the Ph.D. degree in computer science, focused on ontology extension using text mining from Central European University. She is currently a Postdoctoral Researcher and an Expert Advisor with the Artificial Intelligence Laboratory, Jožef Stefan Institute, Slovenia. Her research interests include semantic technologies, text analytics, and knowledge graph construction, with contributions to EU-funded projects, such as enRichMyData, Odeuropa, and DATABENCH.

DUMITRU ROMAN received the Ph.D. degree from the University of Innsbruck, Austria. He is currently a Researcher in the general area of data management. He has wide experience with initiating and leading large-scale projects in the area of data management and analytics.

FEDERICO BELOTTI is currently a Research Fellow with the Department of Informatics, Systems, and Communication, University of Milan–Bicocca. Previously, he was a Research Engineer with Orobix S.r.l., where he developed innovative methods in computer vision and reinforcement learning. Recently, his work has expanded to include reasoning models. His research interests include natural language processing and artificial intelligence, with a focus on enhancing table understanding through large language models (LLMs) and advancing model interpretability.

VLADIMIR ALEXIEV received the master's degree from the Technical University of Sofia, in 1991, and the Ph.D. degree from the University of Alberta, in 1999. He worked on semantic technologies at Graphwise/Ontotext, since 2010, as the Chief Data Architect. He focuses on GenAI and LLMs, semantic industrial data, dataspaces, asset administration shell, and digital product passports.

LUIS REI received the Ph.D. degree in artificial intelligence. He is currently a Researcher with the Department for Artificial Intelligence, Jožef Stefan Institute, Slovenia. He was previously at the University of Porto, where he worked on various topics in artificial intelligence and robotics. His research focuses on natural language understanding and machine learning, with applications across diverse domains.

ROBERTO AVOGADRO received the Ph.D. degree in computer science from the University of Milano–Bicocca. At Smart Data Group, he is currently working on the data linking problem. He is a Researcher at SINTEF Digital in the Smart Data Group. He is focusing on how to apply an AI solution for linking problems.

BABAK KHALILVANDIAN is currently pursuing the joint degree in data science with the Università degli Studi di Milano–Bicocca and in computer engineering with the University of Tabriz. With experience as a Freelancer, since 2021, he has worked on diverse projects in data science, software development, and psychology task implementation.

BOYAN BECHEV received the master's degree in computer science from the National School of Computer Science and Applied Mathematics of Grenoble, with a focus on distributed computing. He is employed at Graphwise/Ontotext where he progressed from Junior to Full Data Engineer, with a focus on knowledge graphs and data infrastructure.

CATALINA ALEXANDRA CHINIE is currently an Associate Professor and a Researcher at Bucharest University of Economic Studies. She has a wide experience as both a member and a coordinator in research projects and market studies. She coordinates the IDEAS Research Center within Bucharest University of Economic Studies. Her main research interest includes digital strategy.

IULIA CIUREA is currently pursuing the Ph.D. degree with the Department of Business Informatics and Statistics, Bucharest University of Economic Studies. She is a Researcher with the Department of Business Informatics and Statistics, Bucharest University of Economic Studies. Her main research interests include digital transformation and emerging technologies.

JANEZ BRANK is currently a Researcher with the Artificial Intelligence Laboratory, Jožef Stefan Institute. He has worked on topics, such as document classification, ontology evolution, clustering of document streams, and the identification of concepts in text. He is the author of Wikifier (wikifier.ijs.si).

COSMIN UDROIU is currently a Technical Project Manager at CS GROUP–ROMANIA, with over 23 years of experience in aerospace and Earth observation systems. He has led multiple ESA-funded projects, including Sen4Stat, Sen4CAP, and Sentinel2Agriculture, focusing on agricultural monitoring and policy support through satellite data processing. He has a strong background in software architecture and design, distributed systems, requirements analysis, design patterns, and project management using traditional and agile methodologies.

AHMET SOYLU received the Ph.D. degree in informatics from KU Leuven, Belgium, in 2012. He is currently the Dean of the School of Doctoral Studies and a Full Professor of computer science with the Kristiania University of Applied Sciences. He also has an adjunct position with the University of Oslo (UiO). His research interests include data management, the semantic web, big data, and cloud computing.

MATTEO PALMONARI is currently an Associate Professor with the Department of Informatics, Systems, and Communication, University of Milan–Bicocca. His research interests include data management and artificial intelligence, with a focus on knowledge graphs, data enrichment, and natural language processing.