DOCTORAL SCHOOL
UNIVERSITY OF MILANO-BICOCCA

Department of **Informatics, Systems and Communication**

Ph. D. program in **Computer Science**, **XXXVIII** cycle

Entity-Oriented Strategies for Information Extraction and Access in Knowledge-Intensive Domains

Riccardo Pozzi
Student Number 807857

Supervisor: **Prof. Matteo Palmonari**
Ph. D. Tutor: **Prof. Claudio Zandron**
Ph. D. Coordinator: **Prof. Leonardo Mariani**

Academic Year **2024–2025**

Abstract

Knowledge-intensive domains such as law require accessing, integrating, and reasoning over large collections of heterogeneous documents, while meeting strict requirements on privacy, traceability, and regulatory compliance. Despite recent advances in large language models, their direct use in these settings is limited by hallucination, lack of grounding, and the legal constraints that complicate the transfer of sensitive data to external APIs. This thesis investigates how to satisfy information access use cases in the legal domain, including precedent retrieval, investigative search on seized data, document navigation, question answering, and statistical monitoring, under these constraints.

The work pursues three objectives. First, it quantifies to what extent general-domain entity extraction pipelines, including entity recognition, entity linking, and NIL prediction, can be applied to Italian legal judgments and investigative chat logs. The results show that incremental entity extraction, where novel (or NIL) entities are identified and added to the knowledge base, suffers from error propagation, and that detecting novel entities (NIL prediction) is a major performance bottleneck, supporting the need for architectures that tolerate imperfect extraction. Second, it designs an entity-centric data integration architecture that integrates heterogeneous legal sources (judgments, investigative chats, attachments) around entities, supports traceability and human oversight via error correction functionalities, remains useful despite extraction errors, and enables the considered use cases. Third, it develops ReFactX, a constrained-generation approach to question answering that injects facts from large knowledge bases into a large language model without retrievers or external calls, producing answers that are traceable and verifiable against grounded evidence while adding only negligible latency, thus remaining efficient and suitable for local deployment.

Together, the contributions represent an integrated approach for information access in knowledge-intensive legal settings. The designed entity-centric data integration architecture integrates the knowledge received from extraction services and can be paired with user interfaces or ReFactX to support downstream use cases, while preserving traceability, verifiability, and error-correction capability in line with the GDPR and AI Act requirements on data control and human oversight.

List of publications

- Riccardo Pozzi, Federico Moiraghi, Fausto Lodi, and Matteo Palmonari. "Evaluation of Incremental Entity Extraction with Background Knowledge and Entity Linking". In: *Proceedings of the 11th International Joint Conference on Knowledge Graphs*. IJCKG '22. Hangzhou, China: Association for Computing Machinery, 2023, pp. 30–38. ISBN: 9781450399876. DOI: 10.1145/3579051.3579063. URL: https://doi.org/10.1145/3579051.3579063

- Mattia Marzocchi, Marco Cremaschi, Riccardo Pozzi, Roberto Avogadro, and Matteo Palmonari. "MammoTab: A Giant and Comprehensive Dataset for Semantic Table Interpretation". In: *SemTab@ISWC*. 2022, pp. 28–33. URL: https://ceur-ws.org/Vol-3320/paper3.pdf

- Valerio Bellandi, Christian Bernasconi, Fausto Lodi, Matteo Palmonari, Riccardo Pozzi, Marco Ripamonti, and Stefano Siccardi. "An entity-centric approach to manage court judgments based on Natural Language Processing". In: *Computer Law & Security Review* 52 (2024). All authors contributed equally, p. 105904. ISSN: 0267-3649. DOI: https://doi.org/10.1016/j.clsr.2023.105904. URL: https://www.sciencedirect.com/science/article/pii/S0267364923001140

- Riccardo Pozzi, Riccardo Rubini, Christian Bernasconi, and Matteo Palmonari. "Named Entity Recognition and Linking for Entity Extraction from Italian Civil Judgements". In: *AIxIA 2023 – Advances in Artificial Intelligence*. Cham: Springer Nature Switzerland, 2023, pp. 187–201. ISBN: 978-3-031-47546-7. DOI: 10.1007/978-3-031-47546-7_13. URL: https://link.springer.com/content/pdf/10.1007/978-3-031-47546-7.pdf

- Riccardo Pozzi, Valentina Barbera, Renzo Alva Principe, Davide Giardini, Riccardo Rubini, and Matteo Palmonari. "Combining Knowledge Graphs and NLP to Analyze Instant Messaging Data in Criminal Investigations". In: *Web Information Systems Engineering – WISE 2024*. Springer Nature Singapore, 2025, pp. 427–442. ISBN: 978-981-96-0567-5. DOI: 10.1007/978-981-96-0567-5_30. arXiv: 2509.26487 [cs.AI]. URL: https://link.springer.com/content/pdf/10.1007/978-981-96-0567-5.pdf

- Ruben Agazzi, Renzo Alva Principe, Riccardo Pozzi, Marco Ripamonti, and Matteo Palmonari. "DAVE: A Framework for Assisted Analysis of Document Collections in Knowledge-Intensive Domains". In: *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, IJCAI-25*. Demo Track. International Joint Conferences on Artificial Intelligence Organization, Aug. 2025, pp. 10984–10988. DOI: 10.24963/ijcai.2025/1246. URL: https://doi.org/10.24963/ijcai.2025/1246

- Riccardo Pozzi, Matteo Palmonari, Andrea Coletta, Luigi Bellomarini, Jens Lehmann, and Sahar Vahdati. "ReFactX: Scalable Reasoning with Reliable Facts via Constrained Generation". In: *The Semantic Web – ISWC 2025*. Cham: Springer Nature Switzerland, 2026, pp. 290–308. ISBN: 978-3-032-09527-5. DOI: 10.1007/978-3-032-09527-5_16. URL: https://doi.org/10.1007/978-3-032-09527-5_16

Contributions to conferences

As a **speaker** I presented:

- Ref. [272] at the 11th International Joint Conference on Knowledge Graphs (IJCKG 2022).
- Ref. [274] at the 22nd International Conference of the Italian Association for Artificial Intelligence (AIxIA 2023).
- Ref. [271] at the 25th International Conference on Web Information Systems Engineering (WISE 2024).
- Ref. [273] at the 24th International Semantic Web Conference (ISWC 2025).

As a **poster presenter**:

- Poster based on Ref. [273] at the Lectures on Computational Linguistics 2025, organized by the Italian Association of Computational Linguistics (AILC).

As an **author**:

- Ref. [272] in the Proceedings of the 11th International Joint Conference on Knowledge Graphs (IJCKG 2022).
- Ref. [224] in the Proceedings of the Fourth edition of the Semantic Web Challenge on Tabular Data to Knowledge Graph Matching (SemTab 2022).
- Ref. [274] in the Proceedings of the 22nd International Conference of the Italian Association for Artificial Intelligence (AIxIA 2023).
- Ref. [271] in the Proceedings of the 25th International Conference on Web Information Systems Engineering (WISE 2024).
- Ref. [4] in the Proceedings of the 34th International Joint Conference on Artificial Intelligence (IJCAI 2025), Demo Track.
- Ref. [273] in the Proceedings of the 24th International Semantic Web Conference (ISWC 2025).

Contents

List of Figures

List of Tables

List of Acronyms

AD annotation database

AI artificial intelligence

ANN approximate nearest neighbor

API application programming interface

ASR automatic speech recognition

BgKR background knowledge repository

CLM causal language model

CoT chain-of-thoughts

CRUD create, read, update and delete

DAVE Document Assistant for Validation and Exploration

DB database

DBMS database management system

DIA data integration architecture

DL deep learning

EE entity extraction

EL entity linking

ER entity recognition

GNN graph neural network

HITL human-in-the-loop

IA information access

IAA inter-annotator agreement

ICL in-context learning

IE information extraction

IMA instant messaging application

IncEL incremental entity linking

IR information retrieval

IRI internationalized resource identifier

KB knowledge base

KBP knowledge base population

KC knowledge consolidation

KG knowledge graph

KGQA knowledge-graph question answering

KR knowledge repository

KRMS knowledge repository management system

LLM large language model

LM language model

LRM large reasoning model

ML machine learning

MLM masked language model

NewKR new knowledge repository

NLP natural language processing

POS part-of-speech

QA question answering

QL query language

RAG retrieval augmented generation

RDF Resource Description Framework

RI retrieval indexer

RLHF reinforcement learning from human feedback

STA semantic text annotation

SW semantic web

UI user interface

UnIOF uniform input-output format

URI uniform resource identifier

Acknowledgements

First and foremost, I would like to express my deepest gratitude to my supervisor, Prof. Matteo Palmonari, for his constant guidance, trust, and invaluable advice throughout these years, and for the kindness and understanding that made working with him a truly enriching experience.

I am also very grateful to Prof. Sahar Vahdati for hosting me during my secondment in Dresden and for her scientific insights, encouragement, and support, as well as for the warm and friendly collaboration we shared.

I would also like to thank my tutor, Prof. Claudio Zandron, for his support during my doctoral studies.

My heartfelt thanks go to my family, for their unconditional love and patience, and to my girlfriend Cecilia, for her love, support, and encouragement in the most demanding moments of this journey.

These years have been an intense and transformative experience, both professionally and personally. I am thankful to all the colleagues and friends who shared this path with me—from the Ph.D. fellows, researchers, and academics across the entire department to those I met during my secondment. Their company, discussions, and collaboration have made this period not only a time of growth but also of genuine enjoyment and discovery.

Chapter 1

Introduction

1.1 Introduction

In recent years, natural language processing (NLP), the field studying automatic processing and understanding of natural language, has been revolutionized by the transformer architecture [361]. Introduced in 2017, transformers improved both performance and training-time parallelization compared to previous approaches [361], enabling researchers to train bigger models on huge natural language corpora, for later "transferring" the generic linguistic knowledge acquired to specific downstream tasks [172, 90, 283, 285], such as question answering, which responds to questions in natural language [28], or entity recognition, which identifies entities like persons or locations in text [24, 179, 172].

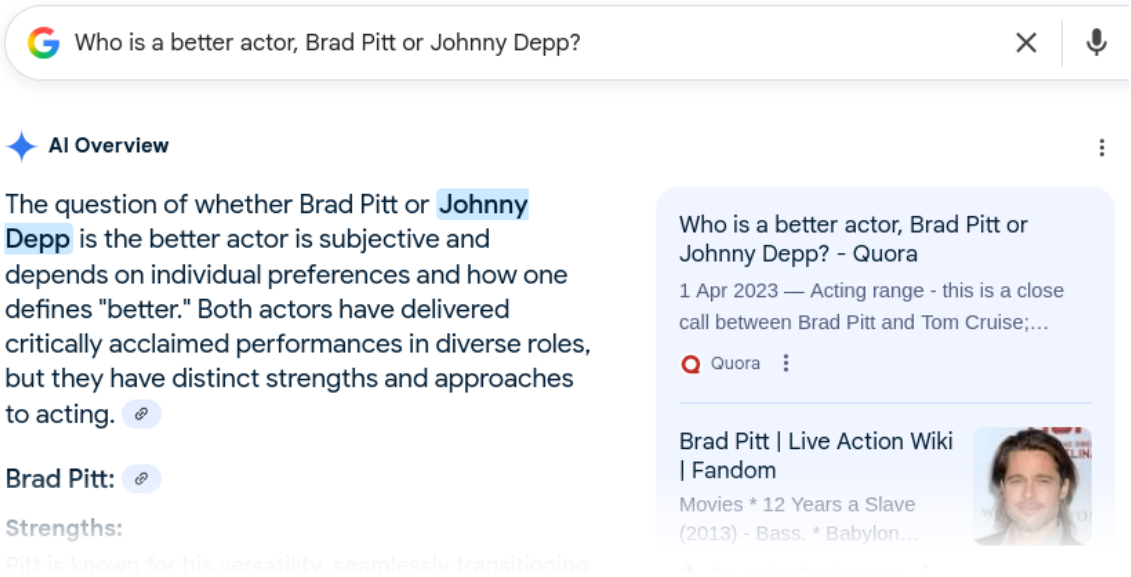


Figure 1.1: Example of question answered with a large language model (screenshot from Google Search).

Later, transformer-based language models (LMs) have been scaled to billions of parameters, demonstrating impressive performance even without fine-tuning [54]. For example, GPT3, which has been released in 2020, counted 175 billion parameters. The real revolution, however, came in 2022 with *instruction tuning* [257] that allowed the alignment of large language models (LLMs) to users' intent, improving their capability of following instructions, reducing undesired behaviors, including bias, toxicity, and other harmful outputs [257]. In the same year, OpenAI released ChatGPT [252], bringing the advances of LLMs to the general public through a chat-based user interface (UI), and providing APIs that enabled researchers, practitioners, and businesses to study the capabilities of instruction-tuned models, and integrate them into applications. Since its launch, ChatGPT's user base has grown rapidly, reaching 800 million users per week, approximately 10% of the world's population [334]. Today, even popular search engines, such as Google Search[1] have integrated LLMs to provide quick answers or overviews of the search results [296], as visible in Figure 1.1.

Despite their usefulness, LLMs introduce non-negligible risks when deployed without safeguards, such as *misuse* and *misinformation* [34, 374]. They have been used as sources of factual knowledge despite being prone to *hallucination* [226, 167]—i.e., the generation of incoherent, nonsensical, or unfaithful content. This risk is not merely hypothetical: in 2023, invented case law generated by ChatGPT was submitted and presented as authentic judicial decisions, an episode that resulted in formal sanctions by the court [232].

A second risk concerns environmental impact: modern LLMs require substantial computation for both training and inference, leading to high energy consumption and a significant carbon footprint [339, 261]. These costs are largely driven by the scale of contemporary models.

Indeed, state-of-the-art LLMs count hundreds of billions of parameters—for instance, DeepSeek-V3 [87] comprises 671 billion parameters[2]—and are therefore generally served as APIs, as their size entails prohibitive hardware requirements for local deployment.

Also, in sensitive domains such as law or healthcare, documents often contain personal data that cannot be transferred to external servers. Using API-based LLMs raises compliance issues under the European *General Data Protection Regulation (GDPR, Regulation (EU) 2016/679)* [107] and *Artificial Intelligence Act (Regulation (EU) 2024/1689)* [106]. The GDPR [107] imposes strict requirements on lawful and transparent data processing, while the AI Act [106] classifies legal and law enforcement applications as high-risk, requiring traceability, security, and human oversight. These obligations are difficult to meet when processing occurs on third-party infrastructures. Conversely, local deployment allows institutions to retain control of the data, simplifying compliance and safeguarding confidentiality.

Besides being sensitive in terms of data protection and confidentiality, the legal and healthcare sectors are also domains that require specialized expertise, deep understanding, and extensive information processing. Decisions in these fields rely heavily on expert knowledge and the careful interpretation of complex information sources. Another example of such a domain is finance, where similar forms of domain-specific expertise and reasoning are essential [249, 58, 329]. In the remainder of this work, we refer to these as *knowledge-intensive domains*, encompassing areas in which effective decision-making depends on the ability to access, interpret, and integrate heterogeneous and highly specialized information sources.

Several approaches based on natural language processings (NLPs) techniques have been developed to support and enhance information workflows in knowledge-intensive domains. In these

[1] https://www.google.com/
[2] https://huggingface.co/deepseek-ai/DeepSeek-V3

areas, professionals often face the challenge of locating, interpreting, and synthesizing large volumes of domain-specific information to make informed decisions. For instance, the need to retrieve precedent cases in law or identify relevant clinical trials in healthcare has driven the development of domain-specific information retrieval (IR) systems [233, 52, 143, 301]. Other works have focused on automatic summarization to handle lengthy and complex documents [332, 181], or on predictive modeling—such as drug response prediction in healthcare [327] and civil case outcome prediction in the legal domain [125]. In finance, NLP methods have been applied to the analysis of financial news, corporate announcements, and social media posts, supporting downstream tasks such as financial sentiment analysis and market forecasting [97].

Even though different knowledge-intensive domains have their own specific characteristics, certain recurring features can be observed across them. These include the central roles of entities—such as parties, judges, and lawyers in the legal domain; patients, doctors, and clinical staff in healthcare; and companies, investors, and regulators in finance—as well as the reliance on heterogeneous information sources. In the legal domain, these sources include court documents, such as civil court judgments, notary documents, or other contracts [134, 10, 55]. In the context of criminal investigations, it might be necessary to analyze chat messages extracted from suspects' mobile phone [341, 267, 176, 362], bank statements [414], emails, or posts on social media platforms [163]. In healthcare, relevant sources consist of laboratory reports [94], clinical trials, electronic health records, medical imaging, and data from health monitoring devices [389], while financial professionals consult financial news, statements, and market data [97].

1.2 Legal Domain and Use Cases

In the rest of this thesis, we focus on different use cases within the legal domain, a broad and knowledge-intensive area that shares many characteristics with other sensitive contexts such as healthcare. Specifically, we examine civil trials and criminal investigations through two representative types of sources in Italian: civil court judgments and investigative chat logs.

Civil court judgments are formal acts issued by the judge at the conclusion of a civil proceeding, constituting the official expression of the court's decision. They typically identify the parties involved, outline the relevant facts and procedural background, and present the legal reasoning supporting the decision, followed by the operative part specifying its effects. Their structure and mandatory elements are established by the Italian Code of Civil Procedure [293, Art. 132], which requires, among other things, the indication of the court and the parties, the statement of facts, and the reasoning in fact and law.

Investigative chat logs consist of messages extracted from instant messaging applications (IMAs), such as WhatsApp[3], from the mobile phones of suspects during criminal investigations. These logs may include textual conversations, exchanged multimedia files, and metadata [265]. In the forensic context, such data are carefully collected and analyzed to reconstruct events [265], identify relationships among participants [362], for gathering evidence in support of the proceeding [267].

While NLP has the potential to enhance information workflows in the legal domain [125], applying NLP techniques in this domain entails specific challenges, including regulatory constraints, evolving legislation, domain-specific language, and legal interpretations that differ from general usage. For instance, in tasks such as estimating the similarity between different legal proceedings, using general-domain methods for document similarity may incorrectly label two cases as similar,

[3] https://www.whatsapp.com/

even when the law treats them very differently. For example, the theft of similar products can be judged under different laws depending on the value of the stolen items [112], potentially leading to very different penalties.

The concept of similarity between legal cases also strongly depends on the outcome of the trials [309], which may be overlooked by general-domain NLP approaches. Moreover, this domain is characterized by a peculiar language that follows implicit structure, which may however vary depending on the country, on the court, or on the legal professional that is writing the document [309, 51]. The complexity of the legal language is further reflected in the nature of legal documents themselves, which are often lengthy and heterogeneous [309].

Other challenges include compliance with regulations, as described above, as well as the evolving nature of the legal framework. Statutes may change over time, meaning that cases previously considered similar may no longer be so under current legislation [112]. In addition, legal documents are structured into multiple sections serving distinct purposes [293, Art. 132], and the relevance of each section depends on the analytical task [112]. For example, when identifying similar cases, the comparison should focus on the factual and legal reasoning sections, which contain the elements that determine the applicability of legal provisions. In contrast, when searching for precedents involving the same individuals or organizations, the relevant information may appear primarily in the introductory sections describing the parties. Models that ignore this internal structure and treat the text as a single unstructured block risk misleading results—for instance, ranking two cases as similar simply because they mention the same persons, even though the underlying legal issues and outcomes differ substantially.

Despite these challenges, advances in artificial intelligence (AI) and natural language processing (NLP) have greatly expanded the potential for content analysis. Yet, in professional and institutional contexts, the ultimate goal is not analysis itself, but the ability to find, interpret, and use information to support decision-making. In such domains, AI technologies are most valuable when they enhance users' *information access* [132], understood here as the ability to effectively identify, retrieve, and utilize information. Over time, information access has evolved from keyword-based search to more interactive paradigms, such as conversational systems and LLMs that assist users in exploring and reasoning over complex data.

Considering the challenges discussed above, it becomes essential to examine how professionals in the legal and investigative domains actually access and use information in their daily work. To illustrate these needs, we identify a set of representative use cases of information access (IA), which highlight typical queries, document navigation, and analytical activities.

UC 1: Case retrieval and precedent search: Judges and lawyers may need to search for previous cases involving the same individuals or organizations [132], or analyze outcomes of similar cases to identify legal trends to ensure uniformity in the interpretation of the law [101].

UC 2: Document navigation and content exploration: When reviewing lengthy judgments, legal professionals benefit from tools that allow them to quickly identify specific information or sections [347, 131, 114], for example by locating all mentions of a given person.

UC 3: Investigative retrieval: Investigators may need to retrieve specific documents from a large corpus from seized devices [30], identify chat messages to understand real events [267]—such as to confirm a meeting between suspects—or quickly visualize the communication network of the individuals under investigation [362].

UC 4: Question answering on documents and collections: In the artificial intelligence (AI) era, when search engines can answer questions directly in natural language [296], legal professionals can benefit from question answering (QA) systems that provide answers on a single document or across document collections, along with references to the supporting evidence [134].

UC 5: Statistical analysis and monitoring: Authorities, such as the Ministry of Justice, may require aggregated statistics, e.g., the number of new cases per court per year or the average trial duration, to support performance monitoring, policy assessment, and resource planning [31].

These use cases reveal a recurrent pattern: professionals operate on large and heterogeneous corpora, leading to time-consuming [267] and often manual workflows [55], which slows down legal and investigative analysis. There is therefore a strong motivation to design information access methods and infrastructures that can accelerate these tasks—e.g., through entity-based faceted search, semantic document exploration, or question answering.

Several systems have been proposed to support legal professionals [10, 55, 51, 134, 132]. Early work focused on ontology-based document management, first for notary documents [10] and later for a broader range of acts and agreements by modeling document structure and associated entities [55]. Breit et al. [51] explored the automation of legal permit procedures. More recent systems include LegalAsst [134], which integrates LLMs, entity recognition, graph views, and decision-tree reasoning to assist courts, and the interactive faceted case-retrieval system of Guan et al. [132], which steers users toward precedent discovery.

On the investigative side, prior work has targeted illicit web data [177, 178] by building knowledge graphs (KGs) from domain ontologies and exposing faceted and map-based UIs. Pérez et al. [265] proposed a modular microservices architecture for heterogeneous investigative data, integrating components such as speech-to-text and keyword extraction to populate a KB. Other work addressed instant messaging application data [176, 267, 341, 362], mainly focusing on extraction pipelines and visual analytics (e.g. communication graphs, statistical dashboards, and maps), with occasional use of ER [341].

However, most investigative systems do not perform knowledge consolidation tasks, for example, determining which mentions of type `Person` refer to the same individual, or linking mentions to the corresponding entities in a knowledge repository (entity linking), such as Wikipedia [378]. As a result, information remains fragmented across documents, limiting cross-document integration; for example, consolidated knowledge, leveraging mention–entity links, allows users to identify other documents that mention the same person, thereby reducing manual effort.

1.3 Challenges and Objectives

The use cases introduced above highlight how AI and NLP can support professionals in fulfilling their information needs in the legal and investigative contexts. However, developing such systems remains far from trivial. In practice, several key challenges must be addressed to ensure that these technologies are effective, reliable, and legally compliant. The main challenges motivating this thesis can be summarized as follows:

Ch. 1: Scalability. Legal and investigative corpora may include thousands of heterogeneous documents, requiring methods that can efficiently process and integrate large volumes of text and metadata.

Ch. 2: Heterogeneity. The data vary widely in format and structure—from judgments and contracts to chat logs and structured evidence—which demands for flexible representations capable of integrating diverse sources within a unified framework.

Ch. 3: Traceability and verifiability. The outputs of AI-driven systems must remain interpretable and auditable: each extracted or inferred element should be traceable back to its source document, supporting transparency and human oversight [106, Art. 14–15].

Ch. 4: Integration of novel entities. Legal and investigative documents—as well as those from other sensitive domains such as healthcare—often mention entities that are not publicly known or documented in existing knowledge repository (e.g., Wikipedia or Wikidata). Handling such *novel* (or *NIL*) entities is therefore a key challenge, as it requires extending the repository to include new individuals emerging from the data, while maintaining consistency and interoperability with existing resources.

These challenges highlight the need for architectural solutions that can organize, connect, and make sense of heterogeneous data while preserving interpretability and compliance.

1.3.1 Research hypothesis

Across the diverse approaches for integrating legal data, reviewed in Section 3.2, a common principle emerges: the design of an intermediate architecture that mediates between users and the underlying sources of information. Such architectures may take the form of search engines indexing documents for efficient retrieval; visualization systems that highlight relevant elements in user interfaces [267, 265, 178, 347, 131, 114]; or data infrastructures such as knowledge graphs and databases [177, 178, 348, 10, 134, 31]. These systems share the goal of organizing and presenting information in a form that is meaningful to users and suited to their analytical tasks, acting as a bridge between information extraction and access and enabling the transition from unstructured data to semantically organized, queryable knowledge, as illustrated in Figure 1.2.

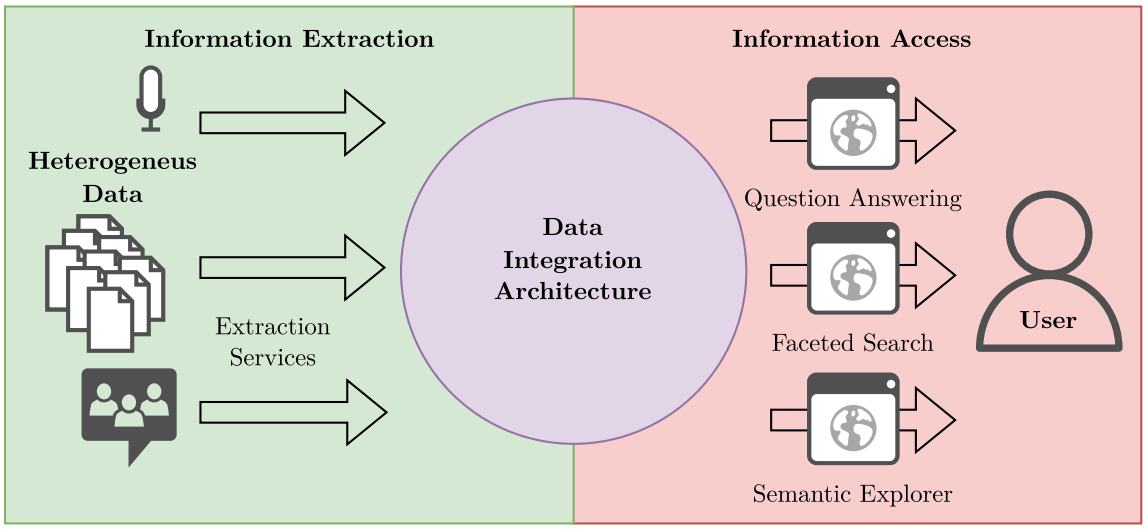


Figure 1.2: High-level overview of the architecture designed in this thesis (*Obj. 2*) for bridging information extraction and information access.

Building on this idea, this thesis is grounded on the hypothesis that *an entity-centric architecture can effectively integrate heterogeneous legal and investigative data, enabling advanced forms of information access and analysis while preserving scalability and traceability.* In particular, such an architecture would support the identified use cases in the legal and investigative domains. For instance, users may perform faceted search to filter relevant documents or messages based on entities such as persons, locations, or dates, ask questions in natural language to retrieve precise information, and even combine these capabilities—for example, applying question answering over a subset of documents filtered with faceted search.

Entities, especially after knowledge consolidation, serve as conceptual anchors that connect information both within individual documents and across multiple documents, enabling cross-references and semantic aggregation. They also enhance generalizability, as entities are common across domains: examples include persons, judges, and legal acts in law; patients, diseases, and treatments in healthcare; or companies, investors, and assets in finance. As illustrated in Figure 1.3, entities act as the connective tissue linking diverse data types—ranging from structured bank transfers to unstructured reports, voice messages, and chat transcripts—and support aggregation, such as computing counts or groupings over certain entities or entity types, as well as faceting, that is, organizing retrieved items along multiple dimensions to support interactive filtering and exploration [17, 141].

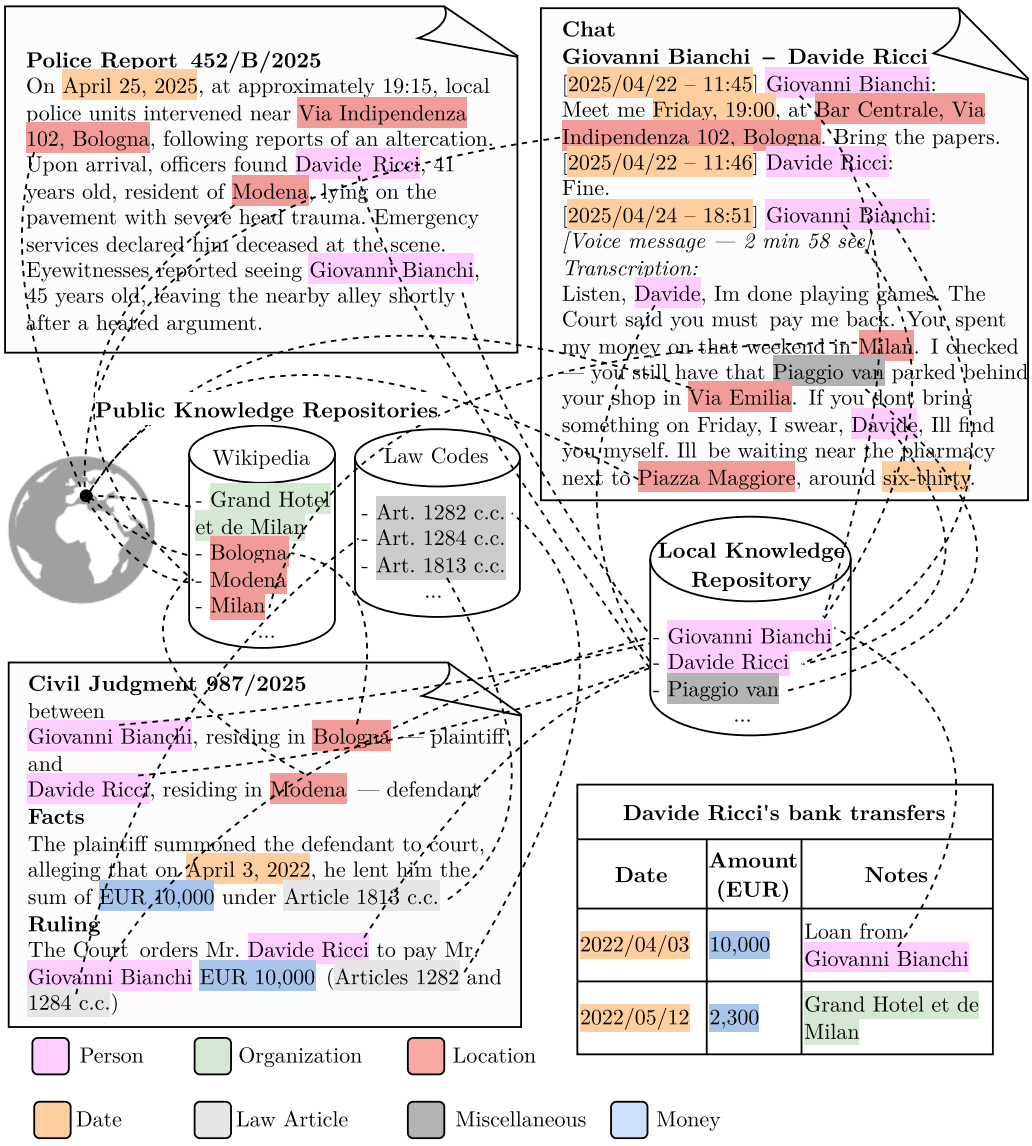


Figure 1.3: The central role of entities in interconnecting heterogeneous data. The example shows how a criminal report, a civil judgment, a chat report (including a voice message), and structured bank transfers can be linked through shared entities.

1.3.2 Research objectives and contributions

In response to the identified challenges and building on the hypothesis that an entity-centric architecture can integrate heterogeneous data and support advanced composable forms of information access, this thesis develops entity-oriented strategies that connect information extraction and access in knowledge-intensive domains, with a focus on the legal and investigative settings. These strategies are articulated through the following objectives:

Obj. 1: **Assessment of existing information extraction technologies considering legal domain challenges.** The first objective is to empirically evaluate the adaptability of existing information extraction components—including entity recognition, entity linking, and NIL prediction—to legal and investigative data, accounting for domain-specific constraints, linguistic heterogeneity, and the occurrence of novel entities (also called NIL entities).

This assessment provides quantitative evidence on domain adaptation needs and shows that handling NIL entities is prone to error propagation, with NIL prediction being a major source of errors. Existing entity extraction models can be applied to legal documents and investigative chats, but achieving satisfactory quality often requires in-domain fine-tuning or human-in-the-loop correction.

Obj. 2: **Design of an entity-centric architecture for data integration and information access.** The second objective is the design of an architecture that organizes heterogeneous and unstructured information around entities (e.g., persons, locations, organizations, or laws), bridging information extraction and access.

This architecture is designed to enable the development of applications for advanced use cases, such as those considered in this thesis. It also supports the combination of these capabilities—for example, performing question answering over a subset of documents filtered via faceted search.

Obj. 3: **Development of an efficient and scalable question answering system grounded in external knowledge.** The third objective is to design a question answering system capable of producing answers explicitly supported by a knowledge base, thereby enabling traceability and verifiability and mitigating hallucination risks. The system should scale to large collections and operate efficiently in local environments, in compliance with regulatory and confidentiality constraints.

To achieve these goals, the thesis introduces ReFactX, a constrained-generation approach that efficiently guides an LLM through a disk-backed prefix tree to produce valid facts directly from a large knowledge base, without the need for retrievers, complex pipelines, or architectural modifications to the underlying LLM.

This Ph.D. thesis is organized as follows. Chapter 2 introduces key concepts, including large language models and the Italian legal system. Chapter 3 reviews the related literature on information extraction, data-integration architectures for knowledge-intensive domains, and question answering. Chapter 4, Chapter 5, and Chapter 6 address the three research objectives, focusing respectively on entity extraction in the legal domain (*Obj. 1*), the design of an entity-centric data integration architecture (*Obj. 2*), and the development of a scalable and evidence-grounded question answering system (*Obj. 3*). Chapter 7 concludes the thesis and outlines directions for future work.

Chapter 2

Background

This chapter presents the background concepts necessary to understand the rest of the thesis, which are directly connected to the thesis objectives and use cases. Section 2.1 introduces information access and the notions of *information needs* and *information objects* [24], which are central to document navigation, question answering, and information retrieval scenarios in the legal and investigative domains (*UC 2*, *UC 4*, *UC 1*, and *UC 3*).

In Section 2.2, we define what constitutes an entity and describe how entity-based knowledge is represented in knowledge repositories and knowledge bases, since entities play a central role in information extraction (*Obj. 1*) and directly relate to design of an entity-centric architecture addressed by *Obj. 2*. Then, since information extraction and access are fundamentally based on natural language processing (NLP) techniques, Section 2.3 situates the discussion within recent advances in the field and outlines the core concepts underlying modern large language models (LLMs).

Once this technical context is established, Section 2.4 focuses on the application domain considered in this thesis, namely the Italian legal context. We describe the Italian legal system and the regulatory framework governing the use of artificial intelligence, together with illustrative examples of legal documents, the analysis of which has motivated several works produced during my Ph.D. [32, 274, 271]. In particular, we present an illustrative civil judgment and chat log from a criminal investigation, rendered in English for clarity but closely resembling the ones used in Italian courts and investigations.

2.1 Information Access

On a daily basis, we use our personal computer for accessing information. Search engines such as Google, Bing, Yahoo! and DuckDuckGo, as well as web browsers such as Mozilla Firefox, Google Chrome and Microsoft Edge, help us to fulfill our *information needs* by proving relevant *information objects* [24]. Search engines aim to find web pages that contain our information need, often represented as a *query*, web browser allow us to browse web pages to finally find the required information. Both are primary tools for facilitating *information access*. In this thesis, we adopt the following definition of information access, which is based on Encyclopedia.com [104] and limited to the digital activities a user undertakes to satisfy an information need. Broader considerations related to social, economic, or political factors that may restrict individuals' freedom to access information are beyond the scope of this work.

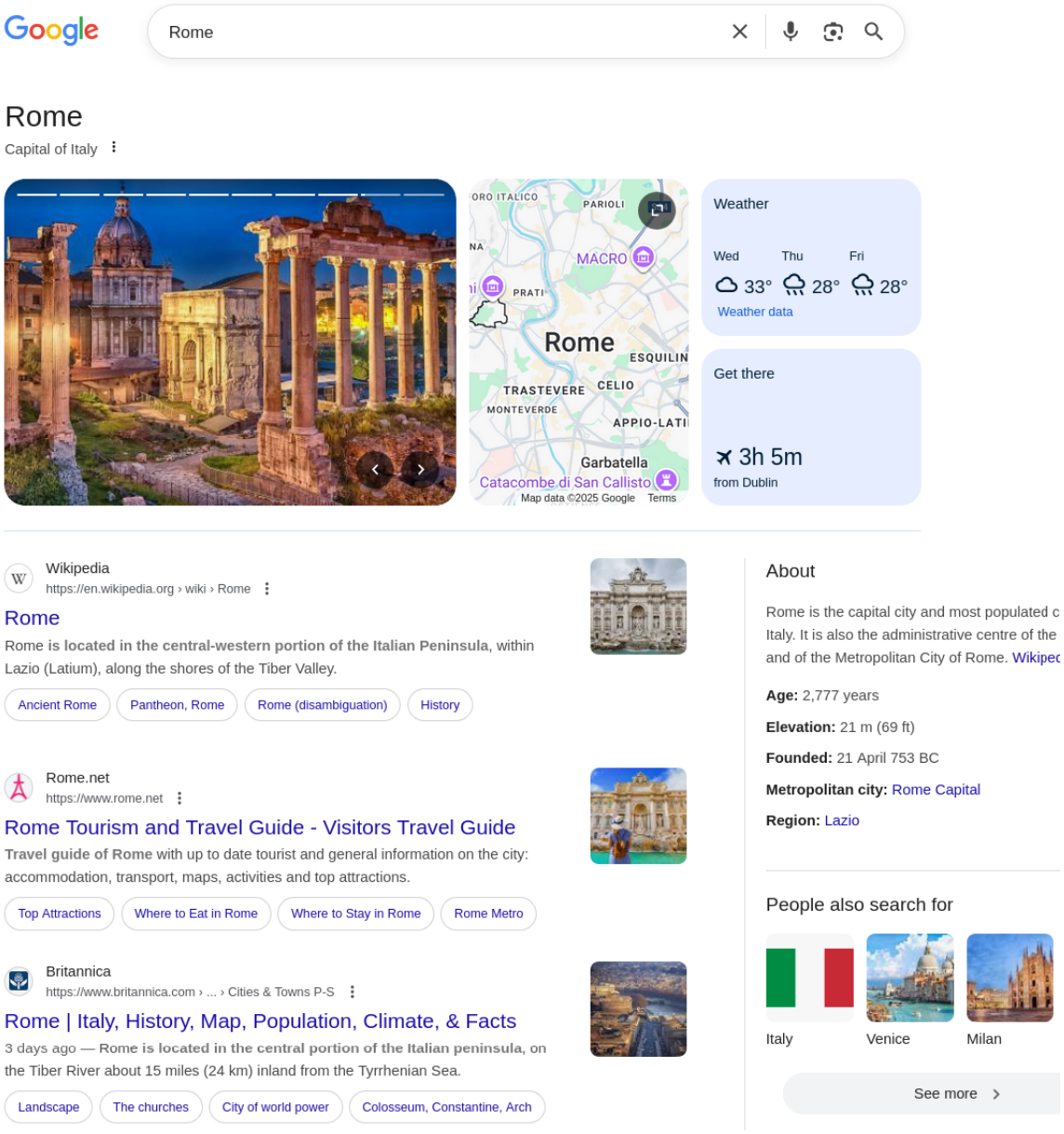


Figure 2.1: Example of a modern search engine result page (screenshot from Google Search).

Definition 1 (Information Access)**.** Information access is the ability to effectively identify, retrieve, and utilize information.

Recently, large language models (LLMs) have further extended the capabilities of search engines by answering natural language questions directly with fluent natural language responses, as illustrated in Figure 1.1. Moreover, these models can themselves be viewed as *information-seeking actors*: to fulfill an information need (eventually on behalf of a human user), they may actively invoke search engines or other tools to acquire information objects [254, 388, 268] and then convey the result to the user.

Applications for information access include, but are not limited to:

- Search engines that retrieve documents or items based on a query;
- Web browsers that allow users to explore documents and follow hyperlinks;
- Recommendation systems that suggest relevant or similar documents;
- Question answering systems that provide direct answers to users' questions based on information from structured or unstructured data.

While traditionally the information objects were documents, search engines have, over the past decades, gradually transitioned toward providing richer answers. As visible in Figures 2.1 and 2.2, results now often display entities and facts directly, enriched with pictures, maps, or other structured information depending on the answer type. A key enabler of these developments has been the emergence of large-scale knowledge bases and knowledge repositories, which organize information around entities [24]—such as the Google Knowledge Graph [333, 126].

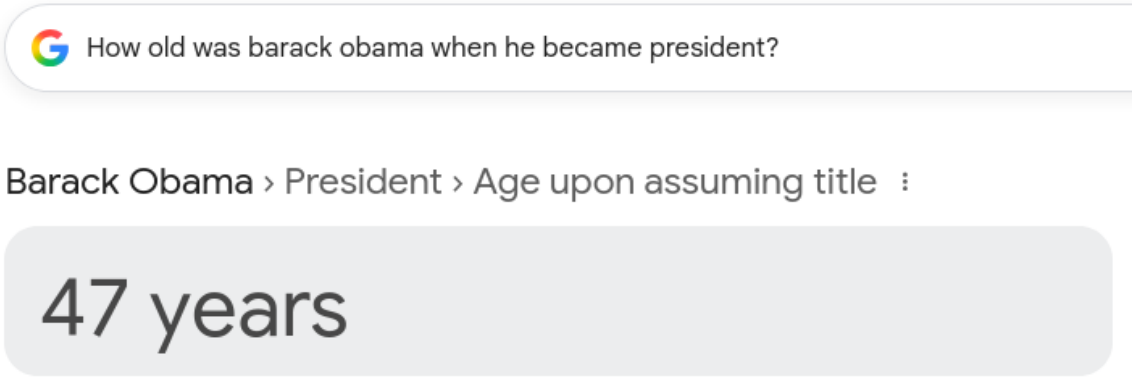


Figure 2.2: Example of a factual question answered via knowledge graph (screenshot from Google Search).

Notably, the most searched queries on Google in the last month from the time of writing are the following, in order of popularity [127]:

`google, weather, youtube, amazon, nfl, news, reddit, halloween, facebook, walmart.`

Several of these queries do not articulate an information need through a complex description, but simply name an entity (e.g., "google", "youtube", "amazon"). This highlights the central role of entities in modern information access systems.

2.2 Entities and How to Represent Them

"An entity is an object or concept in the real world that can be distinctly identified" [24]. Typical examples of entities include "John Smith", "Google Inc.", "Rome", which correspond to a person, an organization, and a location, respectively. These are all *named entities*, i.e., entities that can be identified by a proper noun. In practice, however, natural language processing tools also recognize more general *concepts*, such as dates (e.g., "June 20, 2025"), monetary values (e.g., "100 $"), quantities (e.g., "5 kilograms"), or other abstract objects (e.g., gravity, emotion) [346]. This occurs because, at the implementation level, the same methods can detect both named entities and concepts [24]. Modern natural language toolkits, such as spaCy [151], flair [6], or Stanford CoreNLP [221], provide models capable of extracting both types of entities [109, 113, 338]. For this reason, throughout this thesis, we use the term *entity* to refer to both named entities and concepts, following the definition adapted from Balog [24]:

Definition 2 (Entity)**.** An *entity* is a uniquely identifiable object, thing, or concept. It is described through a set of *entity properties*, always including at least one unique identifier and one name (which may in some cases coincide). Additional entity properties include types, attributes, and relationships to other entities or concepts.

For example, consider the entity Barack Obama. A possible unique identifier is "Barack Obama (44th US president)", which can serve also as a name, along with "Barack Obama." The entity belongs to the types `Person` and `Politician`. Its attributes include the full birth name "Barack Hussein Obama II" and the date of birth "Aug 4, 1961". Relationships connect this entity to other entities such as its birthplace, Honolulu, and its spouse, Michelle Obama.

Entities can be stored and maintained in a *knowledge repository* or a *knowledge base* [24].

Definition 3 (Knowledge repository)**.** A knowledge repository (KR) is a structured or semi-structured collection of entities [24].

Wikipedia [379] is a well-known semi-structured knowledge repository, where each entity is represented as an article (e.g., Barack Obama[1]) containing text semi-structured in different sections, as well as pictures and hyperlinks connected to other entities, similarly to relationships. As of September 2025, Wikipedia contains more than 7 million articles in English alone [381].

While semi-structured repositories are well suited for human consumption, machines require more formalized representations. The *semantic web* community has been working to extend the Web into a machine-interpretable form [24, 84]. This effort led to the development of the semantic web stack, encompassing standards such as uniform resource identifiers (URIs) [36], internationalized resource identifiers (IRIs) [99], XML, RDF, SPARQL [278], and ontology languages like RDFS and OWL [24].

[1] `https://en.wikipedia.org/wiki/Barack_Obama`

The Resource Description Framework (RDF) [186] data model allows to represent an entity as a set of assertions or *facts* about that entity. In this work, we will use the term *fact* according to the following definition.

Definition 4 (Fact)**.** A fact is an atomic statement, asserting a specific piece of information, typically describing a property of an entity or a relationship between entities.

Example of facts are "Barack Obama was born on Aug 04, 1961." or "Barack Obama was born in Honolulu." Instead, in RDF facts are represented as structured triples composed of subject, predicate, and object. The subject and predicate are usually URIs or IRIs, sequences of Unicode characters that uniquely identify resources (differently from URIs they better support non-Latin languages [99]). While the object is usually either a IRI or a literal—used for values such as strings, numbers, and dates, annotated with a datatype and eventually with a language tag [186]. For instance, the previous facts can be represented in RDF using the Turtle [277] serialization as follows:

```
<http://dbpedia.org/resource/Barack_Obama>
    <http://dbpedia.org/ontology/birthName>
    "Barack Hussein Obama II"@en .

<http://dbpedia.org/resource/Barack_Obama>
    <http://dbpedia.org/ontology/birthPlace>
    <http://dbpedia.org/resource/Honolulu> .
```

Listing 2.1: Excerpt from the DBpedia [81, 196, 41, 19] knowledge base entry of Barack Obama, in Turtle format [277].

```
@prefix dbr: <http://dbpedia.org/resource/> .
@prefix dbo: <http://dbpedia.org/ontology/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

dbr:Barack_Obama

       dbo:birthName "Barack Hussein Obama II"@en ;
       dbo:birthPlace dbr:Honolulu ;
       dbo:birthDate "1961-08-04"^^xsd:date ;
       rdf:type dbo:Person ;
       rdf:type dbo:Politician ;
       dbo:spouse dbr:Michelle_Obama ;
       rdfs:comment "Barack Hussein Obama II ([...] born August 4,
                        1961) is an American politician who served as
                        the 44th president of the United States from
                        2009 to 2017. [...] Obama was the first
                        African-American president [...]"@en .
```

Listing 2.1 shows some *entity properties* for the entity Barack Obama from DBpedia [81, 196, 41, 19]. It uses RDF following the Turtle [277] syntax and includes types ("dbo:Person", "dbo:Politician"), attributes ("dbo:birthName", "dbo:birthDate", "rdfs:comment"), and relationships with other entities ("dbo:birthPlace", "dbo:spouse"). Prefixes are defined to avoid repetitive long IRIs.

Structured entity knowledge—for instance, represented in RDF—is usually stored in a *knowledge base*.

Definition 5 (Knowledge base)**.** A knowledge base (KB), or a knowledge graph (KG), is a structured form of knowledge repository that organizes and stores facts about entities. These facts are typically represented using data models such as RDF [24]. When the focus is on the graph structure or on the relational nature of the stored facts, a knowledge base is often referred to as a *knowledge graph* [24].

Figure 2.3 shows an example knowledge graph about Barack Obama and some connected entities.

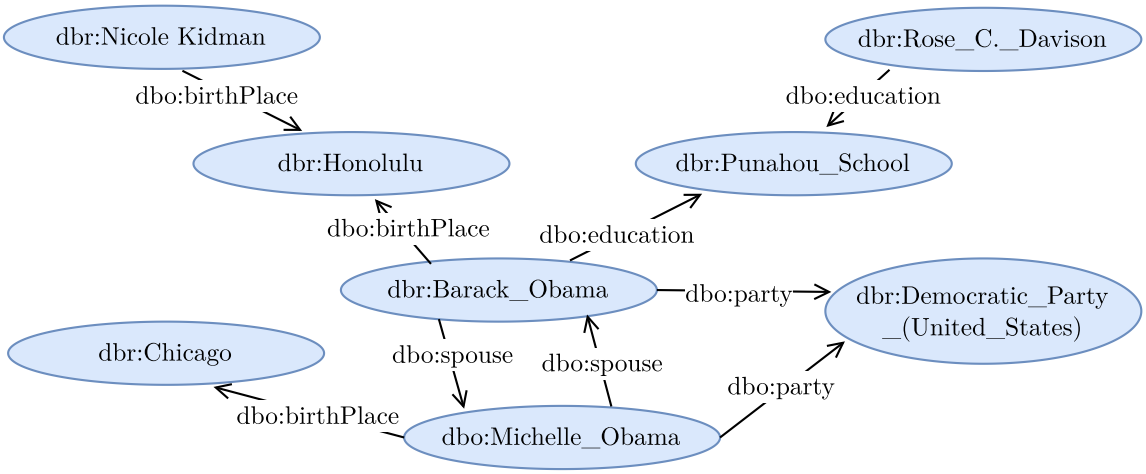


Figure 2.3: Example of knowledge graph. Prefixes are defined in Listing 2.1

Examples of KBs include Wikidata [377, 366], DBpedia [196, 41, 19], YAGO [342, 263, 219, 149, 40, 343], and Freebase [47]. Wikidata is a free collaborative project [366] that, as of September 2025, contains approximately 119 million entities and 1.65 billion facts [376, 380]. DBpedia is obtained by extracting structured, multilingual knowledge from Wikipedia and is accessible through semantic web and linked data technologies [196]; in 2015 it comprised around 1.46 billion facts [196]. Starting with version 4 [263], YAGO aims to mitigate weaknesses of Wikidata by enforcing data quality guarantees, though these require some manual curation [342]; its latest release (4.5) combines Wikidata instance data with the Schema.org ontology, refined with the Wikidata taxonomy [342]. Freebase was an open collaborative KB launched in 2007 and served as the open core of the Google Knowledge Graph [333, 126] before being discontinued and migrated to Wikidata [262]; at shutdown in 2015 it contained over 3 billion facts and nearly 50 million entities [262].

2.2.1 Structure versus Completeness in Knowledge Repositories

Structured knowledge bases are incomplete [375, 24]: for example, as of 2014, 71% of the people represented in Freebase [47] lacked a recorded place of birth [375]. This incompleteness is further supported by several research proposals for questions answering—the task of answering natural language questions (defined in Section 3.3)—that have focused on combining the high-quality, machine-readable but low-coverage knowledge from KBs with the vast amount of unstructured information available in a less reliable form on the Web [313, 375, 312, 275, 195, 411, 393, 88]. Semi-structured knowledge repositories, positioned between structured KBs and unstructured Web text, inherit both advantages and limitations from each end of this spectrum. Compared to raw Web sources, they provide a more explicit structure, while covering a broader range of information than typical KBs (though not as broad as the Web). At the same time, they do not enforce the full rigidity of structured representations, which allows them to retain some of the flexibility and recall of textual sources, albeit with less support for fully compositional querying than strictly structured KBs [195].

KB incompleteness also stems from the constant changes intrinsic to information: for example, the US president must be updated periodically, people may change residency, and novel facts are continually produced—such as the announcement of a new movie. Keeping a KB up to date therefore requires ongoing effort from editors and content managers [24].

Although not required by definition, real-world KBs (such as Wikidata [377, 366]) and unstructured KRs (such as Wikipedia [378]) support *CRUD* operations (Create, Read, Update, Delete) [102] to facilitate maintenance over time [325]. These operations are also formalized in the SPARQL/Update language [319].

As later described in Section 3.1, the need to keep KBs and KRs up to date motivates the development of natural language processing methods that automatically extract entities, relations, and attributes from unstructured text [24]. These methods can be used to populate and extend existing KBs and KRs, a task known as *knowledge base population* [24], thereby reducing the amount of manual curation required for updating.

In this work, NIL prediction is particularly relevant due to the occurrence of novel entities—also known as NIL entities—not present in public KRs in the legal domain, defined as the *novel entity challenge (Ch. 4)* in Section 1.1. Specifically, this task aims at identifying whether an entity mention in text refers to an entity absent from the KB (or KR) [328], thereby allowing the KB (or KR) to be extended with newly identified entities.

2.3 Language Models

To contextualize the use of automated methods for extracting information from text—which will be described in Section 3.1—we now move to the broader background on artificial intelligence and recent advances in natural language processing (NLP). In particular, we introduce the foundations of modern LLMs and discuss their relation with knowledge bases.

The question *"Can a machine think?"* dates back to 1950, when the Turing test was proposed [359]. A computer is said to pass the test if, after posing a series of written questions, a human evaluator cannot tell whether the answers come from a person or a machine [359]. More generally, AI refers to the set of abilities a machine may require emulating human behavior. These include NLP to understand and generate human language, memory to organize and store knowledge, automated reasoning to answer questions and infer new conclusions, machine learning to adapt and

detect patterns, computer vision and speech recognition to perceive the environment, and robotics to manipulate objects and navigate the world [306].

A formal definition of an AI system has recently been established by the AI Act [106]—a European regulation designed to ensure that artificial intelligence systems are trustworthy and human-centric, while protecting health, safety, fundamental rights, and the environment [106, Art. 1].

Definition 6 (Artificial Intelligence)**.** "*AI system* means a machine-based system that is designed to operate with varying levels of autonomy and that may exhibit adaptiveness after deployment, and that, for explicit or implicit objectives, infers, from the input it receives, how to generate outputs such as predictions, content, recommendations, or decisions that can influence physical or virtual environments." [106, Art. 3].

In this thesis, we focus on AI methods applied to text—that is, on techniques from NLP, the branch of AI devoted to processing and understanding natural language [28]. Its roots go back to the 1950s, with early work on machine translation and question answering [28]. Over time, the field evolved from rule-based systems to statistical methods, and more recently to approaches based on machine learning—in particular deep learning [24].

A *deep learning neural network* is composed of multiple layers (hence deep) of computing units, or *neurons*, each mapping an input vector to a single output value [172]. Deep learning (DL) models are typically data-driven [172]: they are trained on labeled data by minimizing a loss function that captures the error on the training set [138].

In recent years, the introduction of transformer architectures [361] has revolutionized NLP, establishing them as the dominant paradigm for building large language models, i.e., models pre-trained on massive text corpora and subsequently adapted to downstream tasks [130].

Definition 7 (Language Model)**.** A language model (LM) is a probability distribution over a vocabulary of tokens, conditioned on an input sequence, and used to predict the next token [340, 172].

We speak about *tokens* instead of words because modern LMs do not operate at the word level but rather on subword units, which can be whole words, word parts, symbols, or characters [245]. The use of a token-based vocabulary, instead of a word-based or character-based one, derives from the need to balance vocabulary size and coverage. Word-based vocabularies suffer from the issue of out-of-vocabulary words. Character-based vocabularies, on the other hand, can lead to longer sequences (a sentence is represented as a long sequence of single characters) and increased computational costs [344]. Token-based tokenizers, such as Byte-Pair Encoding (BPE) [323], can represent frequent words using a single token, still allowing the representation of rare or novel words by combining multiple tokens.

This definition of LM is in-line with *causal language model (CLM)*, that predicts the next token given the previous ones [162, 172], and these models are also referred to as *autoregressive models* [172], i.e., "models regressing the outcomes on previous values of the same time series" [295], where in NLP we have a sequence of tokens instead of a time series.

Causal Language Modeling

Given a sequence of tokens $X = [x_0, x_1, \ldots, x_{|X|}]$ and a vocabulary $V = \{v_0, v_1, \ldots, v_{|V|}\}$ a causal LM estimates

$$P(x_i = v \mid x_0, x_1, \ldots, x_{i-1}) \qquad \forall v \in V, \tag{2.1}$$

that is, a probability distribution over the entire vocabulary V for the next token.

In the remainder of this work, for convenience, we will use the following simplified formulation

$$P(x_i \mid x_0, x_1, \ldots, x_{i-1}) = P(x_i \mid x_{<i}), \tag{2.2}$$

implying that the CLM calculates a probability over the entire vocabulary.

The probability of the whole sequence is commonly estimated by the factorization [172]

$$P(X) = \prod_{i=1}^{|X|} P(x_i \mid x_{<i}). \tag{2.3}$$

Deep-learning based causal LM are trained via maximum likelihood estimation [162], i.e., by maximizing the log-likelihood of the training data [241], that corresponds to N sequences of tokens $[x^0, x^1, \ldots, x^{|N|}]$. The training loss is thus

$$\mathcal{L}_{CLM} = -\frac{1}{N} \sum_{j=1}^{N} \log P(x^j) = -\sum_{j=1}^{N} \sum_{i=1}^{|x^j|} \log P(x_i^j \mid x_{<i}^j). \tag{2.4}$$

Causal language models, that estimates the probability distribution for the next token given the previous ones, are used for text generation by iteratively sampling the next token from the distribution (e.g., greedily selecting the most probable token) and appending it to the input sequence [172].

Masked Language Modeling

Another paradigm is *masked language modeling*, where some tokens in the input sequence are randomly *masked*, i.e., replaced with a special token, and the model is trained to predict the original tokens based on the bidirectional (left and right) surrounding context [90].

Formally, given a sequence of tokens $X = [x_1, x_2, \ldots, x_{|X|}]$ and a vocabulary $V = \{v_0, v_1, \ldots, v_{|V|}\}$, a random subset of positions $M \subseteq \{1, \ldots, |X|\}$ is selected for masking, yielding a corrupted input sequence $\tilde{x}$. For each masked position $i \in M$, a masked LM estimates the probability distribution over the entire V of the original token x_i conditioned on its left and right context:

$$P(x_i \mid \tilde{x}_{\setminus i}) = P(x_i \mid x_{<i}, x_{>i}), \tag{2.5}$$

where $x_{<i}$ and $x_{>i}$ denote the tokens preceding and following position i, respectively. The training loss, corresponding to maximum likelihood estimation over the masked positions of N sequences, is

$$\mathcal{L}_{MLM} = -\frac{1}{N} \sum_{j=1}^{N} \sum_{i \in M^j} \log P(x_i^j \mid x_{<i}^j, x_{>i}^j). \tag{2.6}$$

While CLMs are suited for text generation, masked language models (MLMs) are typically used for sequence labeling or token classification tasks, taking advantage of the context from both sides [172]. Examples of such tasks include part-of-speech (POS) tagging, where each token is

assigned a part-of-speech category such as `Noun`, `Verb`, or `Adj`, and entity recognition (ER)—described in Section 3.1—where each token is assigned a label indicating whether it is part of a named entity and its type, such as `Person`, `Location`, or `Organization` [172].

Note that both CLM and MLM do not require time-consuming manual labeling. The training set is simple text, used for predicting the next word or randomly masked ones. These processes, which allowed training language models on very large corpora, are known as self-supervised learning [172].

Transformers

In 2017, *Transformers* have been introduced and evaluated on machine translation, a language modeling task, demonstrating superior performance and higher parallelization at training-time compared to previous architectures [361]. The transformer uses an encoder-decoder architecture, and it is capable of modeling sequences of tokens relying on the *attention mechanism* [361]. An *encoder* processes tokens as input and transforms each into a vector representation, commonly referred to as an embedding [172].

Vector Semantics and Representation Learning To place the concept of *embedding* in context, it can be connected to the fields of representation learning and vector semantics. Representation learning provides methods for learning useful representations from data, including text (words, sentences, or documents) [35]. With respect to vector semantics, its origin can be traced to the 1950s with the distributional hypothesis [136]—i.e., words occurring in similar contexts tend to have similar meanings [172]—and the idea of using points in a three-dimensional space for representing words connotation [256]. Words like "car" and "automobile" are synonyms, but their "similarity" cannot be captured via string similarity. An example of string similarity is the edit distance, i.e., the minimum number of insertions, deletions, or substitutions needed to transform one string into another [172]. Also, "cats" and "dogs", or "coffee" and "cup" can be considered similar or related: they are close in meaning or usage even though they are not lexically similar. The intuition of vector semantics is to assign each word an *embedding*, i.e., a vector for representing a word in a multidimensional semantic space, where similarities between embeddings would reflect the similarities of words' meanings [172].

Different techniques have been proposed for obtaining word embeddings. In 2013, Mikolov et al. [230] introduced word2vec, which, in the case of the CBOW (Continuous Bag of Words) architecture, learns dense semantic embeddings by training a model to predict a target word from its surrounding context [230]. *Dense* embeddings refer to low-dimensional vectors, in contrast with very long *sparse* vectors, such as one-hot vectors, whose size corresponds to the length of the entire vocabulary (often between 10,000 and 50,000 words) [172]. A standard illustration of the utility of word embeddings is provided by the following analogy:

$$\vec{\text{King}} - \vec{\text{Man}} + \vec{\text{Woman}} \approx \vec{\text{Queen}}. \tag{2.7}$$

Subtracting the embedding of "man" from "king", then adding "woman" produces a vector close to "queen", demonstrating some kind of "vector-based reasoning" [231].

However, assigning a single embedding to each word in the vocabulary is insufficient for polysemous words, such as "mouse" or "bank", which have multiple distinct senses. For instance, "mouse" can refer either to the small rodent or to the device used to control a computer, depending on the context [172].

Transformers are able to compute *contextual embeddings*, i.e., dense vectors representing the meaning of a token in the context [172]. Figure 2.4 shows the contextual embeddings computed

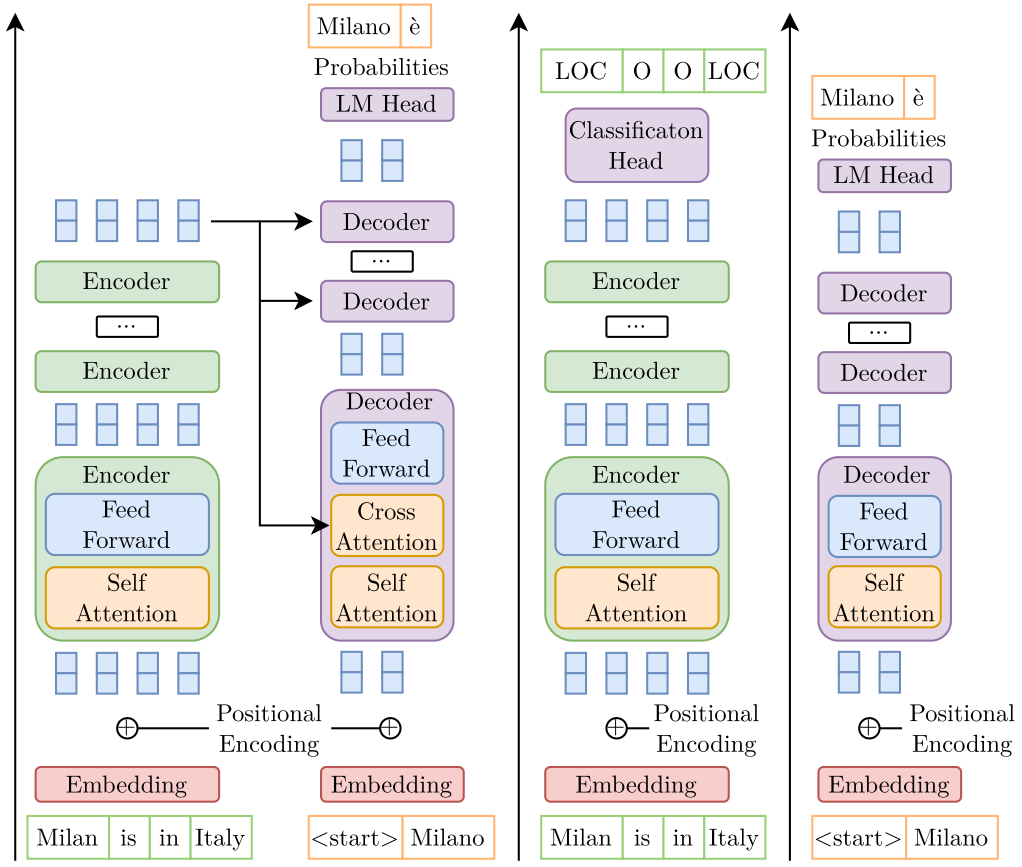
Figure 2.4: Contextual embeddings of the word “die” calculated with BERT [90], projected in two dimensions. Figure from Reif et al. [297].

with BERT [90], an encoder-only transformer, for the word “die”, used in different contexts with different meanings, both in English and in German. The embeddings are positioned in semantic clusters according to the meaning of “die”: DIE (GERMAN), DIE (DEATH), and DIE (DICE).

The Transformer Architecture Turning back to the transformer architecture, it is composed of a stack of encoders and a stack of decoders, as depicted in Figure 2.5. Given an input sentence of N tokens, the sentence is divided in tokens, or *tokenized*, then each token is assigned a static embedding from an embedding matrix $E \in \mathbb{R}^{|V| \times d}$ where V is the vocabulary of the model and d the depth or number of dimensions of the embeddings. Subsequently, the token embeddings are processed by encoder and decoder blocks that contextualize them, by incorporating the meaning of contextual tokens. *Attention* is the mechanism that weights and aggregates the token embeddings [172].

The original encoder-decoder transformer [361], depicted in Figure 2.5(a), has been introduced for machine translation. For example, given the English sentence “Milan is in Italy”, the expected output in Italian is “Milano è in Italia”. The input tokens in English are provided to the encoder (bottom left of Figure 2.5) and the decoder (bottom right) receives a special starting token “<start>”. In the first forward pass the model generates “Milano”. At the next step, the encoder receives the same input, while the decoder receives as input the already generated sequence of tokens in Italian, “<start>” and “Milano”, to generate “è” (the Italian verb to be). This iterative process continues until the model generates an end-of-sequence token or reaches maximum generation length.

Both encoder and decoder receive as input an embedding for each token and produce another embedding as output. Formally, let $X \in \mathbb{R}^{N \times d}$ be the input matrix of shape $N \times d$. After processing



(a) Encoder-Decoder (b) Encoder-only (c) Decoder-only

Figure 2.5: Overview of the transformer architectures encoder-decoder, encoder-only, and decoder-only, respectively for machine translation, entity recognition, and text generation.

X the first encoder produces $X^i \in \mathbb{R}^{N \times d}$, a matrix of the same shape of X. This process is repeated for every encoder block, progressively enriching the tokens' embeddings with contextual information weighted by the attention mechanism.

As visible in Figure 2.5(a), both the encoder and decoder modules include a *self-attention* layer, while the decoder also has a *cross-attention* layer. Self-attention can be considered as a mechanism for constructing contextual representations of a token's meaning by focusing on and combining information from nearby tokens [172]. Indeed, tokens in a sequence attend to the same sequence. The cross-attention, instead, is performed between different sequences. For example in machine translation, the decoder, while generating in the target language, selects which parts of the source sentence (in the source language) to pay attention to [23]. In the transformer [361], each decoder layer attends to the final representations from the encoder stack.

For further details of the attention mechanism and its implementation, refer to Jurafsky et al. [172].

The *encoder-decoder* transformer for machine translation features a language modeling head, as visible in Figure 2.5(a) that estimates the probability $P(x_i \mid y, x_{<i})$ where x are the tokens in the target language, while y in the source language. Using both encoder and decoder layers allow the model to use two different vocabularies, one for each language [172]. This model is usually trained in a supervised setting on translational datasets [361].

Besides the encoder-decoder architecture, two relevant modifications are the *encoder-only* and the *decoder-only* transformers. The *encoder-only*, whose architecture is sketched in Figure 2.5(b), is especially used for embedding or sequence classification tasks, such as document retrieval with dense vector and entity recognition [172, 90], and it is trained with MLM for learning to create useful token representations.

Decoder-only models can be used for language modeling, as visible in Figure 2.5(c), but they use the same vocabulary for input and output tokens, differently from encoder-decoder models that are used for machine translation or even speech recognition—where the input represents speech [172]. Today, the most popular architecture for LLMs is the decoder-only [410, 280], although recent work demonstrates interest in the potential of the encoder-decoder architecture [410].

A core component of the success of transformers is *transfer learning*, i.e., "the transfer of knowledge from a related task that has already been learned" [355], which is often achieved via pretraining on huge corpora with self-supervised objectives, such as MLM or CLM, that do not require human labeling, to subsequently fine-tune the models parameters to downstream tasks [172, 90, 283, 285].

Devlin et al. [90], Radford et al. [283], indeed, demonstrate that rich pretraining on generic language tasks, such as MLM and CLM, is useful also for downstream tasks like entity recognition and question answering.

Furthermore, Raffel et al. [285] and Radford et al. [284], respectively, have achieved state-of-the-art performance with CLMs on several language understanding tasks by mapping them in a text-to-text format, and have demonstrated the ability of CLMs in a zero-shot setting. In this context, *zero-shot* refers to using a model for a task for which it has not been fine-tuned for, by providing a natural language description of the task [54].

Later, by scaling up the number of model parameters, large language models (LLMs) such as GPT-3 [54] (with 175 billion parameters) achieved surprising performance in zero-shot and few-shot settings, the latter extending the former by providing a handful of natural-language examples of the task. In some cases, performance was even competitive with state-of-the-art fine-tuned models [54]. These improvements enabled user to *prompt* or instruct language models to do specific task, eventually providing one-shot or few-shot examples to better take advantage of their *in-context learning (ICL)* capability [172]. A *prompt*, indeed, is a textual instruction the user gives to a model for accomplishing a specific task. *ICL* refers to learning a task or improving the performance on that task without updating model parameters, e.g., with few-shot prompting [172].

However, simply increasing the number of parameters has not been sufficient to improve models' ability to follow user instructions [257]. In practice, models may still produce unhelpful, toxic, or hallucinated content [226, 167]—where hallucination refers to output that is incoherent, nonsensical, or not faithful to the provided input. For this reason, techniques to *align* models to user's intent have been applied to LLMs, including some based on supervised fine-tuning with curated examples [419] and on reinforcement learning, such as *reinforcement learning from human feedback (RLHF)* [257].

Beyond alignment, researchers have also explored methods to strengthen the reasoning ability of LLMs. Chain-of-thoughts (CoT) prompting [373] decomposes complex reasoning tasks into intermediate steps that are explicitly generated before the final answer. More recently, so-called

large reasoning models (LRMs) [353] have been trained to produce long chains of thoughs [255, 86], further enhancing the model's reasoning capability.

For broader and more detailed historical overviews of LLMs see Zhou et al. [418] and Tie et al. [353].

2.3.1 The Computational Complexity of Transformers

While transformers include parallelization advantages at training time, at inference-time the computational complexity of the original self-attention mechanism scales quadratically with the input sequence length N, and thus transformer-based inference is quadratic as well [33]. In MLMs, self-attention is computed once, whereas autoregressive CLMs generate text iteratively and must execute self-attention N times to produce N tokens. The cost of generating a single token in a CLM can nevertheless be reduced to linear time by using key–value caching, which stores and reuses the attention keys and values from previous steps instead of recomputing them [11].

Beyond caching, a large body of work replaces or approximates full attention to handle long contexts [229]—e.g., by performing local attention in a sliding window [33] instead of attending to all tokens.

2.3.2 Large Language Models and Knowledge Bases

Modern large language models (LLMs), such as GPT [253, 54] and Llama [128], have demonstrated remarkable capabilities in understanding natural language, enabling the execution of NLP tasks without task-specific fine-tuning [54, 257, 128]. They are increasingly applied to complex domains, including scientific discovery [5, 244] and intelligence analysis [288]. Furthermore, LLMs retain considerable factual knowledge from their pretraining corpora [68, 8, 266, 300], and state-of-the-art models are trained on trillions of tokens. For instance, DeepSeek-V3, with 671B parameters, has been trained on 14.8 trillion tokens [87].

Although they contain broad knowledge, LLMs suffer from incompleteness due to finite capacity and the evolving nature of information. Their parametric knowledge is also difficult to update reliably: editing methods [402, 372] may introduce side effects such as inconsistencies [204, 71], with Cohen et al. [71] reporting more consistent edits when inserting information in-context rather than modifying model parameters. Moreover, LLMs can hallucinate [167], producing unfaithful or fabricated content.

By contrast, KBs offer explicit, structured and interpretable knowledge: query results can be traced back to underlying facts; they support global questions (e.g., counting queries), which remain difficult for LLMs [8]; and they preserve internal consistency [416]. However, KB maintenance typically requires inserting structured facts, often with human effort [24], whereas LLMs can acquire knowledge from unstructured corpora during pretraining.

Rather than being alternatives, LLMs and KBs are increasingly studied as complementary components [260, 264]. Retrieval augmented generation (RAG) and its variants embody this integration by allowing an LLM to consult external sources at inference time (see Section 3.3), indeed, these sources can include unstructured corpora and structured KBs. Recent work combines LLMs with KGs to improve their performance on answering complex questions [100, 311, 396]. For example, Edge et al. [100] automatically induces a KG from documents, organizes entities into hierarchical communities and summarizes them, enabling LLMs to answer global questions such as "What are the key topics represented in the dataset?".

Complementarily, structured retrieval interfaces—including semantic and faceted search—are not only useful to humans but can also be invoked by agentic LLMs as external tools [388, 268, 315, 401, 120]. This growing line of work suggests that LLMs and KBs serve distinct yet complementary roles: the former provide broad, contextual competence in language, while the latter offer structured, traceable, and queryable knowledge. Accordingly, recent advances in knowledge-intensive applications often rely on combining the two rather than treating them as substitutes [134, 395].

2.4 The Italian Legal System

In this section, we describe the Italian legal system, the different types of trials with the documents involved, and the relationship between Italian law and artificial intelligence.

The Italian Republic uses the *civil law* system, in contrast with countries such as the United Kingdom of Great Britain and Northern Ireland and the United State of America, which use the *common law* system. Civil law is based on codes, while common law is based on case-law [62]. Typically, civil law systems rely on four foundational codes [62, 13]:

- *Civil code*: governs private relations among individuals and organizations;
- *Civil procedure code*: sets the rules for how civil disputes are adjudicated;
- *Criminal code*: defines crimes and the corresponding penalties;
- *Criminal procedure code*: regulates the investigation, prosecution, and adjudication of criminal offenses.

These codes are intended as coherent and consistent principles for supporting the broader legal system. In contrast, common law systems typically develop codes in response to specific issues, often drafting them only after principles have already emerged through case law [62]. Furthermore, in common law, precedent cases are of central importance, as judges are required to follow earlier decisions, and a single ruling from a higher court may determine the outcome of a case. This is different in civil law, where a single precedent case is generally not treated as binding, with the possible exception of decisions in the constitutional sphere [62].

The organizational structure of civil law court systems is broadly similar to that of common law systems, featuring a court of first instance, a court of appeals, and a supreme court. Additionally, specialized courts exist with jurisdiction limited to particular areas of law. Unlike in common law jurisdictions, juries are generally not used in civil law systems. The cases are decided by judges, eventually including lay judges [62], such as in the Italian court of assize [194, Art. 3]—which is competent to judge serious criminal offenses, such as the ones carrying maximum penalties of at least 24 years of imprisonment [85, Art. 5].

In the Italian system, both in civil and in criminal procedures, the three grades of jurisdiction operate as follows. The courts of first instance decide the facts and merits of a case in their entirety [293, Arts. 99–310]; [85, Arts. 429–544]. The courts of appeal then provide a second, comprehensive review to ensure both factual and legal correctness [293, Arts. 339–359]; [85, Arts. 593–605], thereby giving effect to the constitutional guarantee of challenging judicial decisions [74, Art. 111]. Finally, the Court of Cassation represents the third and highest grade, serving as the supreme authority tasked not with re-examining facts but with ensuring the uniform interpretation and application of the law, as regulated in the civil [293, Art. 360–382] and criminal [85, Art. 606–613] codes of procedure. Decisions of the Court of Cassation are not laws and are not formally

binding, yet they strongly influence lower courts and promote a consistent interpretation of the law, reflecting the court's so-called *nomophilactic function* [191]. In Italy, the supreme court has a dual role: to render a final judgment on individual cases, thereby ensuring justice, and to establish interpretative guidelines that shape national jurisprudence [294, Art. 65].

2.4.1 Legal Proceedings

In the Italian legal system, a legal proceeding may arise in either the civil or the criminal sphere, with distinct initiation mechanisms reflecting the nature of the dispute.

Civil proceedings are initiated by a private party. The *plaintiff* begins the process by filing a writ of summons with the competent court. This document must indicate: the identities of the parties, the tribunal being addressed, the factual and legal basis of the claim, the evidence intended to be produced, and the date of the hearing [293, Art. 163].

The *defendant* responds with a statement of defense, outlining their defenses and any counterclaims [293, Art. 167]. From this moment, the judicial authority schedules hearings, manages the collection of evidence, and ultimately renders a decision—a *judgment*—which must include motivations as required by the Constitution [74, Art. 111].

Criminal proceedings begin with the occurrence of a potential criminal offense. When the report of a crime is received, the public prosecutor opens a preliminary investigation [85, Art. 326].

During this phase, investigative activities are performed by the public prosecutor and the judicial police, and documented in official records [85, Art. 373]. The suspect is formally notified through a guarantee notice, a document informing them of the charges and their rights [85, Art. 369]. If sufficient evidence exists, the prosecutor requests an indictment and the judge for the preliminary hearing decides whether the case proceeds to trial [85, Art. 416].

The documentation of civil and criminal proceedings is governed by the provisions of the respective codes of procedure [293, Art. 168–169]; [85, Art. 431, 114], which govern the formation, content, and accessibility of case files. Civil files typically include the summons, payment receipts, pleadings, briefs, hearing minutes, judicial orders, procedural documents related to the taking of evidence, and judgments [293, Art. 168], while criminal trial files contain procedural records related to prosecutability and civil claims, non-repeatable investigative activities documented by the police or the parties, records of evidentiary proceedings, official documents, and, when applicable, seized items [85, Art. 431]. Furthermore, the content of the judgment are regulated, requiring identification of the parties, a statement of facts, the legal reasoning, and the operative part of the decision [85, Art. 546]; [293, Art. 132]. For this reason, judgments can be treated as semi-structured documents, in the sense that they usually contain identifiable sections, even though the expression and organization of legal language may vary—for example, depending on the style of the judge [125, 116, 270].

2.4.2 Artificial Intelligence in the Legal Domain

AI offers considerable opportunities for the legal sector, fundamentally reshaping traditional workflows through the automation of routine tasks and the enhancement of complex decision-making processes [125]. The application of AI is not intended to supplant human legal expertise but rather to act as a sophisticated form of decision-support systems (DSSs), augmenting the capabilities of legal professionals. Such systems can perform an array of tasks, from conducting advanced prece-

dent searches to predicting potential case outcomes, thereby improving efficiency while maintaining the integrity of human judicial oversight [125].

An example of a practical AI tool for the legal domain is the ADELE project [117], which addressed different legal domains, including Value added tax (VAT) and trademarks and patents. The project aimed to support legal research and decision-making through several functions. These included the automatic extraction of citations between legal documents and the construction of citation networks to identify influential cases and clusters of similar decisions. ADELE also applied deep learning models to extract summaries and keywords from legal texts, facilitating information retrieval and organization. In addition, the project experimented with argument extraction and with predicting case outcomes by examining possible correlations between the arguments advanced by the parties and the decisions of the courts [125].

For this thesis, the most important use case for AI in the legal domain is automatically analyzing and organizing—e.g., by criteria such as relevance to a specific case—vast quantities of documents, avoiding time-consuming and expensive manual operations. Indeed, AI systems have been applied in common law jurisdiction to identify relevant documents among millions [125]. However, also in civil law jurisdiction, some cases may require the analysis of a high number of related decisions to ensure the uniform interpretation of the law [101]. Furthermore, criminal investigations frequently involve the examination of vast amounts of heterogeneous documents, such as web pages, chat records, investigative reports, or expert analyses [348, 29, 271].

The use of AI-based systems in this context enables users to satisfy information needs through multiple modalities, such as faceted search [177], which allows filtering based on entities mentioned in documents, question answering [152, 336], and semantic text annotation. The latter can support reading by highlighting salient elements in a judgment [347, 131, 114] (e.g., sections or parties). These use cases motivate part of the work conducted during my Ph.D., in particular the research presented in the following publications, which I co-authored: Pozzi et al. [271], Bellandi et al. [32], Pozzi et al. [274], and in Chapter 4.

In the context of information retrieval and organization, the importance of data annotation cannot be overstated. Projects in Italy, such as the PNRR-PON "Giustizia Agile" (Italian for "agile justice"), have focused on creating labeled datasets for Italian legal documents to support further AI research. This included the crucial task of entity recognition (ER) (described in Section 3.1), which is foundational for enabling AI systems to accurately identify and classify key entities within legal texts, such as persons, organizations, or mentioned laws and regulations, which can be used to improve the retrieval mechanisms [125], e.g., with faceted search [17, 141].

2.4.3 Artificial Intelligence Regulations

The integration of artificial intelligence (AI) into the legal and judicial system presents significant challenges, particularly concerning fundamental rights, transparency, and data protection. The European Union has addressed these concerns through the *Artificial Intelligence Act (Regulation (EU) 2024/1689)* [106], often referred to as *AI Act*, which classifies AI systems used in the administration of justice as high-risk [106, Annex III]. While the AI Act does not prohibit their use, it mandates *human oversight*—i.e., that the final decision-making authority remains with a human [106, Art. 14]—ensuring that AI functions as a support mechanism rather than an autonomous decision-maker [125]. This aligns with the European Ethical Charter on the Use of AI in Judicial Systems [75], which emphasizes principles such as transparency, non-discrimination, and due process.

These principles have been further consolidated in Italy with the approval of the Law on Artificial Intelligence (L. 1146/2024) [158]. This national law integrates and strengthens the European framework, reaffirming the anthropocentric approach whereby AI in the judicial sector may be used exclusively for instrumental and support activities. It also introduces criminal provisions, including penalties for the illicit dissemination of AI-generated or manipulated content, thereby mitigating emerging risks [337].

These regulatory developments are already reflected in judicial decisions across both Italy and the EU. The Court of Bologna, for instance, has considered cases involving algorithmic decision-making, focusing on principles of equality and non-discrimination. Similarly, the Court of Justice of the European Union (CJEU), in the *SCHUFA Holding* case [171], examined whether an AI-based credit scoring system constitutes automated decision-making under the GDPR [107]. Together, these examples highlight the ongoing legal and ethical discourse on the governance of AI in justice systems.

A further crucial aspect concerns *privacy* and the lawful processing of data used to train or operate LLMs. Legal and judicial documents often contain personal data [107, Art. 4], which must be processed lawfully, fairly, and for specified purposes in accordance with the *General Data Protection Regulation (GDPR, Regulation (EU) 2016/679)* [107, Art. 5]. In this context, data controllers—the entities determining the purposes and means of data processing [107, Art. 4.7]—are encouraged to adopt *anonymization* or *pseudonymization* techniques [107, Art. 4.5] to reduce privacy risks, as these measures limit or remove the identifiability of individuals while still allowing useful data analysis. In particular, the controller must ensure that data are not transferred outside the European Economic Area unless an adequate level of protection is guaranteed [107, Art. 44–49]. The use of API-based models complicates compliance with these principles, as such systems typically involve transferring data to external servers operated by third-party providers, thereby making it more difficult to ensure adherence to the principles of data minimization and purpose limitation [107, Art. 5.1.c–b].

Under the AI Act [106], the *provider* is the entity that develops or markets the AI system—for example, a company offering a model through an online API [106, Art. 3.3]—whereas the *deployer* is the natural or legal person, public authority, or other body that uses the system under its authority [106, Art. 3.4]. In legal settings, the deployer may correspond to a court or law enforcement agency integrating an AI model to support document review or investigative analysis. When the model is accessed through an external API, these entities must establish a lawful processing basis (e.g., consent or public interest) [107, Art. 6 and 9], define roles and safeguards via a data processing agreement [107, Art. 28.3], and comply with additionally transfer rules if data is processed outside the EU [107, Art. 44–49].

Moreover, the AI Act reinforces these obligations for high-risk systems, requiring providers and deployers to implement risk management, data governance, and logging mechanisms for traceability [106, Art. 9–12], and to ensure that the system preserves human oversight and data security [106, Art. 14–15]. When using remote APIs, these safeguards depend on the provider's infrastructure, making compliance difficult to demonstrate. Conversely, locally deployed models allow institutions to retain control over the data processing environment, enforce internal access policies, and ensure that confidential information remains within the organization's secure perimeter—a crucial requirement for judicial and investigative applications.

2.4.4 Illustrative Examples of Domain Data

To conclude the section, Figures 2.6 and 2.7 present two illustrative examples: a civil judgment and an investigative chat log extracted from a seized suspect's smartphone.

ORDINARY COURT OF BOLOGNA
Civil Division
Judgment No. 987/2025
Published on February 14, 2025

ITALIAN REPUBLIC
IN THE NAME OF THE ITALIAN PEOPLE

The Court, sitting as a single judge, in the person of Judge Laura Conti, has delivered the following:

JUDGMENT

in the civil case registered under No. 3210/2024
between

Giovanni Bianchi, residing in Bologna, represented and defended by Attorney Paolo De Santis, with elected domicile at his office in Bologna — plaintiff

and

Davide Ricci, residing in Modena, represented and defended by Attorney Silvia Ferri, with elected domicile at her office in Modena — defendant

FACTS
The plaintiff summoned the defendant to court, alleging that on April 3, 2022, he lent him the sum of EUR 10,000, as evidenced by a private written agreement signed by both parties, with repayment due by December 31, 2022, pursuant to Art. 1813 c.c. Despite repeated requests, the defendant failed to repay the sum. The defendant claimed that he had already reimbursed part of the debt in cash and that the remaining amount had been compensated by the plaintiff's use of his vehicle, pursuant to Art. 1241 c.c. (compensation of debts).

REASONS FOR THE DECISION
The documentary evidence and the bank records produced confirm the existence of a loan contract under Art. 1813 c.c. and disprove any effective compensation under Art. 1241 c.c. No evidence was provided of partial repayment. Pursuant to Art. 2697 c.c., the burden of proof rests on the party alleging payment. The plaintiff's version is consistent and corroborated by witness statements under Arts. 2721–2725 c.c.
Therefore, the defendant must return the sum of EUR 10,000, plus statutory interest under Art. 1282 c.c. from January 1, 2023 until payment.

RULING
The Court, definitively ruling, dismissing all other claims,
orders Mr. Davide Ricci to pay Mr. Giovanni Bianchi EUR 10,000 plus statutory interest from January 1, 2023 until payment in full (Arts. 1282 and 1284 c.c.);
orders the defendant to reimburse the plaintiff's legal expenses, quantified at EUR 1,800 plus VAT and surcharges, pursuant to Arts. 91–92 c.p.c.

Bologna, February 10, 2025
The Judge
Laura Conti

Figure 2.6: Illustrative example of an Italian civil judgment. c.c. stands for Italian Civil Code [292] and c.p.c. for Italian Code of Civil Procedure [293].

Participants: Giovanni Bianchi, Davide Ricci
Date Range: 2025–04–22 to 2025–04–25

[2025/04/22 – 11:32] Giovanni Bianchi: Davide, we need to talk.
[2025/04/22 – 11:35] Davide Ricci: About what now?
[2025/04/22 – 11:37] Giovanni Bianchi: You know what. You ignored the court order again.
[2025/04/22 – 11:39] Davide Ricci: I told you, I don't have that money anymore. Things went bad with the workshop.
[2025/04/22 – 11:40] Giovanni Bianchi: Don't lie. I saw your bank statements, the transfer went straight to that trip to Milan and that "business dinner" at *Hotel Duomo*.
[2025/04/22 – 11:42] Davide Ricci: It was all work-related.
[2025/04/22 – 11:45] Giovanni Bianchi: Meet me Friday, 19:00, at Bar Centrale. Bring the papers.
[2025/04/22 – 11:46] Davide Ricci: Fine.

[2025/04/24 – 18:51] Giovanni Bianchi: *[Voice message — 2 min 58 sec]*
Transcription:
Listen, Davide, I'm done playing games. The Court of Bologna, Judge Conti herself, said you must pay me back — it's all there in the papers. You keep saying you're broke, but I know about that account at Banca Emilia Romagna and your new tool deal with Officina Rinaldi in Modena. You spent my money on that weekend in Milan, dinner at Grand Hotel et de Milan and even those tickets for the race at the Monza Circuit.
I checked — you still have that Piaggio van parked behind your shop in Via Emilia. Don't tell me you sold it. You've been moving money through your cousin's account at Credito Padano.
If you don't bring something — even half — on Friday, I swear, Davide, I'll find you myself. I'll be waiting near the pharmacy next to Piazza Maggiore, around six-thirty. Don't make me do something stupid.

[2025/04/24 – 19:02] Davide Ricci: Calm down. I'll come, okay?
[2025/04/24 – 19:03] Giovanni Bianchi: We'll see.

Figure 2.7: Illustrative example of chat exchange.

Chapter 3

Related Work

In this chapter we review the literature on the concrete tasks most directly related to this work. We begin with information extraction (IE), in Section 3.1, and then narrow our focus to entity extraction (EE), since we concentrate on entities and do not address other IE tasks, such as relation extraction. EE tasks include entity recognition (ER), entity linking (EL), NIL prediction, and NIL clustering. The latter two address the *novel entity challenge (Ch. 4)*, i.e., the problem of identifying mentions that refer to previously unseen or NIL entities.

We use the term knowledge consolidation (KC) to collectively denote EL, NIL prediction, and NIL clustering, since these tasks consolidate the extracted knowledge by grouping together mentions of the same entities and thereby support downstream processes such as faceted search [17, 141]. By contrast, ER identifies mentions of entities without determining which of them refer to the same underlying entity.

After reviewing EE and KC, in Section 3.2, we turn to the literature on data integration architectures and document management models for the legal domain, and finally to question answering (QA), with particular attention to knowledge-graph question answering (KGQA), in Section 3.3.

3.1 Entity Extraction

Capturing the meaning of text is a central goal of natural language processing [24], and the automatic extraction of semantic information from text—such as mentions of named entities, relationships, and attributes—is known as *information extraction*. This has been the objective of major competitions, including the Message Understanding Conference (MUC) [144, 346, 70], the Computational Natural Language Learning (CoNLL) 2003 shared task [354], and the Automatic Content Extraction (ACE) program [93].

Definition 8 (Information Extraction)**.** "*Information extraction (IE)* refers to the automatic extraction of structured information such as entities, relationships between entities, and attributes describing entities from unstructured sources" [310].

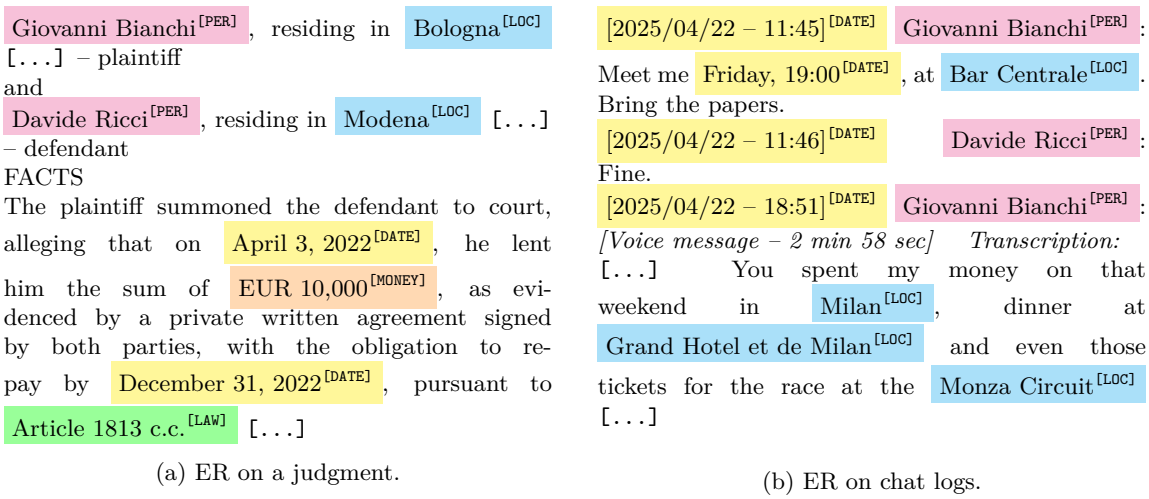
The use of IE techniques is generally aimed at either *semantic text annotation* or *knowledge base population* [24]. In the former case, annotating text with information from a KB enriches the text with additional semantics, enabling advanced information access functionalities, such as entity-based search and filtering via faceting [24, 17, 141]. In the latter case, IE is used to populate or

update a KB with information extracted from unstructured sources to enhance the KB's coverage and keep it up-to-date. In this work, we focus on entity extraction (EE) concentrating on entities, without considering broader IE tasks such as relation extraction.

In order, we define and describe the literature of entity recognition (Section 3.1.1), entity linking (Section 3.1.2), NIL prediction and clustering (Section 3.1.3), EE in the legal domain (Section 3.1.4), and finally we describe EE resources for the Italian language (Section 3.1.5).

3.1.1 Entity Recognition

Definition 9 (Entity Recognition)**.** Entity recognition (ER) is the task of identifying and classifying named entities or concepts in text into predefined categories such as persons, organizations, and locations [24, 179, 172].



(a) ER on a judgment.

(b) ER on chat logs.

Figure 3.1: ER application for semantic text annotation with entity highlights and labels.

An example application of ER on legal judgments and chat logs is shown in Figure 3.1, where entities such as persons, locations, dates, monetary values, and legal references are identified and classified.

While traditionally focused on entities like persons, organizations, and locations, ER systems have been applied for recognizing wide range of entity types, such as dates, monetary values [346], biomedical entities [135, 15, 214], legal references [66, 61, 64], and more [24]. The NLP community manifested interest in extending the types of entities recognized, proposing hundreds of entity types [321, 320, 405] and introducing the *fine-grained entity typing* field, which aims to classify entities into a larger set of more specific categories [370].

ER, as an information extraction task, contributes to semantic text annotation, knowledge base population and downstream tasks such as entity-based search or maintaining KBs up-to-date [24]. For example, in STA, recognizing entities allows to semantically highlight documents to let readers quickly grasp the main topics discussed by looking at the entities mentioned [347, 131, 114].

Besides, ER is applied in specific domains, such as the healthcare [135, 15, 214] or the legal [66, 61, 64] ones, to extract domain-specific entities and support specialized applications. Beyond these areas, ER can contribute to document summarization [182, 208, 303] and machine translation [391, 145], where it helps preserve and highlight key entities, or to question answering [234, 384], by identifying entities mentioned in the query.

Formalization

Formally, given a sequence of tokens X,

$$X = [x_0, x_1, \ldots, x_{|X|}] \tag{3.1}$$

and a set of categories T,

$$T = \{\texttt{Person}, \texttt{Location}, \texttt{Organization}, \ldots\}, \tag{3.2}$$

an ER system produces a set of tuples Y:

$$Y = \{(i_{start}, i_{end}, t^i), \ldots\}, \text{ where } i_{start}, i_{end} \in \{0, 1, \ldots, |X|\}, t^i \in T. \tag{3.3}$$

Considering the example sentence "Barack Obama was born on Aug 04, 1961.", an ER system should identify two mentions of entities: "Barack Obama" as a `Person` and "Aug 04, 1961" as a `Date`. Supposing the tokenization produces the sequence X^I,

$$X^I = [\text{“Barack”}_0, \text{“Obama”}_1, \text{“was”}_2, \text{“born”}_3, \text{“on”}_4, \text{“Aug”}_5, \text{“04”}_6, \text{“,”}_7, \text{“1961”}_8, \text{“.”}_9] \tag{3.4}$$

the expected output is as follows:

$$Y^I = \{(0, 1, \texttt{Person}), (5, 8, \texttt{Date})\}. \tag{3.5}$$

The first tuple correspond to "Barack Obama" (`Person`)—from token 0 to token 1, while the second to "Aug 04, 1961" (`Date`)—from token 5 to token 8.

Typically, ER systems are modeled as sequence labeling tasks [172, 179, 24], where—in the simplest case—each token x_i in the input sequence X is assigned a label $l_i \in L$, where L is defined as follow:

$$T = \{\texttt{I-PER}, \texttt{I-LOC}, \texttt{I-ORG}, \texttt{O}\}, \tag{3.6}$$

`I-`*TYPE* indicates that the token is `I`nside an entity of type `TYPE`, while `O` indicates that the token is `O`utside any entity mention. The output of the ER system is thus a sequence of labels L, that for the above example X^I would be:

$$L^I = [\texttt{I-PER}, \texttt{I-PER}, \texttt{O}, \texttt{O}, \texttt{O}, \texttt{I-DATE}, \texttt{I-DATE}, \texttt{O}, \texttt{I-DATE}, \texttt{O}]. \tag{3.7}$$

The *IO* scheme is the simplest labeling scheme for ER and it cannot distinguish between consecutive entities of the same type [9]. For example, consider the sentence "Barack Obama Michelle Obama" from the keywords of a newspaper, the IO scheme would label it as:

$$L^{II} = [\texttt{I-PER}, \texttt{I-PER}, \texttt{I-PER}, \texttt{I-PER}], \tag{3.8}$$

thus failing to recognize the two distinct mentions of `Person`. More sophisticated and informative schemes exists, such as the *BIO* [287], which introduces the `B-`*TYPE* label to indicate that the

token is at the Beginning of an entity of type `TYPE`. With BIO, the above example would be labeled as:

$$L^{II} = [\texttt{B-PER}, \texttt{I-PER}, \texttt{B-PER}, \texttt{I-PER}], \tag{3.9}$$

thus correctly distinguishing two mentions of `Person`. Refer to Alshammari et al. [9] for a comparison of different annotation schemes for ER and their impact on model performance.

Literature

Early ER systems used rule-based or manually engineered features, rules, or gazetteers. These approaches typically achieved high precision but low recall [179]. Later, machine learning methods enabled more adaptable and data-driven approaches. Effective ER models have been built using hidden Markov models, support vector machines, and conditional random fields (CRFs) [179]—able to consider entire sentences for ER predictions. With deep learning, and representation learning, ER systems further improved. Combinations of deep architectures with CRFs—e.g., BiLSTM-CRF [298, 318] or BiLSTM-CNN-CRF [217]—achieved strong results [179]. The introduction of transformers [361], BERT [90] and its variants, reshaped the field, outperforming previous BiLSTM-CRF models on standard benchmarks [179].

More recently, ER has been approached using large language models with different techniques, such as zero-shot prompting, in-context learning (ICL), and supervised fine-tuning [394]. For example, Wang et al. [371] performs ER via ICL by mapping the task to a text generation problem—i.e., by asking the model to repeat the input text surrounding entity mentions with a prefix and suffix (e.g., "<PERSON> Barack Obama </PERSON>"). Other approaches prompt LLMs to enumerate entity mentions in a structured format without repeating the input text [392], or even to produce code that represents the extracted entities [308, 202]—for instance, by defining a class for each entity type with fields such as the mention string, and then asking the model to output a list of entity objects [308].

The LLM-based approaches have shown strong adaptability, thanks to their zero-shot and few-shot capabilities, reducing the need for annotating large datasets for domain-specific applications [179]. However, LLMs are resource intensive and their adoption is limited by infrastructural costs and latency [179]. Also designing effective prompts is an expensive process requiring domain expertise, and small changes in the prompt can vary the model's behavior, making these approaches not suitable for domains where consistent outputs are required [179].

Other recent studies have explored the use of smaller models, such as DeBERTa [140] and RoBERTa [210] (two improvement of BERT [90]), for zero-shot ER, where the entity types to recognize are provided as textual inputs for the model together with the input text [409, 46]. An example application is to recognize entities of type `Clothing` or `Pet` by proving the strings "clothing" and "pet" in the input. These models achieve competitive performance compared to LLMs, while being more efficient and easier to deploy [409, 46].

Existing Libraries

Finally, several ready-to-use libraries exist for ER, such as spaCy [151], Flair [6], Stanza [279], Stanford CoreNLP [221], and Transformers [383]. These libraries provide pretrained models for multiple languages, and can be easily integrated into production applications.

For instance, spaCy offers efficient and accurate ER models that can be fine-tuned on custom datasets [151]. Its original models use a transition-based architecture [7, 189] in which ER is framed

as a sequence of state transitions over token representations produced by the tok2vec encoder and enriched with sub-word features (e.g., prefixes, suffixes, shapes). SpaCy also provides transformer-based variants in which tok2vec is replaced by a pretrained transformer, as in the `en_core_web_trf` model[1].

3.1.2 Entity Linking

Definition 10 (Entity Linking)**.** Entity linking (EL) is the task of disambiguating and linking entity mentions detected in text to entities in a knowledge repository (KR) or a KB [24, 172].

Giovanni Bianchi [PER] *NIL* , residing in Bologna [LOC] Turin [...] – plaintiff
and
Davide Ricci [PER] *NIL* , residing in Milan [LOC] Milan [...] – defendant
FACTS
The plaintiff summoned the defendant to court, alleging that on April 3, 2022 [DATE] he lent him the sum of EUR 10,000 [MONEY], as evidenced by a private written agreement signed by both parties, with the obligation to repay by December 31, 2022 [DATE], pursuant to Article 1813 c.c. [LAW] Article 1813 c.c. [...]

(a) EL on a judgment.

[25/04/22 – 11:45] [DATE] G. Bianchi [PER] *NIL* :Meet me Friday, 19:00 [DATE], at Bar Centrale [LOC] *NIL* . Bring the papers.

[2025/04/22 – 11:46] [DATE] Davide Ricci [PER] *NIL* : Fine.

[2025/04/22 – 18:51] [DATE] G. Bianchi [PER] *NIL* :
[Voice message – 2 min 58 sec]
Transcription:
[...] You spent my money on that weekend in Milan [LOC] Milan , dinner at Grand Hotel et de Milan [LOC] Grand Hotel et de Milan and even those tickets for the race at the Monza Circuit [LOC] Monza Circuit . [...]

(b) EL on chat logs.

Figure 3.2: EL application for semantic text annotation with clickable links.

An example of EL applied the illustrative legal judgment and chat is available in Figure 3.2.

While entity recognition identifies spans of text that correspond to entity mentions (e.g., "Barack Obama"), entity linking goes one step further by resolving the ambiguity of the mention and associating it with a unique identifier in a knowledge repository (e.g., the Wikidata entity Barack Obama[2] or the English Wikipedia page Barack Obama[3]). This knowledge consolidation process is crucial when mentions can correspond to multiple entities or when different surface forms all refer to the same entity.

For example, in the first case, the mention "Lisbon" may refer to Lisbon, Capital of Portugal[4] or to the movie Lisbon (1956 film)[5]. Furthermore, ambiguity can arise also for entities with the

[1] https://spacy.io/models/en#en_core_web_trf
[2] https://www.wikidata.org/entity/Q76
[3] https://en.wikipedia.org/wiki/Barack_Obama
[4] https://www.wikidata.org/entity/Q597
[5] https://www.wikidata.org/entity/Q3203631

same type, such as Lisbon, Connecticut[6], or the other cities named "Lisbon"[7].

While, in the second case, the mentions "Barack Obama" and "President Obama" should both be linked to the entity Barack Obama, Former US President[8].

Similarly to entity recognition, entity linking is part of information extraction, and contributes to semantic text annotation (STA), knowledge base population (KBP), and downstream tasks like entity-based semantic search. In STA, besides highlighting entities identified with ER, linking them to a knowledge repository allows creating clickable hyperlinks that let users browse directly to entity pages (e.g., Wikipedia or Wikidata). Furthermore, in entity-based search applications, EL enables more powerful facets and filters based on the entries in the KR instead of on the mention surface forms.

entity linking is also applied in question answering and information retrieval (IR). Some QA systems rely on entity linking to understand the entities mentioned in the question and retrieve relevant information from a KR or a KG [199, 368, 98]. Similarly, in information retrieval, linked entities support *query understanding* [137]. In some cases, entity-based queries can be fully reduced to an EL task, while in others, EL serves as a bridge to access the relevant entity facts, as illustrated in Figures 2.1 and 2.2.

Formalization

Formally, given a set of entity mentions $M = \{m_1, m_2, \ldots\}$—for instance, detected by an ER system—and a knowledge repository $\text{KR} = \{e_1, e_2, \ldots\}$ containing entities, an EL system produces a mapping:

$$f : M \to \text{KR} \cup \{\texttt{NIL}\}, \tag{3.10}$$

where `NIL` denotes that the mention does not correspond to any entry in the knowledge repository KR. For example, considering the input sentence "Barack Obama was born in Hawaii." and Wikidata as the reference KR, the mention "Barack Obama" should be linked to the entity `Barack Obama` in Wikidata[9]. While in domain-specific scenarios, such as the illustrative legal judgment depicted in Figure 3.1a, the mention "Giovanni Bianchi" may not correspond to any entity in a general-purpose KB like Wikidata, and thus should be linked to `NIL`.

Literature

Traditionally, EL have been addressed by decomposing the task in two main steps [328]:

1. *Candidate generation*, which involves retrieving a limited (e.g., 100 [385]) set of possible entities from the KR for a given mention, to filter out irrelevant entities and reduce the search space, since KR can contain millions of entities.

2. *Candidate ranking*, which involves ranking (or re-ranking if a preliminary ranking is available) the limited set (e.g., 100 [385]) of candidate entities to select the best match. This step can be performed with powerful but resource-intensive methods that are prohibitive to apply to the entire KB [385].

[6] https://en.wikipedia.org/wiki/Lisbon,_Connecticut
[7] https://en.wikipedia.org/wiki/Lisbon_(disambiguation)
[8] https://www.wikidata.org/entity/Q76
[9] https://www.wikidata.org/entity/Q76

Pre-deep learning approaches for EL relied on dictionaries, acronym expansion, or web search engines for the candidate retrieval phase [328]. For instance, dictionaries can be created from Wikipedia, leveraging redirect and disambiguation pages, that, respectively, can be used for entity aliases and to obtain a list of the entities sharing the same name [328]. The expansion of acronyms served to augment the mention surface form for matching with dictionaries or search engines [328]. For example, the mention "PSG" can be expanded to "Paris Saint-Germain" to facilitate linking to the corresponding entity in a KB. web search engines, such as Google (filtering for Wikipedia pages only) or the Wikipedia search engine, have been used to retrieve candidate entities by querying them with the mention in the context and using the top-ranked resulting pages as candidate entities [328].

For the ranking step, instead, both supervised and unsupervised methods have been applied, either considering the mentions to link as independent or jointly disambiguating mentions in a document to exploit coherence among them [328]. These methods used a variety of features, including string similarity between the mention and entity names, entity popularity (e.g., `Lisbon (1956 film)` is less popular than `Lisbon, Capital of Portugal`), the consistency of the entity types with the entity recognition labels, the textual context, or measures of coherence among entities linked in the same document [328].

Some approaches treated EL as a binary classification problem, where each candidate entity is classified as correct or incorrect for a given mention [328]. Other methods ranked candidates using learning-to-rank algorithm or by calculating mention-entity similarities using training-free vector representations [328]. To jointly link multiple entities exploiting coherence, probabilistic graphical models and graph-based methods have been employed [328].

EL can also be treated as a multi-class classification problem, where each entity in the KB represents a class [173]. However, the consequent large number of classes leads to suboptimal performance [324], and this approach does not generalize to unseen entities, that have not been observed during training.

Extending EL to handle entities unseen at training time is crucial for the *novel entity challenge (Ch. 4)* and incremental EL, where NIL entities are later added to the KR. This setting is known as *zero-shot* EL [212] and can be addressed, for example, by learning to represent new entities from brief textual descriptions [212, 385].

Neural Entity Linking Neural EL approaches typically leverage semantic vector representations of mentions and entities, which they compare to estimate a linking score [324]. For encoding the mention in the context, early neural models used techniques such as convolutional encoders [115, 248, 335, 345] or attention between mentions and candidate entities [118, 192], while more recent methods employ recurrent neural networks (RNNs) [187, 133, 223] and later self-attention [212, 385, 398], which has become the preferred technique for EL encoders [324] with several approaches based on BERT [212, 385, 398].

With respect to entity encoding, some methods apply techniques similar to word2vec [230] to learn entity embeddings that lay in the same vector space as word embeddings [238, 397]. Alternatively, when considering a KG, some approaches use knowledge graph embeddings [25, 335], where entities and relations are embedded in a vector space preserving the KG structure [369]. Other models, instead, propose the use of entity typing information for EL [133, 183, 286, 251, 142], with some using the dense embeddings also at retrieval time with efficient approximate nearest neighbor (ANN) search methods [124, 385].

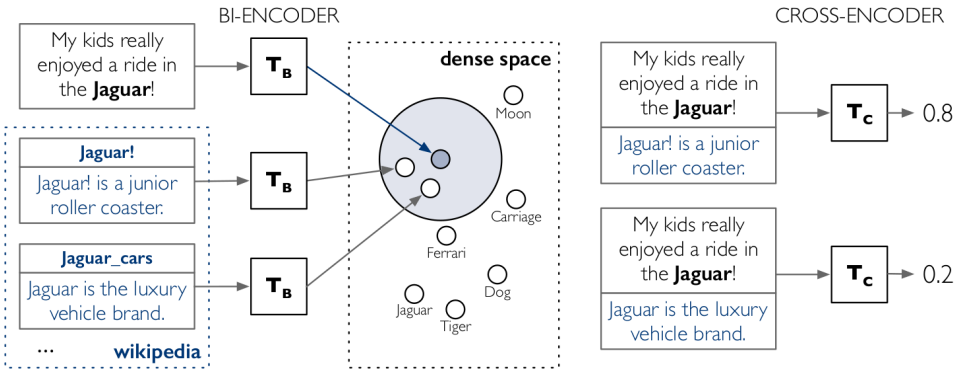


Figure 3.3: Bi-encoder and cross-encoder architectures. Figure from Wu et al. [385]. The bi-encoder encodes mentions and entities independently, enabling fast retrieval. The cross-encoder jointly encodes mention-entity pairs, allowing richer interactions at the cost of higher inference time.

The Bi-Encoder Architecture A notable architecture is the bi-encoder that from its introduction to entity linking [124, 212, 385], has been employed by several studies [175, 142, 39, 95, 21, 269].

The *bi-encoder* independently encodes the mention (in context) and each candidate entity (e.g., using title and a brief textual description) into dense embeddings [385, 156]. A similarity measure between mention and entity can be computed using dot-product or cosine similarity over the vectors. This allows to encode—in advance—all entities in the KR, and—at inference time—to efficiently retrieve the most similar entities given a mention embedding [385] via approximate nearest neighbor (ANN) search algorithms, such as HNSW [220], whose retrieval times scales logarithmically with respect to the KR size.

The functioning of the bi-encoder is exemplified in Figure 3.3, on the left.

Formally, the bi-encoder is composed of two networks, one for encoding mentions T_m, and one for encoding entities T_e. This architecture is commonly referred to also as *dual encoder* or *two-tower* model [124]. When using a transformer like BERT [90], as in Wu et al. [385], the output consists of a vector for each input token and requires a $red(.)$ function to reduce it to a single embedding for the entire input [385, 156]. Example functions are averaging the output embeddings or considering the embedding from the `CLS` token [156], i.e., a special token whose representation from the last layer is used as an aggregate representation of the entire input sentence [90]. Thus, the mention and entity embeddings are obtained as follows:

$$y_m = red(T_m(m)), \tag{3.11}$$

$$y_e = red(T_e(e)). \tag{3.12}$$

Where the input m and e include, respectively, mention and entity information that can be concatenated using ad-hoc tokens. The input formats used in Wu et al. [385] are

$$x_m = \texttt{[CLS]}\ \text{context}_{\text{left}}\ \texttt{[M}_\texttt{s}\texttt{]}\ \text{mention}\ \texttt{[M}_\texttt{e}\texttt{]}\ \text{context}_{\text{right}}\ \texttt{[SEP]}, \tag{3.13}$$

$$x_e = \texttt{[CLS]}\ \text{title}\ \texttt{[ENT]}\ \text{description}\ \texttt{[SEP]}. \tag{3.14}$$

`CLS` and `SEP` are special BERT tokens that are always added at the beginning and at the end of the input sequence. `[M_s]`, `[M_e]`, and `[ENT]` are custom tokens for marking, respectively, the start of the mention (for distinguishing mention and context tokens), the end of the mention, and the start of the entity description (dividing it from the entity title). For additionally exploiting typing information, Heist et al. [142] included, in both templates, a single token after the `CLS` representing the mention and entity types.

A first matching score from the bi-encoder can be calculated as the similarity between the two embeddings, e.g., using *dot-product*:

$$s^{bi}_{m,e} = y_m \cdot y_e \tag{3.15}$$

The bi-encoder networks are trained to maximize the score of the correct mention-entity pair with respect to the remaining entities in the batch [385]. The batches can be randomly sampled or created with "hard negatives", e.g., the ten highest scored entities for linking m that are not the correct entity to which m refers [385]. The loss function for a mention-entity pair (m_i, e_i) appears as follows [385], where B is the batch size:

$$\mathcal{L}(m_i, e_i) = -s(m_i, e_i) + \log \sum_{j=1}^{B} \exp\big(s(m_i, e_j)\big) \tag{3.16}$$

To improve the precision of retrieved candidates, Wu et al. [385] combined the bi-encoder with a *cross-encoder* that jointly encodes the mention and each candidate entity in a single transformer input sequence. By concatenating the mention context, entity title, and description, the model can attend over both components simultaneously and compute a refined similarity score [385, 212, 156]. This setup typically yields higher accuracy than the bi-encoder alone but is computationally expensive, since it must process every mention–entity pair independently. Consequently, it is used only as a re-ranking stage applied to the top-k candidates (e.g., $k = 100$) produced by the bi-encoder [385, 142].

Alternative Formulations of Entity Linking Beyond bi- and cross-encoder architectures, several works reformulate EL as other NLP tasks, such as question answering or text generation. Procopio et al. [276] and Barba et al. [26], for instance, model entity ranking as an extractive QA task, where the model identifies the correct entity span among a set of candidate entities concatenated with the mention context. This formulation enables processing multiple candidates in a single forward pass and allows interactions among them, though it remains computationally demanding for very large KBs.

Alternatively, De Cao et al. [82] proposed to model EL as text generation, performing end-to-end EL, jointly detecting mentions and generating their linked entity names. To guarantee that the generated entities correspond to real KR entries, constrained decoding is applied over a prefix tree built from Wikipedia titles. This approach eliminates the need to store large embedding indexes (tens of gigabytes in BLINK [385]) and allows the model to consider previous links as contextual information. Extending this work, the authors also addressed *multilingual entity linking* fine-tuning a multilingual generative model [83].

End-to-End and Multilingual Entity Linking Recent approaches focus on efficient, end-to-end EL systems that can both recognize and link entities within a single forward pass [21, 269]. Unlike traditional pipelines, these models avoid separate re-ranking stages and instead learn to detect mentions and predict their linked entities jointly. Mention detection typically follows the BIO

scheme [287], and mention representations are derived either by averaging token embeddings [21] or using lightweight feed-forward layers [269]. Linking then proceeds as in bi-encoders, by comparing mention and entity embeddings through similarity search, often with approximate nearest neighbor (ANN). ReFinED [21] also supports *NIL prediction* by including NIL examples during training, while the model proposed by Plekhanov et al. [269] targets a multilingual setting.

LLM-Based Approaches None of the previously discussed models rely on billion-parameter LLMs. Recent work explores using them either as knowledge sources or as reasoning components for EL. Some methods employ LLMs to construct richer entity profiles [307], generating textual descriptions to improve similarity-based retrieval. Others integrate LLMs directly into the linking pipeline. Xiao et al. [390] fine-tune LLaMA-7B [357] for constrained generation of entity names, while Zhou et al. [421] apply LLaMA-2 [356] in the re-ranking stage, selecting the correct entity—or "None of the Candidates"—for each mention.

Prompt-based approaches [209, 92, 123] further reduce fine-tuning requirements. Liu et al. [209] use in-context learning to select entities directly from candidate lists, while Ding et al. [92] combines traditional candidate generation (via priors and BLINK [385]) with LLM-based augmentation and final entity selection. Geng et al. [123] use task-specific grammars to constrain generation, guiding LLMs in NLP tasks such as EL re-ranking without fine-tuning.

Overall, these methods achieve higher accuracy and better domain adaptability [209, 92, 421, 390]. However, efficiency remains a concern: among the discussed approaches, only Xiao et al. [390] explicitly address it by reducing the number of model calls. Nonetheless, LLMs are substantially larger than those of Ayoola et al. [21] and Plekhanov et al. [269], and cannot perform document-level EL in a single forward pass, making their application costly for large-scale or batch annotation tasks.

3.1.3 NIL Prediction and Clustering

As anticipated in Section 2.2.1, knowledge repositories and knowledge bases suffer from incompleteness, and one of the use of information extraction is to automatically populate them for instance, with new movies extracted from newspapers. This task is known as knowledge base population (KBP) [24]. A component of KBP is the identification of novel entities, missing from the KR, and their insertion into it, so that they become available for linking in subsequent documents (entity linking); we refer to this task as *incremental entity linking (IncEL)*.

Since populating a KB can include also the identification of relations between KB entities, in this work we use *incremental entity linking (IncEL)* for referring the task of identifying NIL entities, adding them to the KB or KR, and using them for linking new documents. An illustration of this task is depicted in Figure 3.4.

The entities missing from a KR are usually referred to as *NIL entities* [24], where "NIL" may stand for "not in lexicon" [157]. Another interpretation is that it simply refers to the English word "nil" that means "nothing" [59].

Definition 11 (NIL Prediction)**.** Given an entity mention m and a reference KR, *NIL prediction* is the subtask of entity linking that identifies whether m refers to an entity that does not exist in the KR, i.e., a *NIL entity* [328].

Through this work we will also speak of *NIL mentions*, referring to mentions of NIL entities.

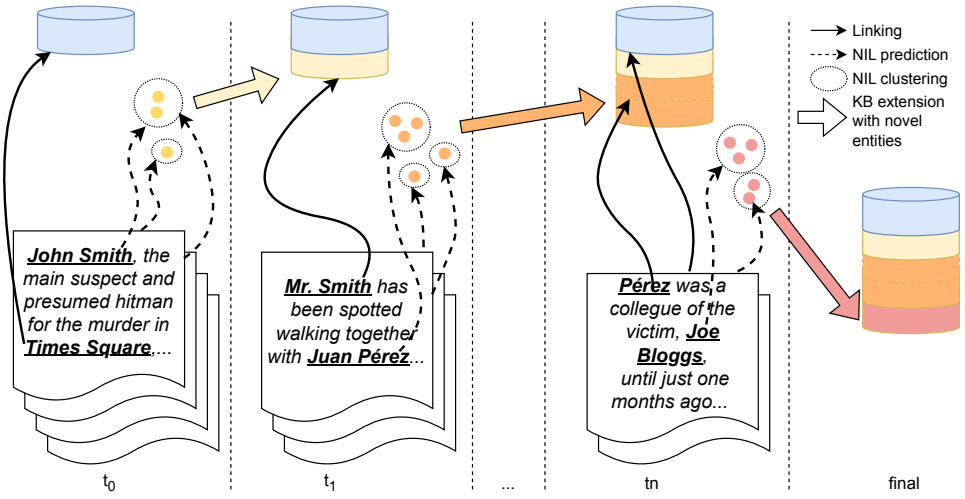


Figure 3.4: In incremental entity linking (IncEL), NIL entities are identified and added into the KR and used for entity linking incoming documents.

This task has also been called with different names: "unlinkable mention prediction" [328], "out-of-knowledge-base mention discovery" [95], "out-of-knowledge-graph entity detection" [235], "unknown entity discovery" [175], and "emerging entity discovery" [147].

Zhou et al. [421] distinguishes between NIL prediction and "none of the candidates" detection, arguing that NIL means the referred entity is not in the KR, while "none of the candidates" means the referred entity has not been retrieved, without assuming it is not present in the KR. However, in practice, NIL prediction is often performed by predicting whether the top-ranked entity, returned by the EL ranking step, is correct or not [328]. It is impractical and prohibitive to compare a mention with every entity in a KR for ensuring the mentions is NIL. Thus, in this work, we consider the practical assumption that if a mention does not refer to the top-ranked candidate or candidates, returned by the EL system, the mention is NIL. Under this assumption, there is no distinction between NIL prediction and none-of-the-candidates prediction.

For addressing the incompleteness problem of KRs, once identified, NIL mentions that refer to the same NIL entity can be clustered to gather more contextual information, useful for obtaining a better representation of the new entity. Indeed, Kassner et al. [175] achieved better results when constructing the entity representation from clusters of NIL mentions with respect to just using individual mentions.

Definition 12 (NIL Clustering)**.** *NIL clustering* is the task that aims at clustering NIL mentions that refer to the same NIL entity [165].

This task is closely related to *coreference resolution*, the task that identifies whether two mentions corefer—i.e., they refer to the same entity [172]. However, coreference resolution typically addresses anaphoric expressions as well [172], for instance by grouping pronouns together with their corresponding explicit mentions.

NIL prediction has frequently been overlooked in EL systems, as many approaches exclude NIL mentions from their evaluation datasets and eventually leave this task to future work [385]. In fact, among the 55 approaches compared by the survey Sevgili et al. [324] only 11 addressed NIL prediction.

Early work on NIL prediction dates back to 2006, with Bunescu et al. [56] using a threshold-based approach. Later, the Text Analysis Conference (TAC)[10] fostered the research in NIL prediction by introducing this task in its knowledge base population (KBP) track starting from 2009 [227]. From 2011, participants were also required to cluster together those NIL mentions that referred to the same entity (NIL clustering) [165]. Organized by the U.S. National Institute of Standards and Technology (NIST)[11], TAC aimed to advance research in NLP and related areas by offering large-scale test collections and standardized evaluation procedures. Since 2008, it has been held annually, inspiring a variety of studies on NIL prediction.

Also, from the 2015 edition, NIL prediction and NIL clustering have been included in the Named Entity rEcognition and Linking (NEEL) challenge [299], which focused on microposts, i.e., posts from social media platforms such as X.com[12].

Approaches to NIL Prediction.

Entity linking with NIL prediction can be framed as a classification problem with a reject option. The classes corresponds to the entities from the KR, while the reject option is NIL [324]. According to Sevgili et al. [324], four main strategies are typically used:

1. Assigning NIL whenever candidate generation produces no possible entities.
2. Applying a threshold to the linking score, below which the mention is treated as unlinkable.
3. Introducing NIL as an additional candidate during the ranking stage, effectively creating a new class in the classification problem.
4. Training a separate binary classifier on mention-entity pairs, often with extra features such as the top linking score or ER labels, to decide whether a mention is linkable or not.

Threshold-based variants remain prevalent [37, 168, 350], while explicit NIL classes [53, 289, 203] or binary classifiers with auxiliary features [238, 399], other than the linking score, have been explored to improve robustness. Joint formulations combine NIL prediction with ER or document-level coherence [223, 110].

Approaches to NIL Clustering.

Early clustering methods relied on surface-form similarity (e.g., Jaccard or edit distance) and hand-crafted rules [111, 205, 400]. Later systems employed feature-based or embedding-based clustering techniques [424, 12, 42], sometimes organized into multi-stage pipelines combining grouping, splitting, and merging operations [236].

Recent work leverages dense mention embeddings for cross-document entity coreference [211], and joint NIL prediction and clustering approaches use both mention and KG embeddings to cluster NIL entities [235].

[10] https://tac.nist.gov/
[11] https://www.nist.gov/
[12] https://x.com/

Incremental and NIL-aware Models.

The problem of dynamically updating a KR motivated incremental EL frameworks [272, 175]. Pozzi et al. [272]—one of the contributions of this Ph.D., related to Section 4.1—introduced a benchmark dataset, an evaluation methodology, and a baseline pipeline for incremental entity linking. Kassner et al. [175] similarly divide data chronologically, performing threshold-based NIL prediction and clustering mentions and entities together by embedding similarity. Other datasets include TempEL [408] and NILK [159], both derived from Wikipedia and Wikidata snapshots. Subsequent work proposed end-to-end EL approach supporting NIL prediction [21] or extended the dual-encoder architectures [385] with NIL-aware mechanisms [20, 142, 422, 95], which add NIL as a candidate or threshold output. Graph-based methods [3] link mentions and entities through similarity-weighted graphs pruned by heuristic constraints, while joint NIL prediction and clustering formulations [235] combine bi-encoder and KG embeddings to rank candidates and cluster NIL mentions by fusing the two representations. More recently, LLM-based models perform NIL prediction as multi-choice question answering, including "none-of-the-candidates" as an option [421, 92].

Overall, NIL prediction has evolved from heuristic thresholds to embedding-based and joint models, while NIL clustering progressed from surface-form similarity to dense, cross-document approaches. Yet, several EL systems still neglect NIL entities, despite their relevance in knowledge-intensive domains such as legal or investigative contexts. This limitation motivated the work described in Section 4.1, which introduced dedicated evaluation methodologies for incremental entity linking, as well as a baseline pipeline, and a benchmark dataset.

3.1.4 Entity Extraction in the Legal Domain

Most prior work on entity extraction (EE) in the legal domain has primarily addressed the entity recognition component. Early ER systems relied on handcrafted rules and statistical models [64], including CRFs [188]. Later work adopted BiLSTM+CRF architectures for Brazilian and German legal corpora [64]. With the advent of transformers [361] and BERT [90], domain-adapted models such as LEGAL-BERT were released [67]. Subsequent comparisons [180], nevertheless, reported that LEGAL-BERT achieves performance similar to simpler neural models (LSTMs, CNNs) on ER.

Only a few works have addressed EL on legal texts. One early study applied ER and EL to European Court of Human Rights decisions [61], using a legal ontology aligned with YAGO [219, 149, 40, 343]. Another focused exclusively on EL over EUR-Lex articles [103] using transfer learning. Hybrid pipelines have also been explored: Tamper et al. [349] combine BERT-based and rule-based ER with an off-the-shelf EL service using popularity-based disambiguation for Finnish court decisions. Similarly, Bellandi et al. [31] proposed a rule-based EL component integrated into a broader architecture to manage Italian judgments.

Some work on EL with NIL prediction has been evaluated on historical legal material, such as the 1641 Irish rebellion depositions [185]. However, we are not aware of prior work that jointly studies recent ER and EL approaches, as well as NIL prediction and clustering on recent legal corpora. These gaps motivate the need for systematic evaluation and adaptation of general-domain EE methods to legal and investigative texts, as pursued in Chapter 4.

3.1.5 Entity Extraction in the Italian Language

Multilingual EL systems have also been proposed, including mGENRE [83], which performs end-to-end linking in a generative manner across more than one hundred languages, as well as mReFinED [207] and BELA [269], which adopt encoder-based architectures to achieve efficient and scalable multilingual EL. However, multilingual models often underperform compared to monolingual ones on high-resource languages and domain-specific data, as parameter sharing across languages reduces language-specific capacity [386, 364]. They are also typically larger and more computationally demanding than their monolingual counterparts (e.g., `bert-base-multilingual-cased`[13] with 178M parameters versus `bert-base-cased`[14] with 110M) [90].

With respect to ready-to-use EL tools, DBpedia Spotlight performs both ER and EL against DBpedia [228, 79] also in Italian.[15] However, it is based on earlier generation techniques and has been shown to require substantial improvement to remain competitive with more recent approaches [65].

Overall, while several multilingual and Italian-specific tools exist, their performance across the complete knowledge consolidation pipeline—including ER, EL, and NIL prediction and clustering—remains largely unexplored, despite the relevance of novel entities in this domain. For this reason, this thesis evaluates and adapts ER and EL methods to Italian legal documents, as described in Chapter 4.

3.2 Data Integration Architectures for the Legal Domain

Several architectures have been proposed to integrate and manage information from legal documents, motivated by the fact that legal professionals must process large collections of heterogeneous material under significant time constraints [31, 51, 91, 134, 132].

To assess how existing approaches relate to the challenges of the legal domain, introduced in Section 1.1, we organize the review along five dimensions:

1. generalizability to heterogeneous legal sources;
2. traceability and verifiability of derived knowledge;
3. error-correction and human oversight;
4. scalability;
5. support for downstream information access tasks.

Finally, we introduce a dedicated subsection on architectural choices, where we summarize the main design strategies adopted in the literature—such as the use of microservices, service-oriented integrations, provenance logging, and orchestrators—independently of the tasks they support.

Among the reviewed approaches, three lines of work are most closely related to the present thesis. Breit et al. [51] do not implement a system but articulate architectural requirements for legal data processing, including traceability, provenance, human oversight, and integration of external knowledge. Pérez et al. [265] address heterogeneous investigative evidence through a modular microservice pipeline that integrates multiple extractors. Bellandi et al. [31] propose a platform for

[13] https://huggingface.co/google-bert/bert-base-multilingual-cased
[14] https://huggingface.co/google-bert/bert-base-cased
[15] https://github.com/dbpedia-spotlight/dbpedia-spotlight-model

court documents that combines entity extraction, a graph-based registry, and API-exposed pipelines for analytics and exploration. These works are therefore the nearest reference points; the remainder of this section places them, together with other methods, within the dimensions introduced above.

3.2.1 Generalizability

Different approaches have addressed different document types separately. Amato et al. [10] focus on notarial documents, while Buey et al. [55] extend a similar strategy to public acts and private agreements by modelling their internal structure and specifying which entities to extract from which section. Breit et al. [51] make the generalization requirement explicit, noting that heterogeneity arises even within a single document class because texts are produced by different legal entities with different drafting practices.

A different line of work considers judicial documents and investigative sources. Bellandi et al. [31] address court decisions and procedural documents, integrating extraction services and analytics. Closer to investigative workflows, Szekely et al. [348], Kejriwal et al. [177, 178], and Pérez et al. [265] propose architectures for heterogeneous evidence, including pages from the dark-web and multimedia, organized via ontologies or microservice-based pipelines. In summary, generalizability is a recognized requirement, though most systems target one class of sources rather than a unified handling of heterogeneous legal and investigative documents.

3.2.2 Traceability and Verifiability

Several systems support tracing structured information back to its textual origin, although this is addressed with different depth. Breit et al. [51] introduce a provenance manager that records data transformations for auditing, and other systems in the investigative domain such as Szekely et al. [348] and Kejriwal et al. [177] record the source document for every node and edge in the constructed knowledge graph. Several approaches, although not formulated around traceability as a requirement, still implement mechanisms that implicitly support it: in Eunomos [44, 45], concepts and roles are explicitly linked to legislative fragments, and Bellandi et al. [31] adopt an annotation model that links extracted entities to document spans. Han et al. [134] likewise incorporate a human-intervention interface allowing users to inspect, verify, and adjust automatically derived analyses.

In precedent exploration, Guan et al. [132] favor step-wise interaction over opaque automation, making intermediate steps inspectable. Overall, explicit traceability is directly addressed in a subset of systems, especially those motivated by auditing or investigative constraints, while others achieve only partial or implicit forms of traceability within the scope of their workflow.

3.2.3 Error Correction and Human-in-the-Loop

Human oversight is supported in several systems. Eunomos [44] uses semi-automatic population, with users validating suggested annotations; Boella et al. [45] extend this with human-in-the-loop (HITL) correction for classification and extraction. Breit et al. [51] provide a correction interface for legal permits.

More recent systems couple extraction with interactive control. LegalAsst [134] includes a human-intervention module that exposes automatic prediction for review and adjustment. Guan et al. [132] explicitly present their system as a form of AI assistance rather than automation, employing step-by-step assistance to ensure that decisions remain under human control.

3.2.4 Scalability

Several proposals address scalability and robustness through architectural choices. A common strategy is the adoption of microservices [265, 55, 239, 302, 31, 51], which support distributed execution, modular replacement of components, and incremental extensibility. Lynx [239, 302] additionally employs an orchestrator and a validation controller to coordinate microservices.

Message-based coordination appears in Bellandi et al. [31] and Pérez et al. [265], where pipelines are orchestrated via the publish-subscribe paradigm [105], enabling asynchronous scaling of components. Finally, Szekely et al. [348] and Kejriwal et al. [177, 178] explicitly consider scalability in the implementation choices for their algorithms and services, such the search engine.

3.2.5 Support for Downstream Information Access

Downstream information access use cases are supported to varying extents. Guan et al. [132] provide user interfaces for faceted search and interactive retrieval of precedents; Szekely et al. [348] and Kejriwal et al. [177, 178] present faceted and map-based visualizations for investigative data; Eunomos [44] enables semantic navigation in legislation.

LegalAsst [134] includes graph-based visualizations and decision summaries, while Bellandi et al. [31] offer semantic document visualization enriched with entities and concepts, alongside a UI for analytics. Investigative systems [265, 267] provide dashboards and graph-based visualizations of the communication network; although Pérez et al. [265] include search functionality, neither system supports question answering.

Beyond their specific functional goals, the reviewed systems differ in how they operationalize integration and control. Many adopt modular, service-oriented architectures [55, 265, 239, 302, 31, 51], which facilitate distributed execution, independent service updates, and incremental extensibility. Lynx [239, 302, 31, 265] complements this modularity with orchestration tools that coordinate the execution of services and maintain workflow consistency. Breit et al. [51] explicitly introduce a provenance manager to record data transformations.

Several systems also specify how processed information is represented and stored. Knowledge graphs and graph databases are commonly adopted to represent extracted entities, relations, and legal concepts [10, 177, 178, 31, 44, 239, 51]. In parallel, some systems [31, 44, 265] describe dedicated storage modules for documents and annotations, relying on different back-end technologies such as relational databases [44], NoSQL stores [265, 348, 177, 178], or combining both [31]. Several architectures also integrate indexing or search-engine components [31, 348, 177, 178, 132] to support keyword-based retrieval as well as more advanced access patterns, such as faceted search.

Extraction services usually provide ER, EL, and rule-based annotation modules that populate these backends [45, 265, 31, 91], and their results are often exposed through web-based UIs for visualization or correction [45, 134, 31, 91]. This organization—comprising extraction, storage and integration, and exposure for information access—appears recurrent across several systems, even when their goals or document types differ.

Overall, these architectural patterns converge toward modular systems that can evolve over time, even though many are scoped to specific legal or investigative subdomains. Taken together, the reviewed approaches demonstrate progress toward individual requirements such as generalizability,

traceability, or scalability, but no existing architecture satisfies all of them simultaneously, or to the extent required in this thesis.

Compared with Breit et al. [51], the architecture presented in Chapter 5 extends generalizability to a broader range of heterogeneous documents, including those from investigative contexts. It also supports consolidated knowledge that goes beyond mention-level recognition (entity recognition) to entity-level consolidation through entity linking, NIL prediction, and clustering, thereby enabling entities to serve as more powerful and semantically meaningful interlinks across diverse document collections.

Similarly, with respect to Bellandi et al. [31], our proposal pursues broader generalization and places stronger emphasis on error correction and traceability. Finally, relative to Pérez et al. [265], our architecture emphasizes traceability, verifiability, and error correction and supports entity-level knowledge consolidation, beyond mention-level, for integrating data.

These gaps and patterns in prior work motivate the architecture introduced in Chapter 5, which translates the dimensions identified in this section—generalizability, traceability and verifiability, error correction and human oversight, scalability, and support for downstream information access tasks—into guiding principles. By embedding these principles, the resulting entity-centric architecture is able to handle heterogeneous data from both legal and investigative contexts while remaining compatible with UIs for advanced information access use cases, such as the ones defined in Section 1.1.

3.3 Question Answering with External Knowledge

Definition 13 (Question Answering)**.** *Question answering (QA)* refers to the task of producing a concise and relevant answer to a question expressed in natural language [24].

QA systems may be focused on a specific domain, e.g., supporting only questions regarding law—in this case we speak of *closed-domain question answering*, or designed to support broad domain-free questions, e.g., questions referring to large general-purpose KR, such as Wikipedia [407]. As a consequence, open-domain QA systems tend to be more generalizable [407], thus, in this thesis we concentrate on open-domain QA.

These systems must acquire external information to answer the given question and this information is generally acquired from unstructured textual document collections, structured knowledge bases, or a combination of both [57]. Modern LLMs contain considerable factual knowledge in the parameters that allows them to answer general domain questions [68, 8, 266, 300], but information is constantly evolving and updating LLM knowledge generally requires expensive fine-tuning [372] or introduces the risk of side effects [204, 71].

Two main paradigms exist for allowing LLMs to acquire external knowledge: *input-based* and *memory-based* integration.

Input-based knowledge injection. Here, the external knowledge is concatenated with the question in the model input, either as plain text or dense vectors. This paradigm includes the traditional *retriever-reader* framework, in which a retriever identifies relevant passages and a reader produces the final answer [423]. Sparse retrievers rely on classical IR models such as TF-IDF or BM25 [172], while dense retrievers use semantic encoders (e.g. BERT [90]) fine-tuned to maximize question-passage similarity [174].

Retrieval-Augmented Generation (RAG) [198, 121] applies this principle to generative models: retrieved passages are appended to the question and fed to an LLM that produces the answer, without fine-tuning, via in-context learning. Subsequent works enhance this pipeline through iterative retrieve-generate cycles [326] or self-directed retrieval [18]. Agentic LLMs [388, 401, 268] continue this trend by dynamically invoking retrieval or reasoning tools, then incorporating their results back into the model input.

While these methods reduce hallucination [331], they introduce latency, architectural complexity as multiple components must be tuned and maintained, and inflates the number of input tokens [150, 420]. Moreover, errors can propagate across pipeline stages [193], and synchronizing knowledge updates among modules (e.g. QA and entity-linking KBs) can be non-trivial.

Memory-based knowledge injection. These methods integrate external information within the model architecture, typically through non-parametric memory modules accessed via cross-attention or learned keys [387, 150, 170, 48]. Although such integration allows tighter coupling between retrieval and reasoning, it requires architectural modifications, thus reducing plug-and-play compatibility with pretrained LLMs. *Fusion-in-Decoder* [160] lies between the two paradigms: it encodes each retrieved passage separately and feeds all encoded representations to the transformer decoder, fusing them during generation but still depending on an external retriever.

Knowledge Graph Question Answering. When the external source is a knowledge graph, the task becomes *knowledge-graph question answering (KGQA)*, where answers are derived from factual triples instead of unstructured text [190]. Two main paradigms exist [190]: (i) *Semantic-parsing* methods translate the question into an executable logical form (e.g. SPARQL [278]), which is run on the KG. These methods are interpretable but depend on parser accuracy and require costly annotations of question-query pairs [201, 225]. (ii) *Information-retrieval-based* methods operate with weaker supervision, learning from question-answer pairs. They extract a question-specific subgraph and then rank or generate answers from it using dense encoders, KG embeddings, or graph neural networks (GNNs) [190, 413]. Most implementations are pipeline-based, combining entity extraction, graph traversal (typically 1-2 hops), and neural ranking.

LLMs have recently been integrated into KGQA [264], where the retrieved subgraph is converted into an LLM-compatible format and then verbalized for generation. Retrieval can rely on graph search, GNNs, or with LLMs, e.g., by iterative relation expansion [412, 184]. In these methods, the KG subgraph is first converted into a textual form—typically as verbalized triples (e.g., "subject → predicate → object") [225] or markup-style encodings [264]—and then provided to the model as input. Despite their effectiveness, they still rely on explicit retrieval and verbalization, thus remaining input-based.

Constrained Generation for QA. Constrained generation [60, 316] restricts the decoding process of LLMs to sequences satisfying predefined syntactic or semantic constraints, thus combining generative flexibility with formal control. It has been successfully applied to tasks such as code generation [316], entity linking [60], and information retrieval [38]. Recent extensions to QA guide LLMs through KG paths during decoding [201, 215]. For instance, DoG [201] alternates constrained triple generation with free-form reasoning in a KG-grounded chain-of-thoughts process, while GCR [215] produces full KG paths and delegates final answering to a larger model.

Yet, they still rely on external entity linking modules and cannot operate efficiently on very large KBs. Chapter 6 addresses this limitation by enabling direct constrained generation on large knowledge bases, without retrievers or pipelines.

Chapter 4

Adapting Entity Extraction Techniques to the Legal Domain

This chapter presents the contributions for *Obj. 1*—through an assessment of existing entity extraction (EE) techniques applied to the legal domain. First, in Section 4.1, we address the *novel entity challenge*—an important problem for the legal domain, since court and investigative documents often mention persons that are not known in public knowledge repository, but whose recognition and linking may enable tasks like precedent case retrieval or the interlinking of heterogeneous documents mentioning suspects during an investigation.

Secondly we study to which extent general-domain NLP approaches for entity extraction work on the Italian legal domain. Initially focusing on Italian judgments from civil courts in Section 4.2 and in Section 4.3, where we evaluate different adaptation strategies for entity recognition. Then, in Section 4.4 we focus on chat logs from an Italian investigation. Indeed, as anticipated in Section 1.1, annotating legal documents with EE systems may be useful for simplifying the analysis of vast amount of heterogeneous documents (challenges *Ch. 1* and *Ch. 2*)—for example, by implementing faceted search applications [17, 141]. Finally, Section 4.5 concludes this chapter.

The main contributions of this chapter can be summarized as follows:

1. introduction of the incremental entity linking (IncEL) task, with a methodology to adapt static entity linking (EL) datasets and the release of a benchmark and baseline pipeline (Section 4.1);
2. evaluation of entity recognition (ER) and incremental entity linking on Italian civil judgments (Section 4.2);
3. study of domain adaptation strategies for ER on Italian civil judgments (Section 4.3);
4. creation of two annotated benchmarks for Italian civil judgments and investigative instant messaging application (IMA) data, and assessment of an entity extraction pipeline on both settings (Section 4.4).

The content of this chapter relates to the following publications: Pozzi et al. [272], Bellandi et al. [32], and Pozzi et al. [274, 271].

4.1 Adapting an Entity Linking Benchmark to Incremental Entity Linking

In this section, we address *Obj. 1*, especially focusing on the *novel entity challenge (Ch. 4)*, by evaluating and adapting existing entity linking techniques to incremental entity linking (IncEL)—which extends EL techniques to an incremental scenario where NIL mentions are identified, clustered, and finally added to the reference knowledge repository (KR) (see Section 3.1.3 for more).

Indeed, entities in domains such as law or healthcare are often unknown in public KRs, such as Wikipedia. Examples are the individuals mentions in court judgments, the suspects of an investigation, or patients in healthcare.

The contributions in this section relate to the Ph.D. publication Pozzi et al. [272] and can be summarized as follows.

1. A general methodology to adapt any EL dataset to incremental EL.
2. A dataset for IncEL, suitable for evaluating EL, NIL prediction, and NIL clustering in the incremental scenario. The dataset was created from an existing EL benchmark [251], applying the proposed adaptation methodology.
3. A baseline pipeline, constructed on the BLINK bi-encoder [385], additionally supporting NIL prediction and clustering.
4. Evaluation procedures for incremental EL.
5. An assessment of the performance and main challenges of the baseline pipeline on the incremental scenario.

The following sections describe, in order, the baseline pipeline with the motivations behind the choice of each component (Section 4.1.1), the procedure to adapt an EL dataset to the incremental setting and its application to WNUM [251] (Section 4.1.2), followed by the definition of an incremental evaluation procedure (Section 4.1.3) an experimental evaluation of the baseline pipeline on the incremental-WNUM (Sections 4.1.4 and 4.1.5), and a discussion of the results (Section 4.1.6).

4.1.1 A Baseline Pipeline for Incremental Entity Linking

The proposed baseline pipeline consists of three main modules: entity linking, NIL prediction, and NIL clustering. As illustrated in Figure 4.1, the process operates as follows: given a mention and its context, the linking module retrieves the most suitable candidate from a *background knowledge repository (BgKR)*. The NIL prediction module then verifies whether this candidate is correct. If so, the mention is linked to the BgKR; otherwise, it is marked as NIL, assuming that the correct candidate is not present in the KR. All NIL mentions in the current batch are subsequently passed to the NIL clustering module, which groups together those referring to the same unseen entity. Each resulting cluster corresponds to a new entity, which is added to the *new knowledge repository (NewKR)* to support future linking tasks. Although the distinction between BgKR and NewKR is not strictly necessary—as IncEL could also be achieved by storing all entities in a single reference knowledge repository—we maintain this conceptual separation for greater clarity.

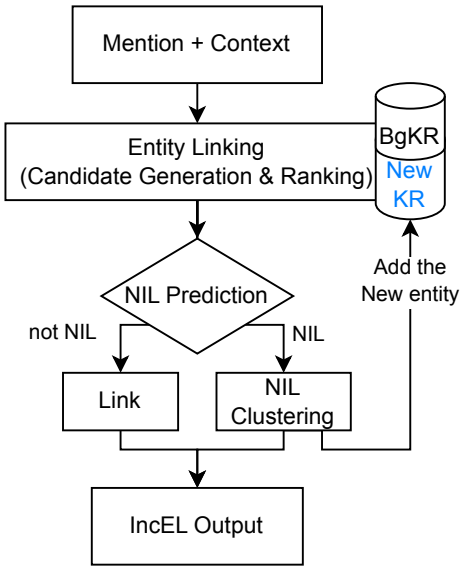


Figure 4.1: Schema of the incremental entity linking pipeline.

Entity Linking The EL module relies on a bi-encoder architecture [156, 385, 124], which retrieves candidate entities through approximate nearest neighbor search and scores mention–entity pairs using similarity measures such as the dot product.

In our experiments, we used the pretrained BLINK [385] bi-encoder, publicly available on GitHub[1]. For the BgKR, we adopted the FAISS [169] index built from the August 2019 Wikipedia dump released by Wu et al. [385], ensuring that entities designated as NIL for the experiments were excluded.

NIL Prediction Given a mention and the scored entity candidates produced by the linking system, NIL prediction determines whether the top-ranked candidate entity is correct or not. According to this prediction, the mention is either linked to the top-ranked entity or marked as NIL. NIL prediction does not alter the ranking produced by the EL module, under the assumption that if the top-ranked candidate according to the EL system is incorrect, than the correct candidate is not present in the KR.

Our implementation employs a logistic regression model whose input features include the score of the top candidate (the dot product between bi-encoder mention and entity embeddings) and the difference between the top and second-best scores (*secondiff*). This configuration was selected through an ablation study, available in Table 4.1. The study explored additional features such as textual similarity between the mention and the entity title—using Levenshtein [197] and Jaccard [161] similarities—and basic statistics of the scores of the top-k candidates—namely mean, median, and standard deviation.

The logistic regression estimates $p \in [0, 1]$, which represents the estimate probability that the top candidate is correct (values close to 0 indicate NIL). If the prediction is NIL, the mention is passed to the NIL clustering step to identify possible coreferent NIL mentions.

[1] https://github.com/facebookresearch/BLINK

F_1	NIL	¬NIL	macro F_1
Max	15.2	82.7	48.9
Max, Levenshtein, Jaccard	23.5	82.9	53.2
Max, Stats10, Levenshtein, Jaccard	50.6	83.4	67.0
Max, Secondiff, Levenshtein, Jaccard	51.2	83.0	67.1
Max, Secondiff	51.4	83.1	67.3

Table 4.1: NIL prediction feature ablation study on the WNUM [251] dataset, using the training set for model fitting and the development set for evaluating. *max* is the score of the best candidate, *levenshtein* and *jaccard* are textual similarities between the mention and the entity title, *secondiff* is the difference between the best and the second-best score.

NIL Clustering NIL clustering is performed in multiple stages, considering both the mention surface form and its embedding representation. This approach helps prevent errors due to homonymy. For each cluster of mentions, we then obtain a representation of the corresponding entity using the cluster medoid vector.

We evaluated three clustering strategies. The first, GNN_B, applies a greedy nearest-neighbor (GNN) algorithm on the dense mention vectors obtained from the bi-encoder. Each mention m is clustered with others whose similarity with m exceeds a predefined threshold. This method, combined with various vectorizers, has shown promising results in Logan IV et al. [211].

The second method, GNN_F, also relies on GNN but uses a feature-based vectorizer following Shrimpton et al. [330]. Mentions are encoded as character skip-bigram indicator vectors for surface text and TF-IDF [172] vectors for context.

The third method, referred to as *three-step clustering (3-step)*, follows Monahan et al. [236]. First, mentions are grouped by surface similarity: mentions with an edit distance ≤ 3 (using the Damerau-Levenshtein edit distance [80]) are clustered together, except for short mentions of less than 3 characters that are clustered only if identical. This step groups similar mentions but may combine semantically different entities (e.g., `Lisbon (1956 film)` and `Lisbon, Capital of Portugal`). Next, we apply single-linkage agglomerative hierarchical clustering [304] within each group based on the semantic similarity calculated with the embeddings from the bi-encoder, to separate polysemous mentions. Finally, each sub-cluster is represented by its medoid vector, and semantically close sub-clusters are merged based on their medoid similarities, with another single-linkage agglomerative hierarchical clustering pass.

All clustering thresholds were tuned via grid search so that the number of predicted clusters on the development set roughly matched the number of unique entities—as in Logan IV et al. [211].

Novel Entity Representation New entities lack meaningful textual descriptions, which are typically required by the bi-encoder for representation. However, by leveraging the dense mention embeddings, we can derive a vector representation for a new entity directly from its mentions. In this sense, the entity representation is inferred from real-world usage examples.

We evaluated three strategies for representing new entities from the clusters of NIL mentions, using respectively:

- the *first* mention encountered,
- the *medoid* vector of all mentions in the cluster at insertion time,

- or including *all* the mention vectors in the KR, treating them as separate entries. In this case, multiple entries may correspond to the same entity.

In this experiment, all the mentions from the AIDA [406] training set have been added to the KR according to the three strategies. Table 4.2 reports the recall values obtained by the bi-encoder linking the mentions from AIDA test$_a$, for each strategy, together with the corresponding time and storage requirements, as measured on the AIDA dataset [406]. Although including all mention vectors yields the highest recall, we adopt the *medoid* strategy as a compromise between efficiency and effectiveness. It achieves comparable performance while requiring only 5% of the time and 22% of the storage compared to the *all* strategy.

Table 4.2: Recall@k EL results between different representation strategies for NIL entities, with retrieval time and the disk usage. The knowledge repository has been built using the mentions in the AIDA [406] training set, while the results refer to linking the mentions from AIDA test$_a$. The indexes were created using FAISS [169] IndexFlatIP class[1].

	R@1	**R@3**	**R@10**	**R@30**	**Vectors**	**Time (s)**	**Disk Usage (MB)**
First	73.7	85.6	91.9	95.8	4022	2.4	16
Medoid	79.5	91.1	95.9	98.7	4022	2.4	16
All	93.9	96.7	98.5	99.3	18319	44.1	72

[1] https://faiss.ai/cpp_api/struct/structfaiss_1_1IndexFlatIP.html

4.1.2 Benchmark Adaptation Procedure: EL to IncEL

To evaluate our pipeline in a realistic setting, the chosen test dataset should reflect the characteristics expected in real-world scenarios. In particular, it should satisfy the following conditions:

1. The entity frequency distribution should exhibit a long right tail, where a few entities appear very frequently, while the majority are mentioned rarely.
2. Entities in the training and test sets should belong to the same domain. Indeed, domain adaptation is not in the scope of this section, which focuses on the *novel entity challenge (Ch. 4)*.
3. The dataset should be sufficiently large to train data-intensive models and to produce a test set that can be split into multiple batches.

While the first condition may not hold in domain-specific contexts—for instance, in legal or investigative documents, where high-frequency NIL entities unknown to public KRs may occur—we consider it essential when using general-domain datasets, as in this section. Otherwise, marking high-frequency entities as NIL would be unrealistic. Additionally, approaches based on LLMs may exhibit bias, since these models encode substantial world knowledge (see Section 2.3.2), which can simplify the IncEL task for entities that appear frequently in their training data.

Candidate datasets for adaptation to the incremental scenario include AIDA [406], the Zero-shot EL dataset [212], KORE50 [148], TACKBP-2010 [166], and WikilinksNED Unseen-Mentions (WNUM) [251]. For this work, we focus on WNUM because it is freely available, includes the above

characteristics, and links mentions to Wikipedia entities, which aligns with several EL systems [385, 82, 26]. AIDA is freely available too, but it is smaller and lacks ground truth for NIL mentions.

Importantly, WNUM ensures that no mention-entity pair appears in more than one set (train, dev, or test) [251], a desirable property for incremental entity evaluation.

New entities. To simulate the presence of new entities, we randomly flag certain entities as NIL while preserving the ground truth—necessary for NIL clustering and subsequent linking. To this aim we assign a probability $p_{\text{NIL}}(e)$ for each entity e, depending on the number of times e is mentioned in the training set ($\#e$), and according to p the desired proportion of entities to flag as NIL. $p_{\text{NIL}}(e)$ is defined as follows.

$$p_{NIL}(e) = p^{\frac{\#e}{M}} \quad \in (0, 1], \tag{4.1}$$

where M is the median entity frequency in the training set. We use the median rather than the mean because the mean is skewed by the long tail of low-frequency entities. This monotonically decreasing function ensures that entities mentioned only once are more likely to be marked as new, reflecting the real-world scenario in which many entities occur rarely, while only a few become popular over a short period.

In this work, we set $p = 0.1$ to obtain 10% of NIL entities, following [2] and apply a Bernoulli trial with probability $p_{NIL}(e)$ to decide, for each e, whether to flag it as NIL. Entities marked as NIL are removed from the BgKR and treated as unknown. To increase the number of new entities in the development and test sets, some mentions of NIL entities are randomly transplanted from the training set. Both development and test sets are assigned 500 new mentions each to ensure robust evaluation (see Table 4.3). Finally, the test set is split into 10 batches using stratified sampling based on entity frequencies and NIL status. Figure 4.2 displays the entire dataset adaptation process, while detailed per-batch statistics are available in Table 4.4.

Table 4.3: Statistics about the IncEL dataset before and after the *transplant*.

		Total Mentions	**NIL Mentions**	**Total Entities**	**New Entities**
Before	Train	2.2M	(25,744)	86,184	(17,957)
	Dev.	10k	(316)	2,397	(61)
	Test	10k	(307)	2514	(63)
After	Train	2.008M	(25,365)	81,858	(17,619)
	Dev.	100k	(501)	7,105	(214)
	Test	100k	(501)	6,473	(248)

Table 4.4: Per-batch statistics about the IncEL test set: number of mentions, entities, NIL mentions, new entities, and *Prev Entities*: new entities already found in a previous batch (that should be linked to an entity previously added to the NewKR)

	Mentions	**Entities**	**NIL Mentions**	**NIL Entities**	**Prev Entities**
b_0	10k	2.5k	50	37	-
b_1	10k	2.5k	50	35	12
b_2	10k	2.5k	50	44	15
b_3	10k	2.5k	50	41	16
b_4	10k	2.5k	50	38	10
b_5	10k	2.5k	50	40	18
b_6	10k	2.5k	51	46	22
b_7	10k	2.5k	50	41	19
b_8	10k	2.5k	50	40	22
b_9	10k	2.5k	50	44	24
ALL	100k	6.5k	501	248	-

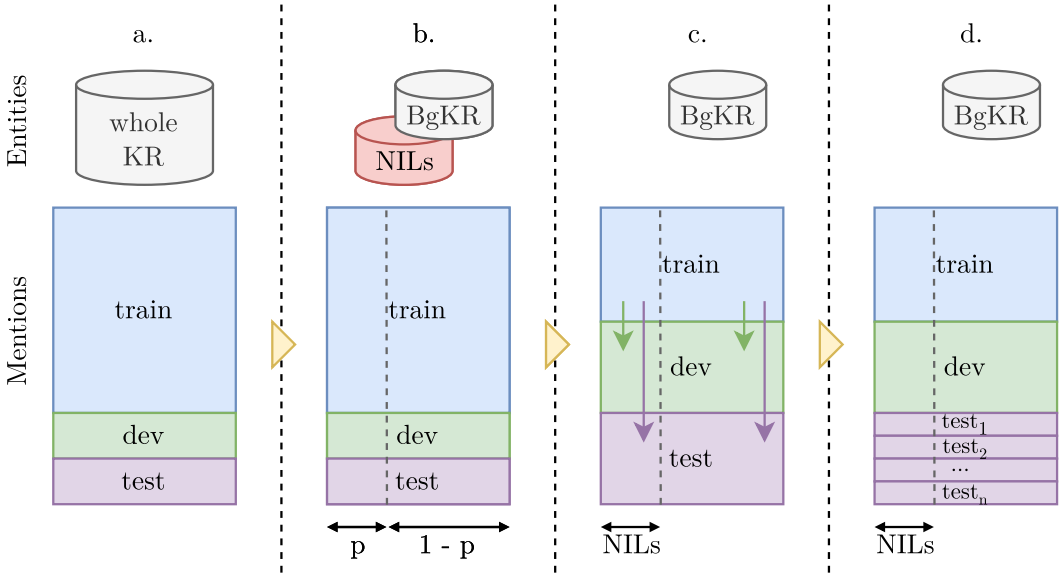


Figure 4.2: Construction of the IncEL dataset from an EL dataset (a): first p% of mentions from the corpus are flagged as NILs and corresponding entities are removed from the KR (b); then observations are just *transplanted* in order to obtain a well-represented distribution for the evaluation (c); finally the test set is split in batches (d).

4.1.3 Evaluation Procedure

To evaluate IncEL, we must consider that the correct handling of a mention m referring to an entity e may depend on previous batches. If e is not present in the BgKR at time t_i, m should be classified as NIL. However, if a previous batch already contained a mention of e, the system should have added it to the NewKR, and m should be linked to e.

The evaluation procedure for the IncEL task should measure both the overall system performance and the performance on subsets of mentions that, in each batch, need to be:

(a) linked to the BgKR;

(b) classified as NIL;

(c) linked to the NewKR.

Finally, to assess the impact of error propagation across batches, the IncEL evaluation should be comparable to a standard single-batch approach. Figure 4.3 summarizes how mentions should be processed depending on the entity they refer to.

Evaluation Measures

Evaluating an incremental pipeline is more complex than evaluating a single model, since errors propagate both through the pipeline and across time (i.e., batches). First, we define measures for the overall IncEL pipeline:

(a) **Link to BgKR:** accuracy for mentions that should be linked to the background knowledge repository:

$$\text{Acc}_{\text{BGKR}} = \frac{TP_{\text{BGKR}}}{N_{\text{BGKR}}} \tag{4.2}$$

where TP_{BGKR} is the number of mentions correctly linked to the BgKR, and N_{BGKR} is the total number of mentions referring to entities in the BgKR, according to the ground truth.

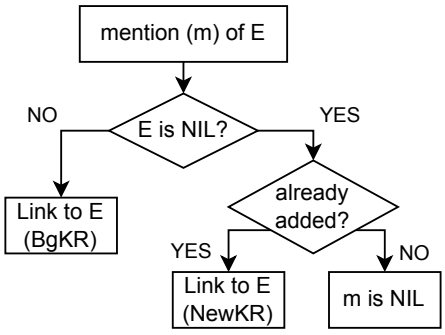


Figure 4.3: Schema of the expected outcome of the IncEL system, given a mention m referring to an entity e, when (a) e is known in the BgKR, (b) e has already been added to the NewKR while processing the previous batches, (c) e is not in BgKR, nor in NewKR.

(b) **NIL:** accuracy for mentions that should be classified as NIL:

$$\text{Acc}_{\text{NIL}} = \frac{TP_{\text{NIL}}}{N_{\text{NIL}}} \tag{4.3}$$

where TP_{NIL} is the number of mentions correctly predicted as NIL, and N_{NIL} is the number of mentions whose ground truth label is NIL.

(c) **Link to NewKR:** accuracy for mentions linked to the NewKR:

$$\text{Acc}_{\text{NEWKR}} = \frac{TP_{\text{NEWKR}}}{N_{\text{NEWKR}}} \tag{4.4}$$

where TP_{NEWKR} is the number of mentions correctly linked to entities previously added to the NewKR, and N_{NEWKR} is the total number of mentions that, according to the ground truth, refer to entities previously added to the NewKR, and thus should be linked to it.

(d) **Overall Accuracy:** accuracy across all mentions:

$$\text{Acc}_{\text{overall}} = \frac{TP_{\text{BGKR}} + TP_{\text{NIL}} + TP_{\text{NEWKR}}}{N_{\text{BGKR}} + N_{\text{NIL}} + N_{\text{NEWKR}}}. \tag{4.5}$$

Secondly, to understand how each component of the pipeline performs on its specific task, we also evaluate the modules separately.

Entity Linking (NEL) For the EL module we report Recall@1, as in Wu et al. [385], which counts a prediction as correct when the top-ranked entity matches the reference one. Formally:

$$\text{Recall@1} = \frac{\#\{\text{mentions with correct top prediction}\}}{\#\{\text{all linked mentions}\}}. \tag{4.6}$$

NIL Prediction For NIL prediction, which is modeled as a binary classification, we compute precision, recall and F_1 on the NIL class:

$$\text{Precision}_{\text{NIL}} = \frac{TP_{\text{NIL}}}{TP_{\text{NIL}} + FP_{\text{NIL}}}, \qquad \text{Recall}_{\text{NIL}} = \frac{TP_{\text{NIL}}}{TP_{\text{NIL}} + FN_{\text{NIL}}}. \tag{4.7}$$

$$F_{1,\text{NIL}} = \frac{2 \cdot \text{Precision}_{\text{NIL}} \cdot \text{Recall}_{\text{NIL}}}{\text{Precision}_{\text{NIL}} + \text{Recall}_{\text{NIL}}}. \tag{4.8}$$

NIL Clustering For NIL clustering we follow the standard coreference evaluation metrics of MUC [363], B^3 [22], and CEAF_e [216], as in Logan IV et al. [211]. Let $K = \{k_i\}$ be the set of annotated clusters and $R = \{r_j\}$ the set of predicted clusters.

MUC [363] measures link recovery:

$$\text{Recall}_{\text{MUC}} = \frac{\sum_{k_i \in K} (|k_i| - |p(k_i)|)}{\sum_{k_i \in K} (|k_i| - 1)}, \qquad \text{Precision}_{\text{MUC}} = \frac{\sum_{r_j \in R} (|r_j| - |p(r_j)|)}{\sum_{r_j \in R} (|r_j| - 1)}, \tag{4.9}$$

where $p(k_i)$ is the set of partitions obtained by intersecting k_i with R (and symmetrically for $p(r_j)$).

B^3 [22] evaluates mention-wise overlap:

$$\text{Recall}_{\text{B}^3} = \frac{\sum_{k_i \in K} \sum_{r_j \in R} \frac{|k_i \cap r_j|^2}{|k_i|}}{\sum_{k_i \in K} |k_i|}, \qquad \text{Precision}_{\text{B}^3} = \frac{\sum_{k_i \in K} \sum_{r_j \in R} \frac{|k_i \cap r_j|^2}{|r_j|}}{\sum_{r_j \in R} |r_j|}. \tag{4.10}$$

CEAF$_e$ [216] aligns clusters one-to-one by maximizing a similarity ϕ. We use the entity-based similarity $\phi_e(k_i, r_j) = \frac{2|k_i \cap r_j|}{|k_i|+|r_j|}$. Let g^* be the optimal one-to-one mapping of clusters (maximizing $\sum \phi$), and let K^* be the subset of annotated clusters that participate in this optimal mapping. Then, we define:

$$\Phi = \sum_{k_i \in K^*} \phi_e(k_i, g^*(k_i)). \tag{4.11}$$

Precision and recall are computed as:

$$\text{Recall}_{\text{CEAF}_e} = \frac{\Phi}{\sum_{k_i \in K} \phi_e(k_i, k_i)}, \qquad \text{Precision}_{\text{CEAF}_e} = \frac{\Phi}{\sum_{r_j \in R} \phi_e(r_j, r_j)}. \tag{4.12}$$

Finally, the F_1 scores are calculated with the harmonic mean, as for NIL prediction, and we calculate their macro average as:

$$F_{1,\text{avg}} = \frac{F_{1,\text{MUC}} + F_{1,\text{B}^3} + F_{1,\text{CEAF}_e}}{3}. \tag{4.13}$$

Note that the handling of a mention m referring to an entity e previously added to the NewKR is considered correct only if it consists of linking m to e, while labeling m as NIL is considered an error. This ensures that the system does not achieve artificially high scores by treating every mention as a new entity.

NIL prediction outcomes can also mitigate EL errors. If the linking is incorrect, the NIL predictor should ideally classify the mention as NIL because the suggested entity is wrong—even if the correct entity exists in the BgKR. Conversely, if the NIL predictor fails to classify a new entity as NIL, it is considered an error.

Finally, if a false positive in NIL prediction creates a new entity while the correct entity already exists in the BgKR, all mentions linked to this incorrect new entity are treated as errors.

4.1.4 Experiments

To enable comparison with other models, we tested our baselines using two experiments. The first experiment evaluates the models on the entire unbatched test set in a *single pass* (∅), performing only one step of clustering.

The second experiment conducts an incremental evaluation over the 10 batches of the test set, which is the main focus of this section and simulates the arrival of new documents to the pipeline. Note that the evaluation procedures differ between the two experiments because NIL clustering adds novel entities to the NewKR that can be linked in subsequent batches.

The first experiment serves two purposes: it provides results comparable to those reported in the literature, and it allows us to estimate the performance drop introduced by the incremental procedure. We evaluated the three baselines, which differ in their clustering approach, using both experimental setups. In Table 4.5, we report only the NIL clustering performance for all baselines, while the remaining metrics are obtained using the top-performing clustering approach (3-step).

Finally, we conducted an additional experiment, called "correct", in which the outputs of previous components are corrected before proceeding through the batches. This setup helps to better analyze error propagation. In Table 4.5, we report the "correct" performance only for the top-performing pipeline (3-step).

4.1.5 Results

In order to investigate which component of the pipeline has a stronger contribute to the final error, we analyzed each component one by one.

The results of the experiments are reported in Table 4.5. The performance of the entity linking module (BLINK bi-encoder [385]) is competitive with more complex models, such as GENRE [82], which achieves $R@1_{total}$ = 74.7 and $R@1_{unseen}$ = 70.4, on mention–entity pairs unseen at training time, values that are very close to ours (excluding error propagation).

As expected, the observed performance drop is mainly due to error propagation in the incremental procedure rather than to limitations of the EL model itself. Indeed, in the "correct" experiment, where errors are prevented from propagating, the performance remains stable across batches.

The NIL prediction component, despite its high precision, suffers from low recall, leading to a large number of false negatives—or *false links*, that is, mentions incorrectly predicted as not NIL (¬NIL) and linked to the KR while actually referring to new entities. Errors from the NIL predictor introduce two critical issues in the pipeline. First, each false positive—or *false NIL*, i.e., a ¬NIL mention incorrectly classified as NIL—creates a spurious entity. This results in redundant representations of the same entity in the NewKR and increases error propagation across batches, since mentions linked to these spurious entities in later steps are counted as errors. For this reason, to mitigate error propagation over time, achieving higher precision is preferable to higher recall, as *false links* count as local errors but do not cause propagation by polluting the NewKR with redundant entities.

With respect to NIL clustering, Figure 4.4 shows the number of predicted entity clusters as a function of their size, together with the expected distribution. The general agreement between the two indicates that the pipeline does not overproduce clusters, and that the high precision yields cluster sizes broadly consistent with the expected values. However, the observed MUC score of 100.0 may be explained by the large proportion of singleton clusters [291].

Among the evaluated approaches, the top-performing one is the *3-step* method, likely because it combines both lexical and semantic similarities between mentions, while the other two (GNN_B and GNN_F) rely exclusively on dense semantic representations.

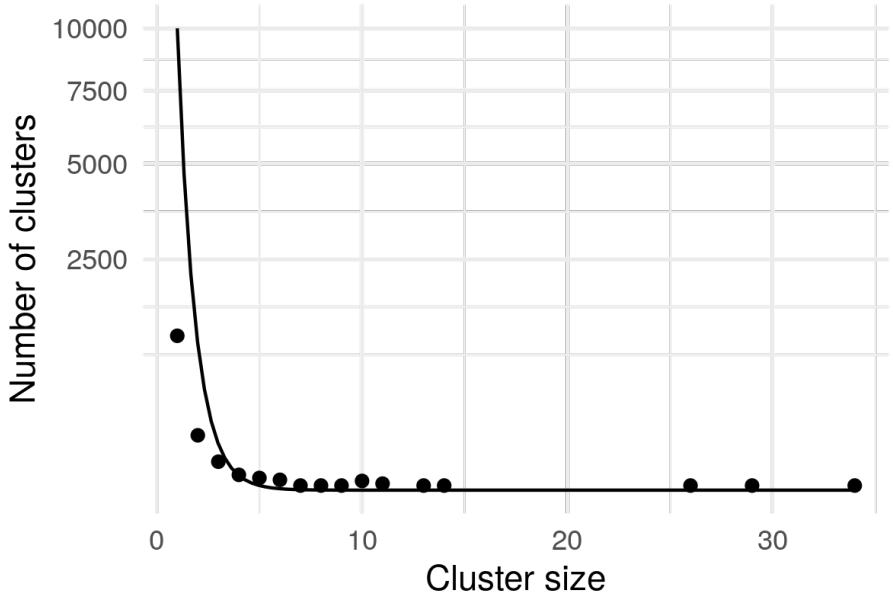


Figure 4.4: The absolute frequency of entity clusters by size (the points) vs the expected values (the curve) from the test-set.

		b_0	b_1	b_2	b_3	b_4	b_5	b_6	b_7	b_8	b_9	$\bar{b}$	$\varnothing$
NEL	R@1	72.64	62.03	55.43	51.68	46.53	45.02	41.71	40.56	39.03	38.39	49.30	72.91
NIL	P	65.16	66.25	70.55	74.30	78.92	78.64	80.00	79.53	81.38	81.96	75.67	64.01
Pred	R	46.74	30.51	30.55	31.47	33.59	34.14	34.32	34.21	35.79	36.10	34.74	46.06
	F_1	54.43	41.78	42.63	44.22	47.13	47.61	48.04	47.84	49.71	50.12	47.35	53.57
NIL Clust: comparison among 3-step, GNN_B, GNN_F: best results highlighted in bold													
3-step	MUC F_1	**96.84**	**95.97**	**95.89**	**96.49**	**96.96**	**97.10**	**96.94**	**97.48**	**97.26**	**97.48**	**96.84**	**97.38**
	B3 F_1	**97.43**	**97.07**	**96.73**	**96.25**	**96.48**	**96.66**	**96.11**	**97.14**	**95.79**	**96.55**	**96.62**	**94.36**
	CEAF F_1	**95.16**	**94.43**	**93.10**	**93.02**	**92.56**	**93.15**	**92.38**	**93.35**	**92.49**	**93.26**	**93.29**	**82.27**
	macro F_1	**96.48**	**95.82**	**95.24**	**95.25**	**95.33**	**95.63**	**95.14**	**95.99**	**95.18**	**95.76**	**95.58**	**91.34**
GNN_B	MUC F_1	86.48	84.22	84.55	87.65	89.23	89.60	89.30	91.07	90.88	91.05	88.40	91.56
	B3 F_1	91.33	90.12	89.07	90.34	90.92	90.74	90.36	91.69	91.72	91.92	90.82	86.46
	CEAF F_1	84.96	83.94	80.62	81.65	80.95	81.58	80.77	82.57	81.38	81.14	81.96	64.61
	macro F_1	87.59	86.09	84.75	86.54	87.03	87.31	86.81	88.44	87.99	88.04	87.06	80.88
GNN_F	MUC F_1	36.08	31.05	31.04	29.84	28.37	28.16	27.67	28.53	26.69	29.17	29.66	65.18
	B3 F_1	71.66	65.16	58.68	56.80	55.03	54.78	53.31	54.40	51.89	51.16	57.29	48.83
	CEAF F_1	62.72	56.91	49.09	47.01	44.90	44.60	43.24	44.22	41.66	40.46	47.48	35.89
	macro F_1	56.82	51.04	46.27	44.55	42.77	42.51	41.41	42.39	40.08	40.26	44.81	49.96
Overall	(a) Link	65.67	56.05	49.68	46.36	41.69	39.87	36.67	35.29	33.99	33.46	43.87	
	(b) NIL	42.00	38.46	51.35	32.35	60.00	50.00	41.18	46.67	44.44	37.93	44.44	
	(c) $Link_{New}$	n/a*	36.36	76.92	68.75	73.33	70.00	70.59	85.00	69.57	61.90	61.24	
	(d) Acc	65.55	55.96	49.72	46.35	41.80	39.96	36.74	35.42	34.10	33.53	43.91	
Correct													
NEL	R@1	72.64	72.11	72.24	72.79	72.26	73.13	71.71	72.77	71.87	72.28	72.38	
NIL	P	55.43	55.17	54.87	54.26	55.52	55.61	55.96	56.21	55.83	54.38	55.32	
Pred	R	43.32	42.43	45.47	43.02	43.57	43.44	42.22	44.91	42.30	43.08	43.38	
	F_1	48.63	47.97	49.73	47.99	48.83	48.78	48.13	49.93	48.13	48.08	48.62	
NIL	MUC F_1	100.0	80.00	–	100.0	66.67	66.67	–	100.0	–	–	51.33	
Clust	B3 F_1	100.0	97.26	100.0	100.0	95.65	97.78	100.0	100.0	97.14	100.0	98.78	
3-step	CEAF F_1	100.0	96.89	100.0	100.0	94.53	96.12	100.0	100.0	95.24	100.0	98.28	
	macro F_1	100.0	91.38	66.67	100.0	85.62	86.86	66.67	100.0	64.13	66.67	82.80	

Table 4.5: IncEL Evaluation results: *EL*, *NIL Pred*, *Overall*, and *Correct* are obtained using the pipeline with the 3-step clustering algorithm (the top performing one). *Correct* results are obtained correcting the output of the previous components. $\varnothing$ refers to the unbatched single-pass experiment. NIL prediction performance are calculated considering correct when an EL error is mitigated. *Nothing can be linked to previously added entities in b_0. "–" represents cases where ground truth clusters contain one element; in this case, *MUC* F_1 score (= 0) is not meaningful [291].

These errors can be mitigated through a human-in-the-loop validation process, where a user, with a dedicated UIs, can merge clusters referring to the same entity or split those that incorrectly group different mentions. This functionality has been developed as part of this Ph.D. and is illustrated in Section 5.5.

4.1.6 Discussion and Challenges in Incremental Entity Linking

The results show that error propagation across batches is a major challenge in incremental entity extraction. A performance drop is especially evident in the EL results. As reported in Table 4.5, the first batch achieves scores comparable to those of the single-pass (∅) and "correct" experiments, but performance progressively degrades in subsequent batches.

As anticipated, a main source of errors lies in false positives (false NILs) introduced during NIL management. This confirms that accurate NIL prediction is a crucial challenge in the IncEL task: lower precision amplifies the propagation of errors throughout the pipeline, ultimately degrading performance.

In our experiments, novel entities are represented by a single vector—the medoid of their NIL mention cluster. However, evidence from Table 4.2 indicates that this representation is sub-optimal, although more efficient. Exploring alternative strategies, such as using multiple vectors per cluster constitutes a promising direction for future work to achieve a better trade-off between effectiveness and efficiency.

Another open problem concerns when to update the KR. In the current setup, the index is updated after processing the first batch in which a new mention appears, but collecting additional observations might yield more reliable representations, possibly guided by a confidence score. Balancing the need for informative representations with the challenge of handling rare entities remains an interesting avenue for further research.

4.2 Entity Extraction from Italian Civil Judgments

In this section, we evaluate the performance of existing entity extraction (EE) systems on Italian civil court judgments. The work presented in this section is related to the publication Bellandi et al. [32].

As discussed in Section 2.4.2, legal data often contain personal information, making compliance with privacy regulations a key challenge. This is reflected in the scarcity of labeled legal data for training or evaluating EE systems [200]. Furthermore, the language used in legal documents differs significantly from everyday language [64], whereas most open datasets employed to train neural models—such as Common Crawl [72] or The Pile [119]—are derived from general web sources.

Another related issue concerns the use of external APIs, such as those offered by OpenAI[2], on legal data containing personal information. Unless the input data are anonymized or pseudonymized, their use may result in transferring sensitive data outside the European Union, raising compliance concerns with respect to the GDPR regulations [107, Art. 44–49]. Consequently, many institutions must rely on locally deployed models, which, while ensuring full control over data storage and security [107, Art. 32], may be constrained by the available computational infrastructure.

In summary, this section makes the following contributions to entity extraction applied to Italian civil judgments:

- Creation of an annotated benchmark dataset for civil judgments, including the calculation of the inter-annotator agreement (IAA), for enabling structured evaluation of entity extraction methods.
- Evaluation of entity recognition and incremental entity linking (IncEL) pipelines on domain-specific data (Italian civil judgments), assessing performance and identifying key challenges.
- Training of the BLINK [385] bi-encoder for the Italian language ($\text{BLINK}_{\text{ITA}}$).

The remainder of this section is organized as follows: Section 4.2.1 describes the creation of the labeled benchmark dataset; Section 4.2.2 presents the Italian EE pipeline; and Section 4.2.3 reports the experimental evaluation.

4.2.1 Annotation of a Legal Benchmark

We started from a corpus of 927,453 judgments in civil trials published from 2008 to 2021 (the majority of them, $\approx 86\%$, have been published from 2016 to 2021). This number can be compared to the total number of trials that are estimated per year according to [76], which falls between 2 and 2.5 millions from 2010 to 2019. For the remainder of this work, we will refer to this corpus or more than 900 thousand civil judgments as *ICCJ900k*—from "Italian Civil Court Judgments".

Data includes the judgment text and 41 metadata with information about the judge (or the president if several judges are involved), the number and year of the judgment and of the trial, the court and the district it belongs to, the instance (trial or appeal), references to the trial in case of appeal, a code describing the subject and additional technical fields. This dataset has been provided by the Italian Ministry of Justice and consists of real documents as they were archived. As a consequence, in many cases, the texts contain spurious lines that must be cleared before

[2] https://openai.com/api/

processing, for instance, some metadata at the very beginning, duplicate headings, extra blank lines or characters from stamps present in the printed version.

The judgments' structure consists of several sections. The most important are the following:

- the preamble with the judge(s), plaintiff(s) and defendant(s) data,
- the description of the case,
- and the final decision and dispositions with the related reasons.

From ICCJ900k, we constructed a benchmark dataset of 146 documents labeled for entity extraction (EE)—which we call *ICCJ146-EE*. The documents have been selected using stratified sampling on the province of the court that made the judgment. The annotation steps was performed semi-automatically according to the following order:

1. the documents have been automatically annotated for ER;
2. human annotators corrected ER errors;
3. EL with NIL prediction methods have been executed on corrected ER annotations;
4. human annotators corrected EL errors. For NIL mentions, they also assigned a novel entity name to each NIL mention so that all mentions referring to the same new entity could be grouped under the same label, obtaining also NIL clustering annotations.

In summary, the annotations include labels for ER, EL, NIL prediction, and NIL clustering (see Section 3.1.3 for details on these tasks).

This approach, where algorithms are used to speed up annotation has been used for labeling ER data in previous work [258], and we can consider this process as a "simulation" of a real *human-in-the-loop (HITL)* use case, where humans manually verify and correct automatic annotations, for enhancing their quality. Furthermore, this "simulation" allows to estimate the amount of time required by the final users-in-the-loop to correct automatic annotations obtained with the EE pipeline described in Section 4.2.2.

Initially, ER annotations have been labeled on all the 146 documents by two annotators. Later, a subset of 30 documents has been additionally labeled for IncEL. We refer to this subset as "ICCJ30-IncEL". The annotation process focused the entities of class `Person`, `Location`, `Organization`, `Money`, `Date`, and `Miscellaneous`.

To obtain an objective measure of the quality of the ER annotations, which served as the starting point for subsequent annotation processes, we calculated an inter-annotator agreement (IAA) measure.

ER Annotation Process An ER label—as described in Section 3.1.1—is composed of the indexes of start and end characters, and by the type of the mentioned entity $(i_{start}, i_{end}, t^i)$. The annotations were carried out by two annotators using *doccano* [243] a web-based application for labeling tasks. The annotator was initially assigned with two disjoint sets of 73 documents. Then, from each of the two set we selected 15 documents and assigned them to the other annotator, so that these $15 + 15$ documents were annotated by both the annotators. In the end, each annotator received 88 documents and the double-annotated 30 were used to measure the IAA.

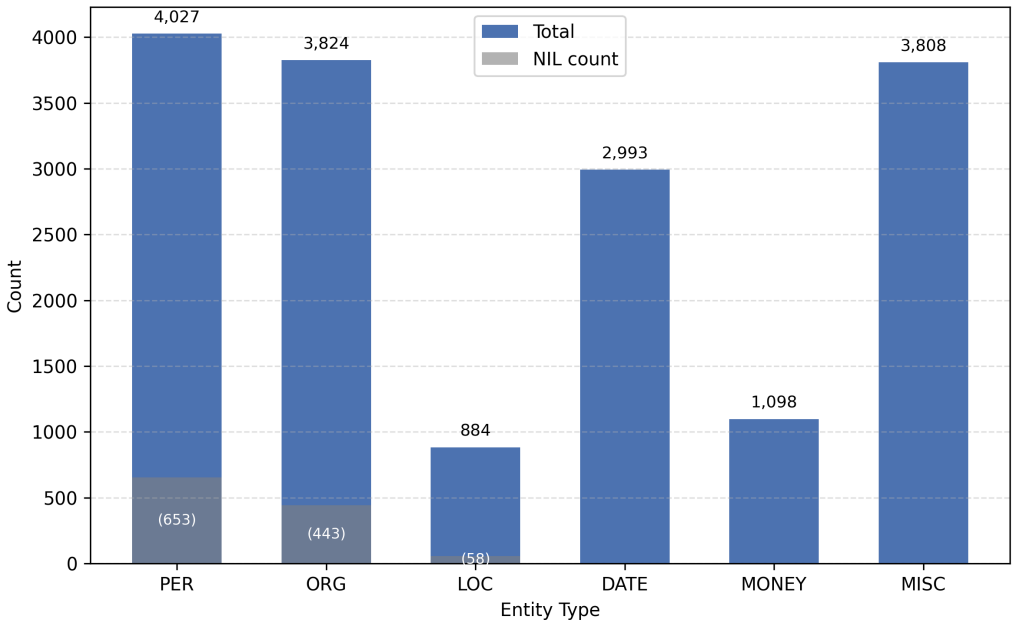


Figure 4.5: Distribution of ER types in the ICCJ146-EE dataset (with NIL counts).

Based on [164], a detailed set of annotation guidelines was prepared and provided to the annotators for reducing the inconsistencies arising from different annotation styles. During the annotation process, the guidelines were iteratively refined by recording and addressing the annotators' doubts and edge cases.

The guidelines cover both span selection and type assignment. For span selection, for example in the case of nested entities, the span corresponding to the most informative and specific entity type should be annotated. "Main Street 18, London[LOC]" should be preferred over "Main Street 18[LOC], London[LOC]".

For type assignment, the entity type should be determined from the surrounding context. For instance, "John Smith, born in France[LOC]" versus "France[ORG] announced new trade regulations".

Each entity type also has its own specific guidelines to handle particular exceptions. For example, titles should be excluded from person names: "Dr. John Smith[PER]" is preferred over "Dr. John Smith[PER]".

Finally, we also decided to annotate mentions such as "the Judge" or "the Court", which are not strictly within the scope of ER but are closely related to coreference resolution, as these entities are particularly relevant in the legal domain.

Document Statistics The resulting corpus counts 16,634 annotations (≈114 annotations per document), with an average document length of ≈1,900 words. Each document required on average ≈15 minutes to be annotated. The distribution of types is reported in Figure 4.5.

Inter-Annotator Agreement To assess the quality of the annotations and the clarity of the guidelines we compute the inter-annotator agreement on the 30 overlapping documents, which are annotated independently by both the annotators.

Inter-annotator agreement measures such as *Scott's* π, *Cohen's* κ, *Fleiss'* κ, and *Krippendorff's* α were developed to correct observed agreement for the portion that may occur purely by chance [360]. In other words, these measures estimate how much of the observed consistency between annotators exceeds what would be expected if both raters assigned categories independently according to their own labeling probabilities [139]. Formally, they compare the observed agreement P_o to the expected agreement P_e under a chance model, and express reliability with the following formula:

$$(P_o - P_e)/(1 - P_e). \tag{4.14}$$

However, these measures may not be suitable for complex labeling tasks such as ER, providing low interpretability values [50]. A problem of these measures for ER is that we deal with text spans, so it is difficult to define a general criterion to identify positive and negative examples to calculate a consistent IAA value [89]. A simple solution to face this problem is to compute the metrics at token-level, however this would yield overly optimistic IAA value. For example, "Barack Obama[PER] was born in Honolulu" and "Barack[PER] Obama[PER] was born in Honolulu" would be considered as a perfect match and the high number of negatives—i.e., the tokens outside of entity mentions—causes the calculation of the metrics on a very imbalanced data [89].

Also, the aforementioned IAA coefficients differ primarily in how the expected-by-chance term P_e is estimated [360], with Fleiss' κ and Scott's π being equivalent when considering two annotators, since the former is a generalization of the latter [139]. Thus, when there are only two annotators and the category proportions are highly unequal—as is common in ER, where most tokens belong to the "outside-of-entity-mention" class—both annotators exhibit almost identical marginal label distributions. Consequently, these coefficients tend to converge toward similar values in such settings.

For these reasons, we adopted the entity recognition F_1 score as the main metric, since it is a more robust alternative to the classic IAA metrics [153, 129]. The F_1 score—calculated with the matching criteria described in Section 4.2.3—is pair-wisely computed between annotators: for each combination, the labels from an annotator are used as the ground truth to evaluate the labels from another annotator; an average is computed in case of more than two annotators.

Table 4.6 shows all metrics averaged over documents. For completeness, we also report the token-level F_1 score for the *strong-typed* match criteria and Scott's π, Cohen's κ, Fleiss' κ, and Krippendorff's α. We can see that *strong-typed (instance-level)*, i.e., the strictest metric, reaches 80.8%, which can be considered satisfactory since it measures perfect matches between annotators. Moreover, by looking at the additional F_1 score-based metrics we can see that the values increase as we relax the matching criterion (partial is the least stringent). This proves that even in cases where the labels from the two annotators do not perfectly match, there is still a significant degree of overlap. These differences between F_1 scores—calculated with harder and softer matching criteria—as well as the difference between *instance-level* and *token-level strong-typed* measures, also point out that a perfect span selection was the most difficult part for the annotators.

Knowledge Consolidation The annotation for the knowledge consolidation process—i.e., EL, NIL prediction and clustering—has been performed semi-automatically from the ER labels of the 30 documents used to compute the IAA. The guidelines, based on the assumptions in [164], are simpler in this case and be summed up with the following two steps.

Metric	Value
F_1 score-based metrics	
Strong	83.9
Strong-typed (instance-level)	80.8
Strong-typed (token-level)	88.6
Approximate	96.6
Approximate-typed	90.4
Partial	96.9
Partial-typed	90.6
Standard IAA metrics	
Scott's $\pi \approx$ Cohen's $\kappa \approx$ Fleiss' $\kappa \approx$ Krippendorff's α	66.2

Table 4.6: Inter-annotator agreement for ER computed on a subset of 30 documents from ICCJ146-EE. The description of each F_1 score-based metric can be found in Section 4.2.3. *based on strong-typed match.

1. If the mention refers to a known entity in the KR (Italian Wikipedia and manually added entities) label it with the entity URI and mark the mention as ¬NIL.
2. Otherwise, mark the mention as NIL, create a new entity, and label the mention with the new entity URI. This latter step is necessary to trace which NIL mentions refer to the same unknown entity (NIL clustering).

To enhance the efficiency of the annotation procedure an application was developed, incorporating both Wikipedia search API and a fuzzy search mechanism. On average documents required ≈15 minutes to be annotated for EL, NIL prediction, and NIL clustering. Out of the 3,006 annotated mentions, 467 refer to Wikipedia entities, and 1,753 to NIL entities. The remaining 786 mentions, of class `Date` and `Money`, were not assigned to any entity.

The 2,200 mentions linked to an entity are organized into a total of 1,025 clusters, with 211 clusters corresponding to Wikipedia entities. The larger portion of 814 refers to NIL entities, as expected within the considered domain.

4.2.2 Italian Incremental Entity Linking Pipeline

Inspired by the IncEL pipeline described in Section 4.1.1 and by recent work on IncEL [142, 175] we develop a pipeline-based system that, first, orchestrates ER and then the knowledge consolidation processes, namely EL, NIL prediction, and NIL clustering. For ER we combine different methods to extract all the types annotated in ICCJ146-EE.

Table 4.7 illustrates the types supported by each ER extractor, as well as by the EL, NIL prediction, and NIL clustering algorithms. Notably, for the latter three tasks, we exclusively consider the types `Person`, `Location`, and `Organization`. This decision is based on the fact that `Date` and `Money` are not directly associated with any linkable entity. With respect to the `Miscellaneous` class, during an exploratory phase, we found that the quality of the prediction for this class—significantly lower than for the other classes—was not promising enough to proceed with the knowledge consolidation process. We further discuss this issue in the experimental evaluation.

Types	Entity recognition				EL	NIL pred.	NIL clust.
	SpaCy	Tint	TrieER*	Combination			
`Person`	✓	✓	✓*	✓	✓	✓	✓
`Location`	✓	✓	–	✓	✓	✓	✓
`Organization`	✓	✓	✓*	✓	✓	✓	✓
`Money`	–	✓	–	✓	–	–	–
`Date`	–	✓	–	✓	–	–	–
`Miscellaneous`	✓	–	–	✓	–	–	–

Table 4.7: Types supported by the ER, EL, NIL prediction, and NIL Clustering algorithms. *Note that the TrieER relies on a gazetteer and it is potentially capable of extracting any entity regardless of its type as long as a pattern is provided. In our experiments, TrieER extracts entities of type `Person` and `Organization`, since its KR (derived from documents metadata) of reference is limited to these two types.

When multiple ER algorithms are employed, it is necessary to combine their extracted annotations and resolve any resulting conflicts. To this end, we introduce heuristic-based combination rules. Both the ER algorithms and the combination rules adopted in this work are described in detail in the remainder of this section.

Entity Recognition

For the ER task, we selected two general-domain, ML-based algorithms for Italian: spaCy [151] and Tint [259]. In addition, we developed a rule-based service, referred to as "TrieER", which performs efficient gazetteer-based lookup using a prefix tree. Each algorithm recognizes a different set of entity types. Table 4.7 reports the correspondence between the entity types in our benchmark dataset and those detected by each algorithm, including their combination.

ER with spaCy is carried out using the Italian transformer model "dbmdz/bert-base-italian-uncased"[3], pretrained by Schweter [317]. The model produces contextual representations that are then processed by a neural transition-based parser (see Section 3.1.1). We fine-tuned the spaCy pipeline, including both the transformer and the parser, on the Italian WikiNER dataset [250], based on Wikipedia articles annotated with `Person`, `Location`, `Organization`, and `Miscellaneous` entities.

Tint [259] is an Italian NLP suite based on Stanford CoreNLP [221]. Its ER module combines CRF-based sequence taggers [188] with rule-based recognizers for `Money`, `Number`, and temporal expressions (`Time`).

TrieER relies on a gazetteer efficiently indexed with a prefix tree (trie) [403], following prior work on tree-based ER [247]. Our gazetteer is built from the document metadata: entity labels are tokenized, enriched with all name permutations (for personal names), and inserted into the trie. The algorithm supports partial matches—e.g., detecting "Smith" when the full label "Jane Smith" exists in the gazetteer. Unlike standard ER, TrieER also provides the link between the matched text and the originating entity (or entities, if shared). This approach leverages metadata fields such as plaintiffs, defendants, and judges to increase the likelihood that these entities are also recognized in the text.

[3] https://huggingface.co/dbmdz/bert-base-italian-uncased

Suppose "Jane Smith" is listed as a plaintiff in the metadata: TrieER searches all token permutations and detects both "Smith" (partial match) and "Smith Jane".

Finally, we merge the annotations produced by the different algorithms. Each ER annotation consists of a text span associated with a type, so conflicts may arise when spans overlap or when overlapping spans have inconsistent types—for example:

> "Mr. Cityville[PER]" (from one algorithm) vs. "Cityville[LOC]" (from another), or "Italy" labeled both as `Location` and `Organization`.

We resolve these conflicts as follows:

1. Span conflicts: overlapping annotations can be total (same boundaries) or partial. In partial overlaps, we select the longest annotation, assuming it provides the most information. As an exception, for entities of type `Person`, we deprioritize spans longer than k tokens, using $k = 6$ after identifying it as the maximum number of tokens for names in ICCJ900k metadata.

2. Type conflicts: different algorithms may predict different types. To handle this, we assign a weight to each algorithm and perform a *weighted majority vote* to select the final type. If multiple algorithms predict the same type, their weights are summed. This mechanism allows us to control each algorithm's influence on the final decision.

Knowledge consolidation

The knowledge consolidation process comprises three tasks: EL, NIL prediction, and NIL clustering. The pipeline is adapted from the English version introduced in Pozzi et al. [272] and detailed in Section 4.1.1. It consists of the BLINK bi-encoder [385], a logistic regression classifier for NIL prediction, and a three-step algorithm for NIL clustering. Although we do not evaluate an incremental scenario with legal documents, we adopt a bi-encoder-based architecture because it supports the representation of new entities—a crucial property for the legal domain.

To enable EL for the Italian language, we trained a bi-encoder following the methodology proposed by Wu et al. [385] for English. We initialized the encoder with weights from the Italian BERT-base model "dbmdz/bert-base-italian-uncased"[4] [317] and fine-tuned it on 9 million mention–entity pairs derived from Italian Wikipedia hyperlinks. Training was conducted for four epochs with in-batch random negatives, using the AdamW optimizer [213] with an initial learning rate of 1×10^{-5} and a batch size of 20. A fifth epoch was then performed using hard negatives—one per sample—selected as the incorrect entity with the highest linking score for the given mention. As the knowledge repository we use the ≈1.5M entities extracted from Italian Wikipedia[5] after filtering out redirects and disambiguation pages. In the remainder of this work, we refer to the resulting model as $\text{BLINK}_{\text{ITA}}$.

For NIL prediction, we use the same Wikipedia-based dataset employed to train the EL model. As in Section 4.1.1, we adopt a *logistic regression classifier* that takes as input:

1. the EL score of the top-ranked entity; and

2. the difference between the top score and that of the second-best candidate (*secondiff*).

[4] https://huggingface.co/dbmdz/bert-base-italian-uncased
[5] https://it.wikipedia.org

The classifier outputs a probability $p \in [0, 1]$, where $p = 1$ indicates that the top-ranked entity is likely correct for the given mention, and $p = 0$ indicates the opposite. In the latter case, the mention is treated as NIL, under the assumption that if the correct entity is not top-ranked, it is not present in the KR.

Finally, NIL clustering follows the three-step procedure described in Section 4.1.1. The clustering thresholds are determined through grid search on the same Wikipedia-based dataset used for EL training.

4.2.3 Evaluating a General-domain Pipeline on Italian Civil Judgments

We evaluate the entity extraction pipeline in the following experimental settings:

1. *Atomic*: we analyze the performance of each algorithm atomically to isolate it from the effects of error propagation. In practice, before analyzing the performance of each task, the prediction from the previous tasks in the pipeline are corrected according to the ground truth. For ER we use the 146-documents ICCJ146-EE as the evaluation dataset and the 30-documents subset ICCJ30-IncEL for the knowledge consolidation tasks. EL, NIL prediction, and NIL clustering are applied, and evaluated, only on annotations of type `Person`, `Location`, `Organization`. This analysis aligns with the concept of human-in-the-loop (HITL) validation, in which the results of each step are refined by humans prior to being utilized as input for the subsequent stages within the pipeline. For ER, we additionally calculate performance metrics on a per-type basis and with different matching criteria, including more relaxed ones that also accept spans that are partially correct (defined below).

2. *EL with NIL prediction*: we study the combined performance of EL and NIL prediction, since these two tasks are closely related, isolating them from ER errors. To achieve this, we execute EL and NIL prediction on ground truth ER annotations from ICCJ30-IncEL.

3. *End-to-end*: we evaluate the performance of our system on each mention in ICCJ30-IncEL considering ER, EL, and NIL prediction—as done, for example, in the NEEL challenge [27]. In this setting, each mention can be either linked to an entity in the KR or classified as NIL. We exclude NIL clustering from this evaluation, since assessing whether a NIL mention is in the correct cluster would overly complicate the analysis; instead, clustering is evaluated separately in the *atomic* setting.

To better study the difficulty of the domain, we evaluate our ER and EL systems also on standard benchmarks in the Italian language: WikiNER [250] and I-CAB [218] for ER; Italian VoxEL [305] and NEEL-IT [27] for EL.

In the remainder of this section, we present the evaluation metrics, the matching criteria for ER, and the results.

Evaluation Metrics and Criteria

For the *atomic* evaluation of ER, we calculate precision, recall, and F_1 score with different matching criteria that determine when an extracted mention is correct with respect to the human-annotated ground truth. Indeed, in some contexts, "partially-correct" annotations might be acceptable—for instance, identifying "A4 Highway" instead of the more specific "A4 Highway Torino-Trieste" might be informative enough in some settings [242, 358]. Furthermore, in a HITL scenario, we prefer to

detect partially-correct annotations rather than to miss them, since a human can quickly correct annotation boundaries.

The adopted criteria—taken from [358] and extended to provide both a typed and untyped evaluation—allow us to assess the behavior of the algorithms at different severity levels. The metrics are defined as follows:

1. *Strong:* the predicted entity has an exact span match with the ground truth annotations.
2. *Strong-typed:* the predicted entity has an exact span match and the type is correct with respect to the ground truth annotations.
3. *Approximate:* the predicted entity is contained in the correct span (or vice versa).
4. *Approximate-typed:* the predicted entity is contained in the correct span (or vice versa) and the type is correct.
5. *Partial:* the predicted entity is overlapping with the correct span.
6. *Partial-typed:* the predicted entity is overlapping with the correct span and the type is correct.

For the sake of clarity, let c denote the chosen evaluation criterion, Y_c^i the number of correctly predicted annotations according to c, Y^i the total number of predicted annotations, and Y_{GT} the number of annotations in the ground truth. Precision, recall, and F_1 score are then defined as:

$$P = \frac{Y_c^i}{Y^i}, \qquad R = \frac{Y_c^i}{Y_{GT}}, \qquad F_1 = 2 \times \frac{P \times R}{P + R}. \tag{4.15}$$

Here, P measures the fraction of correct predictions, R measures coverage over the ground truth, and F_1 represents their harmonic mean.

EL is *atomically* evaluated on accuracy, that is the number of mentions linked to the correct entity divided by the total number of mentions to link, and recall@k. This latter metric assesses whether the correct entity is among the top-k candidates with the highest linking score (in our setting, accuracy and recall@1 are equivalent). Note that EL, and also NIL prediction and NIL clustering, is evaluated solely on the types `Person`, `Location`, `Organization`.

EL is evaluated *atomically* using accuracy and recall@k.

$$\text{Accuracy} = \frac{\text{Number of correctly linked mentions}}{\text{Total number of mentions to link}}, \tag{4.16}$$

$$\text{Recall@k} = \frac{\text{Number of mentions for which the correct entity is among the top-}k\text{ candidates}}{\text{Total number of mentions to link}}. \tag{4.17}$$

The NIL prediction classifier is first evaluated *atomically*, computing precision, recall, and F_1 for both the NIL and ¬NIL classes. Since NIL prediction is tightly coupled with EL, we decouple it by counting as correct also the cases where a mention linked to a wrong entity is classified as NIL: in such cases the top-ranked entity according to EL is incorrect, and the NIL classifier correctly assumes that the true entity is not present in the KR.

NIL clustering is also evaluated *atomically* using MUC, B_3, and $CEAF_e$ as in Section 4.1.3, reporting precision, recall, and F_1 for each. To avoid error propagation, in this setting, NIL clustering is run only on the NIL mentions from the ground truth.

Finally, we conduct a joint evaluation of EL with NIL prediction. Starting from the reference ER annotations and following Section 4.1.3, we compute the accuracy on:

1. mentions that should be linked to the KR—as in (a) of Section 4.1.3,
2. mentions that should be classified as NIL—as in (b),
3. all mentions—as in (d).

These correspond, respectively, to the fraction of correctly processed ¬NIL mentions, NIL mentions, and all mentions.

Finally, for the *end-to-end* evaluation of ER, EL, and NIL prediction, we define two extended criteria: *approximate linking* and *approximate-typed linking*. These criteria consider an annotation correct only if its ER boundaries—and ER type, for typed linking—are correct, it is linked to the correct entity and classified as ¬NIL or correctly classified as NIL. For both criteria, we report precision, recall, and F_1 score.

The hyperparameters for the combination rules were set prioritizing TrieER, followed by spaCy, and then Tint. This choice reflects the reliability of TrieER, which leverages in-domain knowledge from metadata, and the fact that spaCy employs more recent models than Tint. The resulting weights are 0.6 for TrieER, 0.3 for spaCy, and 0.1 for Tint, ensuring that in cases of type conflict, TrieER's prediction—based on metadata—takes precedence.

4.2.4 Results

Model	Dataset	Precision	Recall	F_1
spaCy	WikiNER [250]	91.9	91.9	91.9
GilBERTo [290]	WikiNER [250]	92.7	92.7	92.8
BERTino [240]	WikiNER [250]	-	-	90.4
BERTino Teacher Model [240]	WikiNER [250]	-	-	91.8
Tint	I-CAB [218]	84.4	80.0	82.1

Table 4.8: Results obtained by our ER algorithms (spaCy and Tint) on Italian benchmark datasets compared with recent competitive approaches. I-CAB [218] contains the types `Person`, `Location`, and `Organization`. WikiNER [250] additionally considers `Miscellaneous`.

First, Table 4.8 reports a comparison between two of our ER algorithms, spaCy and Tint, on public benchmark datasets. For reference, we also include the results of two recent competitive approaches: GilBERTo [290] and BERTino [240]. As shown, our spaCy model, which leverages a BERT encoder, performs on par with state-of-the-art systems.

Table 4.9 presents the results of the *atomic* evaluation of ER. As expected, there is a clear gap between the scores obtained under the *strong* and *approximate* criteria, indicating that while our system often detects entities correctly, it sometimes fails to identify their exact boundaries. Such boundary errors can, however, be corrected efficiently through HITL revision. The difference between the *approximate* and *partial* criteria is negligible; therefore, we regard the *approximate* criterion as sufficiently permissive and omit the *partial* one in the remainder of this evaluation.

	Untyped			Typed		
	Precision	**Recall**	**F_1**	**Precision**	**Recall**	**F_1**
Strong	51.9	43.7	47.5	44.7	37.7	40.9
Approximate	86.9	68.0	76.3	64.6	53.1	58.3
Partial	87.4	68.6	76.9	64.9	53.3	58.6

Table 4.9: *Atomic* ER evaluation on ICCJ146-EE, with different matching criteria (rows), typed and untyped. These results are obtained using the combination of the ER algorithms and considering all the evaluated types.

In Table 4.10, we provide a per-type comparison of the *atomic* evaluation for both the individual ER algorithms and their combination. The entity types `Date`, `Money`, and `Miscellaneous` are each handled by a single algorithm—Tint for the first two, and spaCy for the latter. The seemingly counterintuitive differences between the combined results and those of the algorithms responsible for these types arise from boundary conflicts, which may also occur between mentions of different types.

SpaCy proves particularly effective in identifying `Person` and `Location` entities, whereas Tint performs better on `Organization`. The TrieER evaluation yields high precision but low recall, as it can only recognize entities present in its dictionary, which mainly includes persons and a few organizations.

Type	Algorithm	Precision	Recall	F_1
`Person`	spaCy	**92.1**	76.3	**83.5**
	Tint	90.3	62.1	73.6
	TrieER	76.7	34.6	47.7
	Combination	81.5	**79.9**	80.7
`Location`	spaCy	35.4	**90.8**	51.0
	Tint	49.2	82.6	**61.7**
	TrieER	-	-	-
	Combination	**59.9**	60.1	60.0
`Organization`	spaCy	58.2	40.9	48.1
	Tint	65.6	56.5	**60.7**
	TrieER	**90.0**	0.2	0.5
	Combination	34.2	**92.0**	49.9
`Date`	spaCy	-	-	-
	Tint	**83.7**	**55.2**	66.5
	TrieER	-	-	-
	Combination	**83.7**	55.1	66.5
`Money`	spaCy	-	-	-
	Tint	**98.1**	**57.3**	**72.3**
	TrieER	-	-	-
	Combination	**98.1**	56.9	72.0
`Miscellaneous`	spaCy	25.1	**4.1**	**7.0**
	Tint	-	-	-
	TrieER	-	-	-
	Combination	**26.4**	3.9	6.8
Overall*	spaCy	63.8	43.2	51.5
	Tint	74.8	60.1	66.6
	TrieER	**76.7**	11.3	19.7
	Combination	66.0	**67.6**	**66.8**

Table 4.10: *Atomic* ER evaluation on ICCJ146-EE by type and algorithm using *approximate-typed match*. *`Miscellanea` mentions are excluded from the overall calculation. **Note that the type Organization is underrepresented in the knowledge repository used by TrieER, thus only a few entities of this type (i.e., ≈10) have been predicted.

Our combination rules offer a good balance between precision and recall, achieving an F_1 score of 66.8

The results of spaCy and Tint on standard benchmarks confirm the challenging and heterogeneous nature of our target domain. Comparing Tables 4.8 and 4.9, we observe a substantial performance drop when applying ER models to domain-specific data. A closer look at the per-type evaluation (Table 4.10) shows that performance generally decreases across most categories compared to benchmarks. Notably, the `Person` class maintains relatively strong results ($F_1 > 80\%$), while the recognition of `Miscellanea` entities is unsatisfactory—likely due to the distinct nature of

domain-specific `Miscellanea` compared to those in standard corpora. For the remaining categories (`Location`, `Organization`, `Date`, and `Money`), F_1 scores range between 50% and 70%, suggesting that although the general-domain models can handle these types to some extent, domain-specific fine-tuning could yield substantial gains.

The results of the atomic evaluation for the various knowledge consolidation tasks, as well as the joint EL and NIL prediction evaluation, are reported in Table 4.11. The EL module performs satisfactorily, achieving an accuracy of 73.5% and a recall@100 of 90.8%, consistent with results obtained on benchmark datasets which are available in Table 4.12.

Regarding NIL prediction, the classifier effectively identifies NIL mentions but struggles with ¬NIL ones, yielding an overall F_1 score of 64.5%. The main weakness of the consolidation pipeline lies in the NIL clustering stage, which achieves low F_1 scores under both B^3 and $CEAF_e$ metrics.

When jointly evaluating EL and NIL prediction (disregarding ER errors), 79.1% of mentions are correctly processed. Among these, NIL mentions are identified with very high accuracy (91.9%), whereas mentions that should be linked to the KR reach only 46.8% accuracy. This discrepancy can be attributed to a bias in the NIL classifier, which tends to over-predict the NIL class.

Entity linking	Accuracy	73.5		
	Recall@100	90.8		
NIL prediction		**Precision**	**Recall**	**F_1**
	NIL	92.2	86.5	89.2
	¬NIL	58.5	72.0	64.5
NIL clustering		**Precision**	**Recall**	**F_1**
	MUC	71.9	83.9	77.4
	B^3	16.4	60.1	25.8
	$CEAF_e$	7.2	31.7	11.7
EL & NIL prediction		**in KR**	**NIL**	**All**
	Accuracy	46.8	91.9	79.1

Table 4.11: *Atomic* evaluation of the knowledge consolidation tasks and joint evaluation of EL and NIL prediction on ICCJ30-IncEL.

	s-VoxEL-it	**r-VoxEL-it**	**NEEL-IT**
Accuracy	88.9	64.7	69.0
Recall@100	96.8	91.5	-

Table 4.12: EL results on benchmark datasets: strict (s-) and relaxed (r-) version of Italian VoxEL [305] and NEEL-IT@Evalita 2016 [27]. for NEEL-IT, Twitter profile tags (@username) and hashtags (#tag) were filtered out.

Finally, Table 4.13 presents the end-to-end evaluation results using the *approximate linking* and *approximate-typed linking* criteria. The *approximate linking* setup, which ignores the type predicted with ER, yields noticeably better performance than *approximate-typed linking*: the F_1 score reaches 64.0%, nearly eight points higher. This result suggests that once a mention is successfully linked by EL, the linked entity can help refine or even correct the type initially assigned by ER.

	Precision	**Recall**	**F_1**
Approximate linking	59.8	68.9	64.0
Approximate-typed linking	52.3	60.9	56.3

Table 4.13: *end-to-end* ER, EL, and NIL prediction evaluation.

4.2.5 Discussion

In the *end-to-end* evaluation, the system achieved an F_1 score of 64.0% (under the *approximate linking* criterion), meaning that more than one mention out of two is correctly processed across ER, EL, and NIL prediction. Considering the domain's high specificity, the noisy input, and the absence of in-domain training or fine-tuning, these results represent a promising baseline for entity identification and linking.

The main weakness lies in the final NIL clustering stage, which obtained low B^3 and $CEAF_e$ scores. Interpreting clustering metrics is inherently difficult, and prior work has shown wide variability across datasets and evaluation settings [211]. Since suboptimal clustering directly affects the consolidation of NIL mentions, improving this step is crucial.

Also, the NIL prediction component shows a bias toward the NIL class, which could be reduced through domain-specific training. However, in the context of court judgments, this limitation is less problematic, as most key entities (e.g., plaintiffs, defendants, judges, and attorneys) are in fact often NIL.

Performance degradation was also observed in ER models trained on publicly available datasets, with the only exception of the `Person` category. As ER is the first component in the pipeline, its errors propagate to subsequent stages, amplifying their impact. This underscores the importance of domain-specific fine-tuning for ER models—a topic further discussed in the next section (Section 4.3).

Overall, these findings confirm the feasibility of extracting and semantically consolidating entity mentions from Italian court judgments, particularly in settings that implement human-in-the-loop revision of automatically generated annotations. The HITL process used to construct the labeled dataset allowed us to estimate an average of 30 minutes to fully revise a single judgment, corresponding to approximately 320 judgments per month for one full-time annotator. This estimate provides an upper bound on the cost of human-assisted entity extraction.

4.3 Adapting Entity Recognition to Italian Civil Judgments

After assessing the performance of a general-domain EE pipeline on the Italian civil judgments corpus (ICCJ146-EE), we study different adaptation strategies to this domain. This analysis, related to the Ph.D. publication Pozzi et al. [274], highlights which techniques offer the most favorable balance between computational cost and performance gain.

We consider transformer-based spaCy [151] ER models which combines the representations from transformers with transition-based parsers (see Section 3.1.1). We evaluate five different BERT [90] encoders as backbone transformers. These encoders differ in their pretraining data, which we categorize into three levels of specificity: general-domain Italian data (*ITA*), legal-domain data (*LGL*), and the ICCJ900k corpus comprising approximately 900,000 Italian civil court judgments.

Table 4.14 summarizes all five ER backbone configurations, indicating the pretraining data used for each and whether they have been adapted to ICCJ900k with masked language modeling (MLM). Notably, some models (*LGL* and *LGL+ICCJ900k*) have been exclusively trained on legal-domain data, without prior training on generic Italian data.

As a baseline, we use the general-domain *ITA* model, which is available pretrained on HuggingFace as "dbmdz/bert-base-italian-xxl-cased"[6] [317]. Additionally, the *ITA+LGL*[7] [206] and *LGL*[8] models are also available pretrained on HuggingFace.

Finally, all five configurations are fine-tuned for ER with the spaCy library on the annotated ICCJ146-EE corpus, which have been split into three subsets. The 30 documents annotated for IncEL serve as the test set, while 12 randomly selected documents from the remaining set form the development set, leaving 102 documents for training. Table 4.15 presents detailed statistics, including the number of annotations per class.

Fine-tuning is performed using AdamW [213] with an initial learning rate of 5×10^{-5} and early stopping on the development set. To reduce bias from random initialization, we train five models per configuration with different random seeds for the ER layers (see Section 3.1.1). This results in a total of 25 ER fine-tuned models.

Evaluation Settings and Measures For the evaluation we adopt the same experimental settings as in Section 4.2.3—*Atomic*, *EL with NIL prediction* on ground truth ER, and *end-to-end*—and examine the impact of backbone transformers differing in their pretraining data. As before, the EL and NIL prediction components are applied only to mentions classified as `Person`, `Location`, and

[6] https://huggingface.co/dbmdz/bert-base-italian-xxl-cased
[7] https://huggingface.co/dlicari/Italian-Legal-BERT
[8] https://huggingface.co/dlicari/Italian-Legal-BERT-SC

Table 4.14: ER backbones' pretraining domains (one model per column). Model names indicate the order in which the domain data were used for training. Legal domain data used for the LGL adaptation vary: *3.7*GB* legal corpus from the National Jurisprudential Archive; **6.6*GB* legal corpus composed of civil and criminal cases.

	ITA	**ITA+LGL+ICCJ900k**	**ITA+LGL**	**LGL+ICCJ900k**	LGL
ITA	✓	✓	✓	-	-
LGL	-	✓*	✓*	✓**	✓**
ICCJ900k	-	✓	-	✓	✓

Table 4.15: Statistics of ICCJ146-EE. Number of NIL annotations is indicated in parentheses.

	Doc.	Ann.	Person	Location	Org.	Date	Money	Misc.
Train	102	11,940	2,997	612	2,761	2,088	791	2,691
Dev	14	1,688	308	77	369	350	84	500
Test	30	3,006(2,539)	722(653)	195(58)	694(443)	555	223	617
Total	146	16,634(2,539)	4,027(653)	884(58)	3,824(443)	2,993	1,098	3,808

`Organization` by the ER component. Please note that the ICCJ146-EE training set is only used to fine-tune the ER component. All results refer to the ICCJ146-EE test set.

ER is evaluated using the *strong* and *partial* matching criteria from Section 4.2.3. We calculate *precision*, *recall*, and F_1-*measure*, micro- and macro-averaged on the class, and separately for each class.

We also compute the mean and standard deviation of the micro-averaged *precision*, *recall*, and F_1-*measure* across the five random initializations for each transformer. The top-performing model, based on its F_1-*measure*, is then used as the ER component for subsequent evaluations, which replicate the experiments in Section 4.2.3 using the best in-domain fine-tuned ER model. It is important to remind that the EL evaluation and the following ones (NIL prediction, and end-to-end) exclusively focus on the classes `Person`, `Organization`, and `Location`. We calculate the following metrics:

- Accuracy and recall@100 for EL.
- Precision, recall, and F_1 for NIL prediction.
- For EL and NIL prediction, accuracy on (a) *mentions to link*, (b) *NIL mentions*, and (d) all mentions, as in Section 4.2.3.
- For the end-to-end EE, we proceed similarly to Section 4.2.3, using the *strong typed linking* and *partial typed linking* criteria. An annotation is correct if the predicted class matches the ground truth, the span matches according to the criterion, and the mention is linked to the correct entity (if ¬NIL) or correctly identified as NIL. We compute micro- and macro-averaged *precision*, *recall*, and F_1-*measure* for each class, as in the ER evaluation. While previous experiments privileged the approximate criterion, here we use the strong and partial criteria—the strictest and most relaxed—to estimate lower and upper bounds of performance achievable with fine-tuning.

4.3.1 Results and Discussion

Comparison of backbone transformers Table 4.16 shows the results for the comparison of the 5 backbone transformers for ER. Based on the average F_1 across the five random initializations, the encoder that achieves the best results is ITA+LGL+ICCJ900k.

In order to properly analyze the presence of statistical differences based on the choice of the backbone transformer, we conducted an analysis of variance (ANOVA) test on the F_1-measure. The results reveal a highly significant difference (with significance level $\alpha = 0.05$). To further investigate the pairwise differences, we conducted a Tukey's HSD test with a significance level of $\alpha = 0.05$.

Table 4.16: Comparison of the backbone transformers (one per row) for ER on ICCJ146-EE test. Using strong matching we calculate mean (± std) on 5 random initializations.

	Precision	**Recall**	**F_1**
ITA	81.96(±0.76)	83.77(±1.39)	82.76(±0.63)
ITA+LGL+ICCJ900k	**82.08(±0.87)**	**84.69(±0.52)**	**83.36(±0.41)**
ITA+LGL	81.11(±1.00)	83.57(±1.04)	82.41(±0.55)
LGL+ICCJ900k	80.87(±0.73)	82.62(±1.55)	81.72(±0.52)
LGL	79.90(±1.05)	82.62(±1.36)	81.23(±0.47)

Table 4.17: ER evaluation with strong and partial matching on ICCJ146-EE test.

	Strong Match			**Partial Match**		
	Precision	**Recall**	**F_1**	**Precision**	**Recall**	**F_1**
`Person`	90.37	91.00	90.68	95.77	95.43	95.10
`Location`	86.34	84.62	85.49	94.24	92.31	93.26
`Organization`	76.58	80.12	78.31	89.12	92.83	90.93
`Date`	83.49	80.18	81.80	92.12	87.84	89.93
`Money`	96.19	90.58	93.30	99.52	93.72	96.54
`Miscellaneous`	73.97	70.02	71.94	91.27	85.14	88.10
Macro by Class	84.50	82.75	83.59	93.51	91.21	92.31
Micro	82.70	81.74	82.22	92.53	90.97	91.74

We observe that *ITA*, the pretraining on general-domain Italian data, has a positive impact on performance: the models *ITA+LGL+ICCJ900k* and *ITA+LGL* tend to perform better than those trained from scratch on domain-specific data (*LGL* and *LGL+ICCJ900k*).

Surprisingly, the findings suggest that employing a domain-specific legal BERT does not result in a substantial enhancement in ER performance compared to a generic Italian BERT. This observation extends to the adaptation to the corpus of judgments (ICCJ900k) as well. Furthermore, we emphasize that the use of a pretrained generic Italian BERT significantly reduces the effort required for adaptation in terms of time, costs, and environmental imprint.

Entity Recognition The detailed evaluation results for the ER component, as shown in Table 4.17, are promising. All the strong matching measures exceed 80%, and all the partial matching measures surpass 90%, indicating overall proficiency in entity recognition. The classes `Money` and `Person` achieve high recognition rates, surpassing 90% with the strong matching measure. However, the performance for `Miscellaneous` is lower compared to other types. This discrepancy may be attributed to the intrinsic heterogeneity of the `Miscellaneous` class, which exhibits the largest disparity between strong and partial matching performance. A significant difference (approximately 12%) between strong and partial matching outcomes also affects the class `Organization`, highlighting the difficulty in precisely detecting the boundaries of organization mentions.

We also consider the successful results achieved by the ER component indicative of the good quality of our annotated corpus ICCJ146-EE.

Table 4.18: EL and NIL Prediction evaluation on ICCJ146-EE test. $EL_\perp$ and NIL $Pred_\perp$ are independent from other tasks. EL & NIL $Pred_\perp$ evaluate the two tasks independently from ER. *$EL_\perp$ also reports results on VoxEL [305] for comparison.

Entity linking$_\perp$			**NIL Prediction$_\perp$**				**EL & NIL Prediction$_\perp$**	
	Acc	Rec@100		Prec	Rec	F_1	(a) to $Link_{Acc}$	52.95
ICCJ146-EE	73.52	90.81	NIL	92.15	86.51	89.24	(b) NIL_{Acc}	86.31
sVoxEL-it*	88.89	96.83	¬NIL	58.45	72.02	64.53	(d) $Overall_{Acc}$	76.85

Entity Linking and NIL Prediction Table 4.18 reveals that the EL and NIL prediction components do not exhibit the same level of effectiveness as the ER component. The independent evaluation of the EL component ($EL_\perp$) demonstrates a lower accuracy (73.52%) but achieves a recall@100 of 90.81%, suggesting that the integration of a re-ranking system could potentially enhance our results. Additionally, the comparison with the outcomes obtained with the news-based public benchmark VoxEL [305] (available in the same Table 4.18), where our EL model's accuracy reaches 88.89%, further underscores the challenges presented by the domain. We also remind that the EL component has not been fine-tuned on domain data, and that the knowledge repository has not been restricted to domain-related entities. These two factors represent possibilities for enhancing this component.

The NIL prediction classifier ($NIL_\perp$) is effective in recognizing the NIL class, while it suffers with ¬NIL mentions: the low precision of 58.45% highlights that several NIL mentions are wrongly predicted as ¬NIL.

During the evaluation of EL with NIL prediction$_\perp$, we notice the overall accuracy is acceptable (76.85%) and the recall on the NIL mentions is satisfactory at 86.31%. However, we observe that the performance on ¬NIL mentions, which should have been linked to the KR (marked as "(a) to $Link_{Acc}$" in Table 4.18), is not up to the desired standard. The errors for this measure include both mentions linked to incorrect entities and mentions inaccurately identified as NIL. After the NIL prediction, indeed, only 52.95% of the ¬NIL mentions are correctly classified, whereas the accuracy of $EL_\perp$ stands at 73.52%. This substantial 20% decline in performance can be attributed to the prediction of false-NILs.

For these reasons, we consider the NIL prediction to be the most significant challenge in EE. It is important to further study and improve this component in order to enhance the overall performance and reliability of EE systems.

End-to-end Entity Extraction Lastly, Table 4.19 presents the comprehensive results for the end-to-end EE task. `Person` and `Location` exhibit similar satisfactory performance levels. On the other hand, `Organization` entities appear to be more challenging.

Furthermore, the difference between strong and partial matching is limited for `Person` and `Location`, but significant for `Organization`, confirming the difficulty in accurately detecting boundaries for `Organization` entities previously observed in the ER results. Additionally, the relatively modest overall difference of 6% between partial and strong matching, along with the disparity with ER-only results (72.24% vs 91.74%), highlights that the EL and NIL prediction components are responsible for the majority of errors. This observation, combined with the fact that we fine-tuned only the ER component, suggests that fine-tuning the EL and NIL prediction components on the data could potentially enhance the overall performance of the end-to-end EE system.

Table 4.19: EE end-to-end evaluation of `Person`, `Location`, `Organization` mentions on ICCJ146-EE test set.

	Strong Match			**Partial Match**		
	Precision	Recall	F_1	Precision	Recall	F_1
`Person`	76.89	77.42	77.16	80.19	80.86	80.52
`Location`	75.92	74.36	75.13	80.10	78.46	79.27
`Organization`	51.10	53.46	52.25	60.61	63.22	61.88
Macro by Class	67.97	68.41	68.18	73.63	74.18	73.89
Micro	65.39	66.73	66.05	71.53	72.95	72.24

4.4 Entity Extraction from Investigative Chat Logs

Investigators often need to explore and extract insights from large volumes of heterogeneous documents, including instant messaging application (IMA) data. To address this need, we proposed an entity-centric approach to integrate different data sources (*Obj. 2*), for enabling information access applications such as faceted search and graph-based visualizations.

During my Ph.D., I collaborated with prosecutors and judicial police officers from two public prosecutor's offices in the context of two separate investigations. While the work presented in this thesis refers exclusively to the second case—an investigation into suspected corruption—Table 4.20 reports the statistics of the chat logs extracted and processed in both investigations, to better illustrate the volume of data that investigators must handle.

Our work was conducted under an official consultancy agreement within the framework of judicial investigations. All data, due to the private information contained, was stored locally and accessed only by authorized consultants, through a secure virtual private network (VPN) with user-password authentication; no external APIs have been used at any stage.

Accordingly, this section focuses on the extraction and organization of entities from IMA data. Messages—including transcribed voice notes—were modeled within a knowledge graph designed to represent the network of contacts of the main suspect (Chapter 5). An improved version of the entity extraction (EE) pipeline introduced in Sections 4.2 and 4.3 was then applied and evaluated on a labeled corpus of chat logs. This corpus has been created through a procedure similar to the one employed for legal judgments in Section 4.2. As in that case, the availability of labeled data for IMA analysis is limited by the presence of personal and sensitive information.

The work presented in this section is related to the Ph.D. publication Pozzi et al. [271]. Its main contributions, with respect to entity extraction applied to instant messaging application data, can be summarized as follows:

- Modeling of chat metadata in a knowledge graph to support graph-based investigative queries (*UC 3*).
- Creation of an annotated benchmark dataset for investigative IMA data, enabling structured evaluation of EE methods.
- Selection of a suitable speech-to-text model for transcribing voice messages.
- Assessment of entity recognition (ER) and incremental entity linking (EL) on domain-specific data, identifying performance trends and key challenges.

The following sections describe the construction of the knowledge graph modeling chat metadata (Section 4.4.1), the annotation process for obtaining labeled benchmark data (Section 4.4.2), the methodology adopted for handling voice messages (Section 4.4.3), the entity extraction pipeline (Section 4.4.4), and the dataset, experiments, and evaluation procedures (Section 4.4.5).

4.4.1 Metadata Extraction

Investigators usually use forensic software tools to extract the content of a seized smartphones (logical extraction) [108]. In our case, we received chat logs in Microsoft Excel[9] ".xlsx" format.

[9] https://www.microsoft.com/en-us/microsoft-365/excel-c

IMA dumps consist of semi-structured data. In fact, unstructured message corpora are accompanied by structured metadata, which include information such as the timestamp of each message and the contact names of the sender and receiver. These metadata are highly valuable for investigators, as they allows them to reconstruct the suspect's communication network based on the number and direction of exchanged messages. For this reason, in this work we directly leverage chat metadata to construct a KG that models contacts, chats, and messages, respectively through the entity classes `Person`, `Chat`, and `Message`.

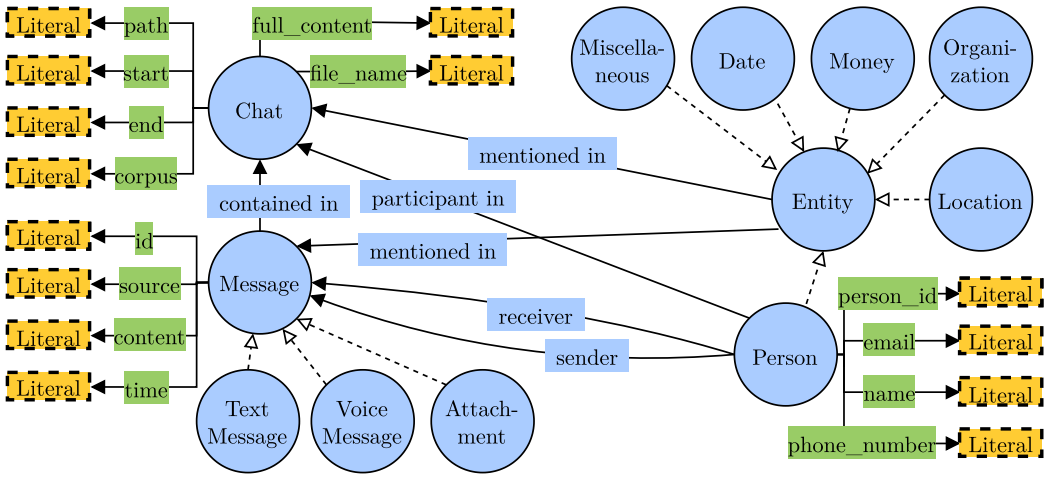


Figure 4.6: Knowledge Graph schema

The entire schema of the *chat knowledge graph* is depicted in Figure 4.6. It incorporates all the metadata present in chat dumps and it is predisposed to host the new entities identified from an IncEL pipeline—namely `Location`, `Organization`, `Date`, `Money`, and `Miscellaneous`, all of which are subclasses of `Entity`, together with `Person`. Finally, the *mentioned in* relation allows persisting the association between entities and the messages and chats in which they are mentioned. Consequently, investigators can trace entities to the source messages and verify the correctness of potential deduction reading the original messages (*Ch. 3*).

We consider the metadata from the dump we received to be applicable to most IMAs. For each chat they include

- the list of participants with their phone numbers,
- the start time,
- and the time of the last activity.

And each message is described by

- the timestamp it was sent,
- the name of the sender,

- the number of the sender,
- and optionally the attachments it contains.

Topic	Chats	Proc. chats	Msgs	Attach. (img-audio-docs)	Persons
1) Fraud	1,133	801	45,252	3,324 (575-304-1,590)	1,365
2) Corruption	1,442	1,442	364,690	63,24 (51,532-6,273-4,066)	2,351

Table 4.20: Statistics about two investigations in which I have been consultant. The work described in this section refers only to the second investigation about suspected corruption.

Table 4.20 shows the resulting statistics of the metadata extraction in both the investigations I was involved. The remainder of this section refers solely to the second investigation about suspected corruption.

4.4.2 Benchmark Annotation

Table 4.21: Number of annotated mentions per entity type in the chat dataset.

Entity type	Person	Location	Organization	Date	Money
Mentions	668	207	268	157	11

We annotated a benchmark dataset for evaluating ER using six chat conversations, three of which are group chats. These were selected among those sufficiently long and accompanied by audio transcripts. We adopted the same semi-automatic annotation procedure used for the judgments, described in Section 4.2.1. Specifically, we first generated automatic annotations with the ER component and then manually revised them using doccano [243]. The final dataset includes 1,311 annotated entity mentions, distributed across five entity types as reported in Table 4.21.

4.4.3 Audio Transcription

While we plan to extend our system to handle additional multimedia content, our experiments focused on audio, as this is the most time-consuming media type according to the investigators we collaborated with. To transcribe audio files, we employed Whisper (Large) [282], an automatic speech recognition (ASR) model developed by OpenAI. This choice followed an empirical evaluation of several ASR systems. We randomly selected 500 Italian audio files and their corresponding validated transcriptions from three sources: M-AILABS [63], CommonVoice [16], and VoxForge [365].

Since these datasets contain high-quality recordings, we artificially degraded the audio to better approximate the conditions of IMA voice messages, which often exhibit environmental noise and compression artifacts. We applied various distortion techniques: Gaussian noise, background noise, speed variation, pitch shifting, delay, and signal distortion.

We evaluated the transcription quality using two complementary metrics: the *Word Error Rate* (WER) [172] and the *BERTScore* [415] (F_1). The WER measures the edit distance between the predicted transcription and the reference, normalized by the reference length [172]:

$$\text{WER} = \frac{\text{Insertions} + \text{Substitutions} + \text{Deletions}}{\text{Number of words in ground-truth transcript.}} \tag{4.18}$$

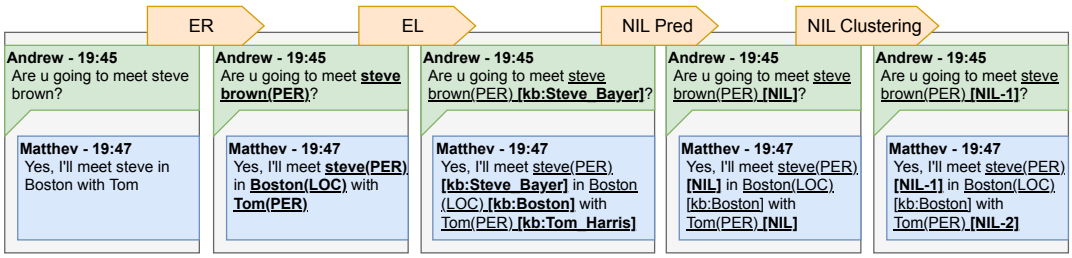


Figure 4.7: IncEL pipeline on chats.

Lower WER values indicate better performance.

BERTScore [415] instead measures the semantic similarity between two sentences based on contextual embeddings from a pretrained BERT model [90]. It computes precision, recall, and F_1 by aligning words in the reference and candidate transcriptions according to cosine similarity in embedding space. Unlike WER, BERTScore captures paraphrastic and semantic equivalence beyond exact word matching.

The best-performing model was Whisper (Large), which achieved a BERT score of 90.8% and a WER of 28.2%.

4.4.4 Algorithms for Entity Extraction

Most pipeline components for entity extraction are the same as in Sections 4.2 and 4.3, while NIL clustering has been improved for this study. Figure 4.7 depicts an example application of the IncEL pipeline on chat messages. In the remainder of this section, we describe each component of the IncEL pipeline.

As the ER component we use *ITA+LGL+ICCJ900k*, the best performing model according to the adaptation study on judgments (Section 4.3). It is based on the library SpaCy-transformers[10] and uses the SpaCy transition-based parser with the contextualized token representations obtained from a transformer [361]. Using this model in the experiments can give us clues on how the best model for judgments (see Section 4.2) performs on investigative IMA data, allowing to understand the feasibility of using a single ER model for both judgments and IMA data.

For EL, we use the bi-encoder architecture of $\text{BLINK}_{\text{ITA}}$ considering the KR obtained from Italian Wikipedia, as in Section 4.2.2, extended with the chat participants from the chat KG. To represent chat participants, we leverage BLINK zero-shot capability to encode an entity given its title and textual description. For each of them, we obtain a representation by giving the following input to the bi-encoder:

```
[CLS] {name} [ENT] phone number: {phone1(,phone2,...)} [SEP].
```

As the majority of the people of interest for the investigation refers to entities not in Wikipedia, we link mentions of persons only to the chat participants. For organizations and locations, instead, we consider Wikipedia entities.

In the NIL prediction component, we employ the logistic regression classifier described in Section 4.2.2. It takes the EL top-ranked score and "secondiff" (the difference between the top-ranked

[10] https://spacy.io/universe/project/spacy-transformers

score and the second-best) and produces a probability $p \in [0, 1]$ as output, where 1 denotes the top-ranked link is correct and 0 assigns NIL.

The entity clustering approach, which has been improved with respect to previous work in Section 4.2, uses a supervised XGBoost classifier [69], trained with the mentions from the labeled IMA dataset, to calculate a similarity score between pairs of mentions. The input features include lexical similarity measures, namely Jaro-Winkler [382] and Jaccard [161], as well as semantic similarity computed as the cosine similarity of the mention vectors obtained from the $\text{BLINK}_{\text{ITA}}$ bi-encoder. The classifier outputs a synthetic similarity score.

Next, the pairwise similarity scores are used to create weighted graphs, where mentions are represented as nodes and the edge weights reflect the mention similarities. The final mention clusters are obtained by applying a community detection algorithm (Louvain method [43]) on these graphs.

Entity clustering is applied at document-level to group the NIL mentions referring to the same unknown entity.

The application of IncEL has two outcomes: the chat file is annotated (semantic text annotation) and the knowledge graph is updated with the discovered entities, which correspond to the entity clusters, using the "mentioned in" relationship (see Figure 4.6).

4.4.5 Experiments

Our evaluation has three main objectives:

1. Demonstrating that the combination of graph-based modeling, multimedia enrichment, and NLP-based entity extraction enhances IMA data analysis in criminal investigations through graph-based querying and semantic search.

2. Assessing the current quality of the proposed solution.

3. Discussing limitations and challenges.

IncEL Contribution to Knowledge Graph Enrichment. The IncEL pipeline successfully processed 1296 out of 1442 input chats, with errors caused by technical issues that can be fixed by splitting the very long chats. Statistics about the extracted entities are reported in Table 4.22: the table indicates the impact of IncEL on enriching the KG with the "mentioned in" relationship, and the informativeness of audio transcriptions, which contain on average four times more entities than text messages, i.e., 0.083 vs. 0.026. The statistics also show that 2,361 mentions of persons have been linked to entities in the chat knowledge graph, which validates the introduction of this in-domain linking mechanism.

Preliminary Insights on the Quality of IncEL Annotations. We discuss a first evaluation of the ER component.

Results shown in Table 4.23 are not satisfactory, confirming that IncEL on IMA data is challenging because of the specific data distributions and suggesting that in-distribution fine-tuning is necessary. On the other hand, metadata are precisely identified enabling accurate filtering with faceted search.

Type	Mentions		Links		Entities
	Text	Audio	to KR	NIL	
`Person`	7765	520	2361*	5404	5701
`Location`	2578	185	1118	1460	1892
`Organization`	1753	113	614	1139	1339
`Date`	916	53	–	–	–
`Money`	124	23	–	–	–
`Miscellaneous`	32	1	–	–	–

Table 4.22: Statistics of the IncEL extraction from chats. *Links for Person refer to the chat knowledge graph, not to Wikipedia.

	Precision	Recall	F_1
Strong-typed	44.0	21.4	28.8
Partial-typed	71.4	35.1	47.0

Table 4.23: ER evaluation on the chat dataset. *Strong* counts perfect matches of span and type. *Partial* counts as correct when there is an overlap between predictions and ground truth.

4.5 Conclusion

This chapter addressed *Obj. 1* by empirically assessing the applicability of general-domain EE methods to legal and investigative contexts. In the first part (Section 4.1), we introduced the incremental entity linking (IncEL) problem, developed a methodology to adapt static entity linking (EL) datasets to the incremental setting, and released both a benchmark and a baseline pipeline. The evaluation showed that incremental settings suffer from error propagation, and that NIL prediction is major source of errors.

In the second part, we evaluated entity recognition (ER) and IncEL pipelines on Italian civil judgments and studied strategies for adapting ER to the domain. Without in-domain fine-tuning, models trained on public corpora degraded on all categories except `Person`, and errors at this stage propagated to subsequent components. This confirms the need for domain-specific adaptation and motivates the human-in-the-loop strategy adopted to improve the quality of machine annotations.

Considering fine-tuning, the combination of in-domain adaptation (via masked language model) and task-specific ER fine-tuning achieved the best results, but simply fine-tuning a general-domain model already proved to be a good compromise between performance and computational cost. These results indicate that the size of our labeled dataset is sufficient to fine-tune effective ER models. A natural direction for future work is to quantify the amount of labeled text needed to reach satisfactory performance for this domain.

Accurate NIL prediction remains a key challenge for entity extraction (EE) in the legal domain, where relevant entities are often not present in public knowledge repositories (KRs), and NIL prediction errors contaminate the KR, affecting subsequent extraction steps. In-domain fine-tuning may mitigate this effect and represents another avenue for future work.

We subsequently extended the analysis to investigative IMA data, creating an annotated benchmark and evaluating an improved EE pipeline. Results indicate that entity extraction in IMA is challenging and could benefit from in-domain fine-tuning, whereas metadata can be reliably

extracted to enable faceted search and other information access functionalities.

These findings motivate, in the subsequent chapters, the design of architectures that remain useful despite imperfect extraction, thanks to traceability and error-correction capability.

Chapter 5

Data Integration Architecture for Knowledge-Intensive Domains

In this chapter we address *Obj. 2* by presenting an entity-centric architectural model for integrating heterogeneous data in knowledge-intensive domains, with a particular focus on the legal domain. Although the work takes inspiration from different projects in the legal domain, the goal is not to reproduce those designs. Rather, we distill a general architecture that can be adopted beyond those specific cases, including other knowledge-intensive domains where entities play a central role in shaping information needs, user interactions, and downstream tasks.

Over the past years I have contributed to several research projects aimed at incorporating artificial intelligence (AI) technologies into the Italian legal ecosystem. As shown by the document types considered in this thesis, my work has primarily concentrated on two contexts: Italian civil judgments and criminal investigations. This activity began in 2021, when I started collaborating with my future Ph.D. supervisor, Professor Matteo Palmonari, on the application of entity extraction methods to investigative data. During this collaboration I contributed to the development of proofs of concept and architectural prototypes for integrating and managing documents in these contexts, which gave me continuous exposure to real use cases and the opportunity to collect feedback from legal professionals.

In the remainder of this chapter we define the requirements for the architectural design (Section 5.1), then we describe our proposal, including the data model with the chosen interchange format (Section 5.2) and the model of the architecture (Section 5.3), compare it with the prototypes previously developed as part of this Ph.D. (Section 5.4), and finally describe the advanced user interfaces (UIs) that can be paired with an implementation of the proposed data integration architecture (DIA) (Section 5.5). The chapter concludes with a summary of the main points (Section 5.6).

The Ph.D. publications related to this chapter are the following: Bellandi et al. [32], Pozzi et al. [271], and Agazzi et al. [4].

5.1 Requirements

As anticipated in Section 1.1, the proposed data integration architecture serves as the bridge between the information extraction and access layers. It ensures that extracted information is stored, integrated, and made accessible in a consistent and verifiable manner.

The architecture is designed to support both structured legal documents (e.g., judgments, legislative acts) and the heterogeneous data produced in investigative contexts. Its design follows a set of guiding principles:

1. **Generalizability** to heterogeneous documents and new use cases, as different investigations may require ad-hoc workflows or specialized processing chains.
2. **Traceability and verifiability**, ensuring that the provenance of any extracted information can always be inspected. For example, a user querying for documents mentioning a person should be able to view the original mention in its textual context, while a question answering component should expose the passages supporting its answers.
3. **Error correction capability**, enabling users to identify and correct extraction errors through a human-in-the-loop (HITL) [185] validation approach, supporting human oversight [106, Art. 14] and reinforcing system quality.
4. **Scalability** to large and continuously growing document collections.
5. **Loose coupling** between components, including external services built on top of the architecture, allowing independent evolution, replacement, or reconfiguration of modules (e.g., switching to an alternative search engine or updating external entity recognition or question answering components) without affecting the rest of the system. Loose coupling also contributes to the architecture availability, as failures or maintenance operations affecting a single component do not compromise the operation of the overall architecture.
6. **Interoperability** with internal and external systems, to support complex workflows in which several extractors are executed in sequence. For this reason, it is necessary that all services *communicate* using a uniform input-output format (UnIOF), so that sequential workflows can be seamlessly modified by adding or removing extractors.

Document–Annotation–Entity Triad Before introducing the functional requirements that support the architectural principles and the domain-specific use cases, we first define the three conceptual pillars that underpin our representation of heterogeneous textual data: *documents*, *annotations*, and *entities*.

A *document* is the abstract unit of textual data in our model. Its granularity is not fixed and may range from entire legal judgments to individual chat messages, depending on the use case. Documents serve as the primary carriers of text to which structured information can be attached.

An *entity*, defined in Section 2.2, represents conceptual objects of interest that can be mentioned across documents. Entities enable the integration of heterogeneous sources into a unified knowledge representation, supporting the generalizability requirement.

An *annotation* establishes a structured link between a document and a portion of its content, optionally associating it with an entity. Annotations serve as the mechanism that enables traceability from structured representations back to their textual evidence (traceability and verifiability), and they generalize beyond entity mentions to any meaningful fragment of text. For example they can represent document sections, or events.

We now introduce the additional functional requirements that support the domain-specific use cases considered in this thesis. Table 5.1 summarizes both the principal architectural requirements and the additional functional requirements, together with the use cases they support. For convenience,

Type	Requirement	Motivation / Supported Use Cases
Principle	Generalizability	To support heterogeneous documents and evolving needs.
Principle	Traceability and Verifiability	Needed by legal professionals to justify results of automatic operations, tracing them to the source data.
Principle	Error Correction Capability	Human-in-the-loop correction is required for improving system quality in sensitive domains.
Principle	Scalability	Necessary for handling vast document collections, as during investigations.
Principle	Loose Coupling	Reduces cascading failures in services.
Principle	Interoperability (UnIOF)	Simplifies integration of new services and composition of pipelines.
Principle	Minimum Necessary Functionality	To restrict the DIA design to only what is needed to support the requirements and all use cases (*UC 1*–*UC 5*).
Functional	Create, read, update and delete (CRUD) APIs	Foundation to create and revise resources required in all use cases (*UC 1*–*UC 5*).
Functional	Multiple Document Collections	Supports generalizability to different granularities (chat vs. messages) useful for investigations.
Functional	Annotation Versioning	Required for traceability and error correction capability.
Functional	Retrieval APIs (sparse, dense, hybrid)	Required for *UC 1* (case retrieval), *UC 3* (investigative retrieval), and *UC 4* (QA).
Functional	Support for a query language (QL)	Required for *UC 5* (statistical analysis and performance monitoring).
Functional	Visualization UI	Needed to verify and trace extracted information (e.g., where an entity is mentioned). Providing basic *UC 2* (document navigation).
Functional	Error Correction UI	Needed for error correction capability.

Table 5.1: Summary of architectural principles and functional requirements of the DIA.

and to remind the reader of the domain-specific use cases defined in Section 1.1, we briefly restate them here:

- case retrieval and precedent search (*UC 1*);
- document navigation and content exploration (*UC 2*);
- investigative retrieval (*UC 3*);
- question answering over documents and collections (*UC 4*);
- statistical analysis and performance monitoring (*UC 5*).

The additional functional requirements are the following:

- ◇ *create, read, update and delete (CRUD) [222] APIs* for creating, reading, updating, deleting documents, entities, and annotations. However, an important exception need to be applied: since the annotations refer and depend on the document text, modifying it should normally be forbidden as it invalidates all the existing annotations.

- ◇ Support for *multiple collections of documents*. While, for example, storing data from different investigations in the same DIA instance may not be advisable due to risks in data protection, this requirement allows handling—and interlinking via URIs—different but related set of documents, or to host multiple "models" of the same data. As a practical example derived from the experience as investigative consultant, investigator may need to visualize entire chats from instant messaging application (IMA) data and also to navigate single messages—e.g., searching for messages sent in a specific time interval. In the first scenario it is preferable to model chats as documents, while in the second to atomically model messages as documents. This requirement enables this kind of flexibility, also allowing to interlink items, e.g., for visualizing the entire chat containing a message. Finally, different document collections should be allowed to refer to different knowledge repositories.

- ◇ Support for *annotation versioning*. The architectural model prototyped in previous work [32] was already able to manage multiple versions of the same document. This functionality was introduced for storing the original document and additional pre-processed copies. In this formalization of the DIA we simplify this functionality to versioning only the annotations—which can be modified by humans through the Error correction UI—in order to allow *error correction capability* and *traceability* at the same time. For fully supporting these principles, we borrow consolidated concepts from version control systems for software development [425]: it should be possible to maintain, and potentially restore, previous versions of all the annotations and any modification should be recorded with an identifier of the user who performed the operation and an explanatory message describing the changes—similarly to a git commit [425]. For more complex needs requiring documents' modifications, users can rely on the support for *multiple collections* and ingest the processed documents in a new collection.

- ◇ *Retrieval APIs*, supporting:
  - sparse information retrieval (IR) for keyword-based retrieval or faceted search (*UC 1*, *UC 3*) [17, 141];
  - dense semantic retrieval for question answering (*UC 4*) [14];

  - hybrid retrieval combining sparse and dense [14].

  In fact, users may benefit from both: sparse retrieval systems allow to exactly search for a certain keyword; while RAG applications generally use dense retrieval [155]; and combining both may provide complementary information [14].

- Support for a *query language (QL)* capable of expressing complex queries to enable advanced statistical analysis (*UC 5*).
- *Visualization UI* that allow users to visualize documents with automatic annotations, to trace and verify the mentioned entities, in support of the requirement for *traceability and verifiability*.
- *Error correction UI* that allow users to access documents and automatic annotations, to correct machine errors in support of the requirement for *error correction capability*.

Moreover, to support users in pursuing their information access needs for the considered use cases, we propose several UIs in Section 5.5, while in Section 5.4 we discuss the differences between our proposal for a data integration architecture (DIA) and the previous works [32, 271, 4] that have contributed to the expertise underlying the formalization presented in this chapter.

However, differently from those works, the present chapter—which addresses *Obj. 2*—focuses strictly on the data integration architecture. Some design principles for peripheral services (e.g., extractors) and UIs are discussed, but they are considered external to the DIA core. Our objective is to define the *minimum necessary* directives to guide the implementation of a data integration architecture for knowledge-intensive domains—and, in particular, for the legal domain—so that it provides the fundamental functionality required to satisfy the architectural principles.

5.2 Data Model and Interchange Format

As anticipated, to support the integration of heterogeneous data sources within a unified representation, we formalize three core concepts: *documents*, *annotations*, and *entities*. These concepts establish the structural basis for storing and integrating heterogeneous data from knowledge-intensive domains. Their design is inspired by the GateNLP [77, 122] project, from which we adopt the notions of documents[1], annotations[2], and annotation sets[3]—that allows to organize annotations in sets—as well as the use of key–value metadata.

GateNLP [77, 122] is a Python framework for natural language processing that provides a flexible representation of documents and annotations, also offering interactive visualization in Jupyter notebooks.

We also adopt GateNLP python objects serialized in JSON format [49] as the *data interchange format*, to achieve interoperability of different services. The integration of dedicated services, indeed, is supported by the designed DIA for the purpose of generalizability, as dedicated processors may be required by specific use case. For example, the input of the *create document* API or the output of systems that produce annotations, such as for entity recognition and entity linking, must be in *GateJSON* format—which means an instance of the GateNLP document object[4] serialized

[1] https://gatenlp.github.io/python-gatenlp/pythondoc/gatenlp/document.html
[2] https://gatenlp.github.io/python-gatenlp/pythondoc/gatenlp/annotation.html
[3] https://gatenlp.github.io/python-gatenlp/pythondoc/gatenlp/annotation_set.html
[4] https://gatenlp.github.io/python-gatenlp/pythondoc/gatenlp/document.html

Listing 5.1: The GateJson format with an illustrative judgment. Metadata are represented in the "features" field.


```
{ "name": "Judgment No.~987/2025",
  "text": "Giovanni Bianchi, residing in Bologna [...] -- plaintiff and Davide Ricci,
      residing in Milan [...] -- defendant FACTS [...] April 3, 2022 [...] EUR 10,000
      [...] December 31, 2022 [...] Article 1813 c.c. [...]",
  "features": {"uri": "https://anndb.example.org/document/Judgment9872025},
  "annotation_sets": {
    "annset_entities": {
      "annotations": [
        { "id":0, "type":"Person", "start":0, "end":16, "features":{"NIL":true, "link":"
            https://newkr.example.org/Giovanni_Bianchi", "uri":"https://anndb.example.org/
            annotation/12342"} },
        { "id":1, "type":"Location", "start":30, "end":37, "features":{"link":"https://en.
            wikipedia.org/wiki/Bologna"}, "uri":"..." },
        { "id":2, "type":"Person", "start":61, "end":73, "features":{"NIL":true, "link":"
            https://newkr.example.org/Davide_Ricci"}, "uri":"..." },
        { "id":3, "type":"Location", "start":87, "end":92, "features":{"link":"https://en.
            wikipedia.org/wiki/Modena"}, "uri":"..." },
        { "id":4, "type":"Date", "start":182, "end":195, "features":{"uri":"..."} },
        { "id":5, "type":"Money", "start":219, "end":229, "features":{"currency":"EUR", "
            uri":"..."} },
        { "id":6, "type":"Date", "start":331, "end":348, "features":{"uri":"..."} },
        { "id":7, "type":"Law", "start":362, "end":379, "features":{"link":"https://www.
            normattiva.it/...", "uri":"..."} }
      ]}}}
```


in JSON. The serialized documents, indeed, also contain the annotation sets and the annotations, as shown in Listing 5.1. This choice also maintain compatibility with GateNLP functions (after deserializing documents into Python objects) which support advanced operations on annotation sets, useful, for instance, to identify overlapping annotations, or annotations contained in another annotation.

As a real example, suppose to have a document, derived from a legal judgment, with two annotation sets, one containing annotations for the sections of the judgment, and the second containing ER annotation for the people involved in the judgment. For identifying plaintiffs and defendants we usually focus on the beginning of the document, in the section "identification of the parties". So given this section, with GateNLP APIs we are able to obtain which people are mentioned in that section.

To clearly define the components of our document–annotation–entity triad, additionally enriched by annotation sets, we describe each element separately and provide a conceptual schema in Figure 5.1.

1. **Document.** A document represents a general unit of data and may vary in granularity depending on the use case—for instance, a legal judgment, a contract, or a single IMA message. Each document is characterized by:

    - a unique URI, ensuring global identifiability;

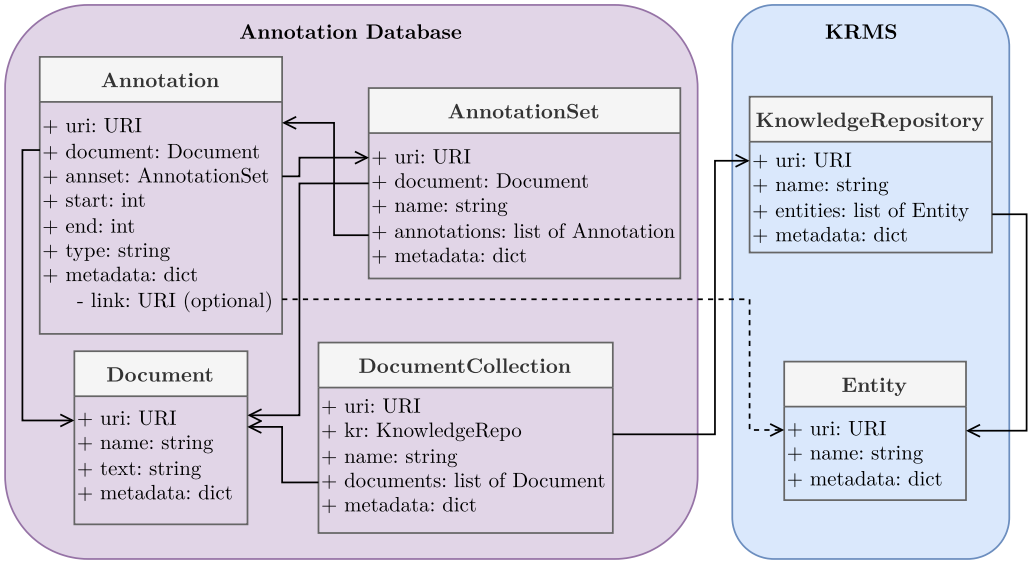


Figure 5.1: Data model for documents, annotations, annotation sets, and entities, additionally including the knowledge repository (KR). KRMS stands for knowledge repository management system.

- a *name*, used as a human-readable label;
- the *text content*;
- optional *metadata*, stored as key–value pairs.

Documents serve as the source layer for annotations and the reference point for traceability.

2. **Annotation.** An annotation connects a portion of a document to a specific concept or entity, thereby linking unstructured text with structured knowledge. An annotation is defined by:

   - a unique URI;
   - the URI of the *source document*, ensuring that the annotation can always be traced back to its origin—even when processed within collections derived from another collection;
   - the URI of the *annotation set* it is part of, for ensuring traceability;
   - the *start* and *end* character offsets in the document text;
   - the *type* of the annotation, e.g., the ER type, or the name of a section;
   - optional *metadata*, as key–value pairs, representing additional properties, e.g., the linked entity URI for EL annotation or the version of the annotation algorithm.

   Annotations allow users to visualize, verify, and trace where entities are mentioned in documents. Beyond entity linking, this concept generalizes to any document fragment, from short spans of text to entire sections, enabling flexible annotation schemas.

3. **Annotation Set.** An annotation set is a collection of annotations organized according to a shared criterion, such as annotation type, purpose, or version. This structure allows for a better organization of the annotations and for the coexistence of multiple annotation layers over the same document (e.g., manual vs. automatic annotations, or distinct semantic layers), as well as for annotation versioning—which can be achieved by creating a modified annotation set, saving the version number, the user who modified it, and a message explaining the changes in the annotation set's metadata. Each annotation set includes:
   - a unique URI;
   - the URI of the *document* to which the annotation set refers;
   - a *name*, describing the set (e.g., "ER-v1" or "EL-v1");
   - a list of contained *annotations*;
   - optional *metadata*, stored as key–value pairs.
4. **Entity.** Each entity, in line with the definition in Section 2.2 is characterized by:
   - a unique URI, ensuring global identifiability;
   - a *name*, used as a human-readable label;
   - optional *metadata*, stored as key–value pairs. For example, metadata may contain the entity type (or types).
5. **Document Collection.** A document collection contains multiple documents and provides a way to organize and manage different documents separately in the same DIA instance. Each document collection is characterized by:
   - a unique URI;
   - the URI of the *knowledge repository* considered by this collection;
   - a *name*, used as a human-readable label;
   - a list of contained *documents*;
   - optional *metadata*, stored as key–value pairs.
6. **Knowledge Repository.** We additionally include the KR in the data model to enable using different principal KRs with different document collections. Indeed, while URIs already allow referring to multiple KRs, including public ones, a principal KR is needed for representing new entities (see *novel entity challenge (Ch. 4)* in Section 1.1) and using only one KR for different document collections may generate conflicts. In Figure 5.1, knowledge repository management system (KRMS) refers to a system that manages multiple KRs, similarly to database management systems (DBMS), which can hold multiple databases. A KR is characterized by:
   - a unique URI;
   - a *name*, used as a human-readable label;
   - a list of contained *entities*;
   - optional *metadata*, stored as key–value pairs.

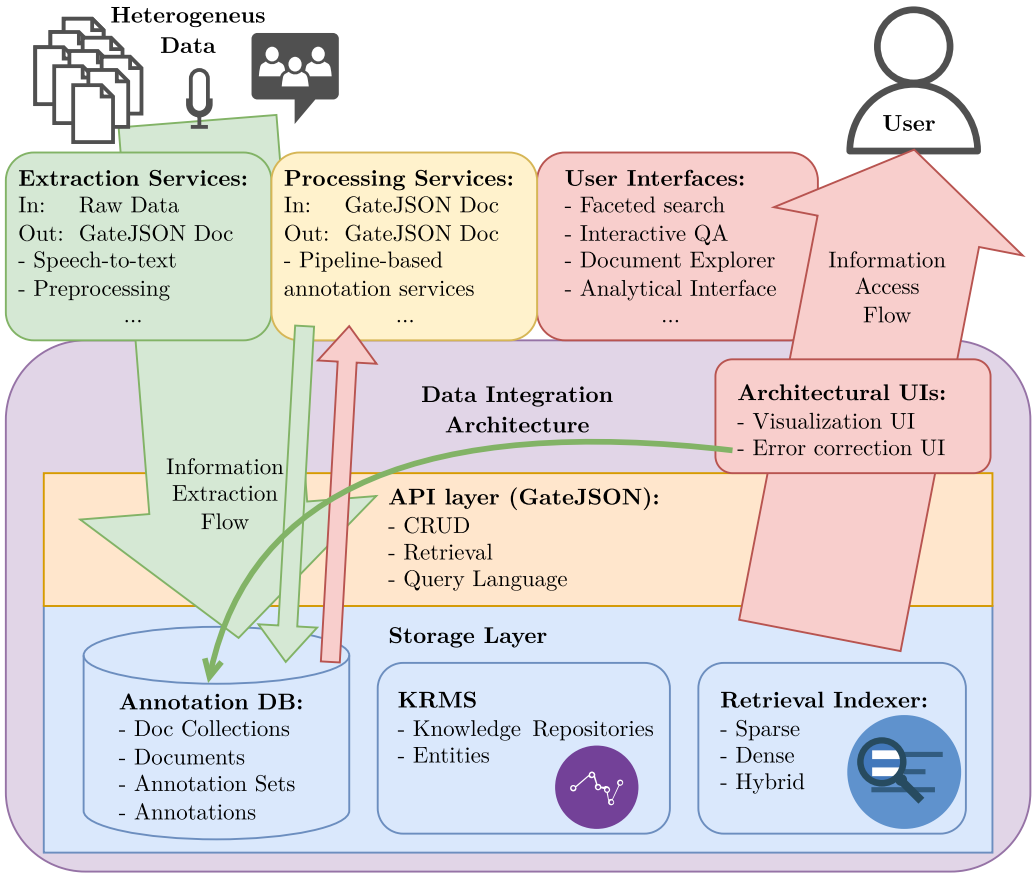


Figure 5.2: High-level architectural model of the data integration architecture.

While GateNLP uses internal numeric identifiers for its objects, we extend it by introducing globally unique URIs, which simplify locating resources, enabling inter-document linking, and the use with distributed storage. Furthermore, URIs allow for the seamless integration of public KRs and KBs, allowing to link annotations to entities from Wikipedia, Wikidata, or domain specific KRs.

In Figure 5.1, entities are placed in the knowledge repository, since they are stored and managed there, as described in the next section.

5.3 Architectural Model

An overview of the data integration architecture is depicted in Figure 5.2. The figure shows a clear division between what is part of the DIA and what is outside it. Inside we find the *storage layer*,

whose objective is to store information regarding documents, annotations, and entities, as well as to organize this information for efficient access via indexing. This layer is composed of:

- *Annotation database (AD)*, which stores documents, annotations, and annotation sets as defined in the data model, supporting multiple document collections.
- *Knowledge repository management system (KRMS)*, which stores local entity knowledge, eventually in multiple knowledge repositories (see Section 2.2 for the definitions of entity and knowledge repository). Note that, while in the DIA design this is not enforced, the KR components can be knowledge bases holding facts about entities in the form of RDF triples.
- *Retrieval indexer (RI)*, which holds indexes for efficient retrieval and supports sparse, dense, and hybrid retrieval paradigms.

On top of the storage layer sits the API layer, which exposes interfaces to interact with the persisted data. This is the component that must provide the mandatory APIs defined by the requirements:

- ◇ *CRUD APIs* for all the resources in the data model: documents, annotations, annotation sets, and document collections, stored in the AD, entities and knowledge repositories, stored in the KRMS.
- ◇ *Sparse, dense, and hybrid retrieval APIs* using the indexes from the RI for efficient retrieval.
- ◇ *query language APIs* supporting aggregation and filtering operations required to compute statistics, derive distributions, and answer structured analytical queries over documents, annotations, and entities. These interfaces must support at least the standard family of operations found in declarative query languages—such as grouping and aggregation (e.g., counts, distinct counts, maxima/minima, or averages)—as documented in mainstream DBMS (e.g., PostgreSQL [351]). Such capabilities are essential to support use case *UC 5*, i.e., statistical analysis and monitoring. This requirement is reflected in the implementation choice of the annotation DB, which has to support such operations.

These APIs must operate over the network to support distributed deployments, enabling parallel processing and horizontal scalability. Furthermore, they must support the GateJSON format for both input and output of annotated documents, ensuring interoperability across heterogeneous components.

Besides the storage and API layers, the DIA features the two fundamental UIs for fulfilling traceability-verifiability and error-correction capability. The *Visualization UI* must provide the following functionalities to users:

- visualize annotated documents, selecting which annotation sets to visualize;
- access the properties (defined in the data model) of documents, annotation sets, annotations, and entities, while visualizing the annotated document—e.g., by clicking on an annotation to inspect annotation properties—or given the URI of the resource;
- given an entity, trace which annotations mentioned it and visualize them in the document.

On the other hand, the *Error correction UI* provides functionalities to correct annotation errors from automatic tools and, in general, provides delete operations for removing automatically created resources whose creation was an error—such as duplicated entities in the KR. While additional functionalities, such as manual ingesting of documents, may be useful, we only include the minimum necessary design principles for allowing *error correction capability*.

- For documents, it only provides the delete operation, which consequently deletes all the associated annotation sets and annotations.
- Also, for document collections it provides the delete operation, which consequently deletes all documents and annotations.
- Similarly, for knowledge repositories, it allows for deletion, which propagates to all contained entities.
- For annotations, it provides the following operations:
  - create;
  - delete;
  - move—same as modify start and end;
  - modify properties (type and metadata)—for example to change the linked entity;
  - versioning: each modification—with annexed the user ID and a message explaining the changes—must be persisted in a *history* at the annotation set level. This centralizes the annotation versioning management to the annotation set, allowing to visualize the history of the entire annotation set and to perform rollback operations.
- For annotation sets, it allows to:
  - delete a set, together with all the annotations;
  - view history, including user IDs and message for each modification;
  - rollback to a specific version, undoing all the modifications performed afterwards.
- For entities, it allows to:
  - create a new entity to manually assign it to annotations;
  - merge two entities and consequently assign all the linked annotations to the new entity;
  - delete an entity and optionally chose another entity to assign to the linked annotations and get the list of affected annotations for manually revising them;
  - modify name and metadata.

We conclude the core architectural description by defining three information flows that the DIA must support. These flows cover all interactions with external services—such as preprocessing components, text annotators, or UIs—and assume interoperability through the GateJSON format. We also recall the design guideline of ensuring loose coupling among services to reduce cascading failures.

1. *Information extraction flow.* External services process raw data, serialize it in GateJSON, and inject it into the DIA via APIs. This corresponds to the green arrows in Figure 5.2, incoming from outside to the core of the DIA.

2. *Information access flow.* UIs retrieve information stored in the DIA via read or retrieval APIs (using GateJSON) and present it to users. This corresponds to the outgoing red arrow in Figure 5.2.

3. *Circular flow.* Some services both read from and write to the DIA, such as sequential processors that enrich documents (e.g. by adding annotation sets). The *Error-correction UI* is similar: it reads from the DIA and persists human-made revisions back into it.

These three flows enable external services to ingest, access, and iteratively refine information, allowing the DIA to generalize across use cases. Moreover, executing processing steps on top of previously generated results, combined with network-based APIs, enables complex workflows and distributed execution. The ability to operate on separate annotation sets minimizes conflicts among concurrent processes, while loose coupling supports scalability when combined with an orchestrator. The systematic use of URIs ensures that documents and annotations can be retrieved unambiguously via the APIs.

Before proceeding with the architectural details, Table 5.2 summarizes how the proposed design and data model support each requirement.

Table 5.2: Support of each requirement in the proposed DIA design and data model.

Requirement	**Support in the DIA**
Generalizability	Document–annotation–entity triad; multiple collections with URIs for interlinking; easy integration of new services via APIs and interoperability.
Traceability and Verifiability	Annotations store source text spans and URIs of the originating document and set; annotation versioning records all corrections, preserving trace of changes. Visualization UI to browse annotated documents and trace entity–annotation links.
Error Correction Capability	Error correction UI, coupled with annotation versioning with history and rollback.
Scalability	Network APIs enables distributed execution, loose coupling services and independent annotation sets simplify parallelization, circular flows simplify multiservice orchestration.
Loose Coupling	Supported as a design principle for services.
Interoperability (UnIOF)	GateJSON as a unified interchange format across all services.
Minimum Necessary Functionality	Core restricted to storage + APIs + minimal UIs only.
CRUD APIs	Provided by the API layer over all the resources of the data model.
Multiple Document Collections	Supported in the data model with document collections and multiple knowledge repositories.
Annotation Versioning	Part of the error correction UI; history handled at annotation-set level with rollback.
Retrieval APIs	Sparse, dense, hybrid retrieval via the RI.
Support for a QL	Necessary requirements for choosing the annotation DB implementation.
Visualization UI	Designed to display annotated documents, selecting annotation sets, inspecting properties, tracing entity-annotation links, supporting traceability and basic *UC 2* (navigation).
Error Correction UI	Designed to revise or delete annotations and entities, with versioning to preserve history and support error correction capability.

5.4 Previous Prototypes

The architectural model introduced in this chapter derives from the experience acquired in previous projects. In this section we compare it with previous prototypes developed as part of my Ph.D., namely Bellandi et al. [32], Pozzi et al. [271], and Agazzi et al. [4]. Among these, only Bellandi et al. [32] explicitly treated architecture as a research goal, but all these works implemented parts of the principles and functional requirements that are now formalized in this chapter.

Table 5.3 compares the requirements defined here with what was actually supported by earlier prototypes. The comparison shows that the systems did not converge to a stable architecture, but they provided the practical experience and feedback that led to the general model presented in this chapter.

Table 5.3: Support of DIA requirements across prior works (✓ full, △ almost, ~ partial, – none).

Req.	Aspect	Ref. [32]	Ref. [271]	Ref. [4]
Generalizability	doc–ann–ent	✓	✓	✓
	URI	~ not universal	~ not universal	~ not universal
	multi collect.	△ not all UIs supp.	–	–
	service integration	✓ RAE	✓ in principle	✓ in principle
	interop. UnIOF	✓ GateJSON	✓ GateJSON	✓ GateJSON
	loose coupl.	✓	– not addressed	– not addressed
Traceability	trace. annotations	✓ w/ GateNLP	✓ w/ GateNLP	✓ w/ GateNLP
	visual. UI	△ no entity page	✓	✓
Error Correction	version/rollback	–	–	–
	err corr. UI	~ no ent. no hist.	~ no ent. no hist.	~ no ent. no hist.
	ann. versioning	~ multi doc vers.	–	–
Scalability	net API	✓	✓	✓
	annset parallel.	✓	✓	✓
	other	implements service orchestrator	–	–
Minimum-necessary		– orchestrator in architecture	–	–
CRUD APIs		✓	✓	✓
Retrieval Modes		~ sparse only	~ sparse only	✓
Query Language		✓ SQL,MQL [237]	✓ MQL,Cypher[78]	✓ MQL [237]

In Bellandi et al. [32] we built the first prototype that integrated storage, extraction, and information access UIs for civil judgments. The study explicitly considered architectural aspects, prototyped search and visualization over annotated judgments, adopted the document–annotation–entity representation, and showed generalizability and interoperability with ad-hoc services—i.e., rule-based extractors (RAE)—for extracting law articles and postal addresses.

In Pozzi et al. [271] we applied a similar design to investigative IMA data. The evaluation made clear that it was necessary to generalize the notion of document (chat vs. message) and that

the existing implementation lacked support for multiple document collections—which would have been needed to represent both chats and messages within the same instance. In that prototype we instead used a knowledge graph (implemented with neo4j [246], a graph database) to represent messages. The graph-based visualization provided in neo4j [246] UI received positive feedback from investigators.

Although, in that case, only a limited set of relations was considered, using a knowledge graph (KG) enables storing entity facts as triples and increases the modeling capacity, as shown for instance by the *participant-in* relation between `Person` and `Chat`. Such facts can also support more advanced information access methods such as ReFactX, discussed in Chapter 6. Nevertheless, a KG was not prescribed in the final architecture, since it is not strictly required under the minimum-necessity principle, and the knowledge repository (KR) is a more general abstraction than a KG.

Finally, Agazzi et al. [4] focused on improving user interfaces for information access, adding support for all retrieval modes. This work did not address other requirements such as multiple document collections, annotation versioning, or full error correction (e.g., entity or document deletion was not available from the UIs).

In summary, the prototypes developed during the Ph.D. anticipated many components of the data integration architecture and also revealed the missing parts that motivated its explicit formalization.

5.5 User Interfaces and Supported Use Cases

In this section we describe advanced user interfaces (UIs) for information access, developed as part of this Ph.D., which can be paired with an implementation of the proposed DIA to support the use cases considered in this thesis (introduced in Section 1.1), and we indicate which of the use cases they address.

An important contribution in this line is Document Assistant for Validation and Exploration (DAVE), which has been recently published [4] and released with Apache-2.0 license[5]. It consists of an entity-centric framework for assisting users in analyzing documents from knowledge-intensive domains and its main UIs functionalities follow below. Furthermore, these are illustrated at the end of this section, in Figures 5.3 to 5.7, using court documents from proceedings concerning the 1995 Oklahoma City bombing.[6]

- *Faceted search* [17, 141]: users can retrieve documents (in support of the retrieval use cases, *UC 1* and *UC 3*) by entering keywords, as shown in Figure 5.3. Additionally, they can refine their search results by applying multiple filters corresponding to the entities in the documents or their types. Users can filter the corpus based on specific entities, such as individuals or locations. For example, a user interested in bombing cases in Austin, Texas might search for the keyword "bombing" and later narrow the focus using the filtering panel and selecting Austin, Texas from the list of identified entities. In this case, the resulting document will match the keyword "bombing" and contain a link to the entity Austin, Texas.

- *Document explorer*: users can visualize documents enriched with semantic text annotation by coloring the entity mentions [347, 131, 114] according to their types (Figure 5.4). Each annotation is clickable allowing to inspect its details (Figure 5.5). Furthermore, this UI

[5] https://github.com/unimib-datAI/DAVE
[6] https://en.wikipedia.org/wiki/Oklahoma_City_bombing

enables entity-based document exploration by providing, in a panel on the left, the list of mentioned entities grouped by type. Users can expand a type, search for specific entity, and visualize the list of all its mentions. By clicking a mention, the document explorer focuses it in the document allowing the user to see it in its entire context. This interface supports the navigation and exploration use case (*UC 2*) and implements *traceability and verifiability* to a considerable extent, although the entity view is per-document rather than global.

- *Conversational question answering* supports natural language queries (use case *UC 4*), with the retrieval augmented generation (RAG) paradigm, and allows users to ask questions about entities and factual information across a single document, a selection of documents (previously filtered with faceted search), or all the documents. Besides giving an answer in natural language, it provides the text passages used by the LLM for generating the answer—preserving traceability. It is illustrated in Figure 5.6
- *Error correction and refinement*: users can refine annotations by deleting them from the document explorer, or modify the linked entity. Additionally, entity clusters can be refined with drag-and-drop by selecting two clusters from a dedicated UI and dragging the mentions from one to another, as shown in Figure 5.7. These capabilities support error correction.

Overall, DAVE provides a unified environment that integrates entity-based document management, faceted search, conversational question answering (QA) with RAG, and human-in-the-loop refinement in a single framework. Rather than proposing new algorithms for extraction or retrieval, the system operationalizes known components in a way that is directly usable by professionals, while preserving traceability, verifiability, and error correction capability.

Finally, the graph-based visualization for instant messaging application in investigative contexts [271] is worth a mention: it represents communication networks derived from chats and, thanks to the underlying graph database, it supports investigative queries that cannot be expressed through keyword search alone—related to use case *UC 3*. This UI, illustrated in Figure 5.8, was reported as useful in practitioner feedback, and its integration with the other functionalities in DAVE is a direction for future work.

Table 5.4: User interfaces and supported use cases.

UI type	**Supported UC / principles**
DAVE Document explorer	*UC 2* (navigation); error correction (partial)
DAVE Faceted search	*UC 1* (case retrieval); *UC 3* (investigative retrieval)
DAVE Conversational QA	*UC 4* (QA over documents/collections)
DAVE Cluster refinement	error correction (partial, entity-level refinement)
Graph-based exploration	*UC 3* (investigative retrieval)

Summarizing, Table 5.4 shows how the UIs presented in this section cover the target use cases of the thesis. While a substantial subset of the intended functionality is already supported, the coverage is not complete, highlighting gaps to fill in a full implementation of the DIA. In particular:

1. No deployed interface exists for the computation of advanced statistics (*UC 5*).

2. Partial error correction: some of the designed correction functionalities are not yet available; for example, entity or document deletion cannot be performed from the UIs.

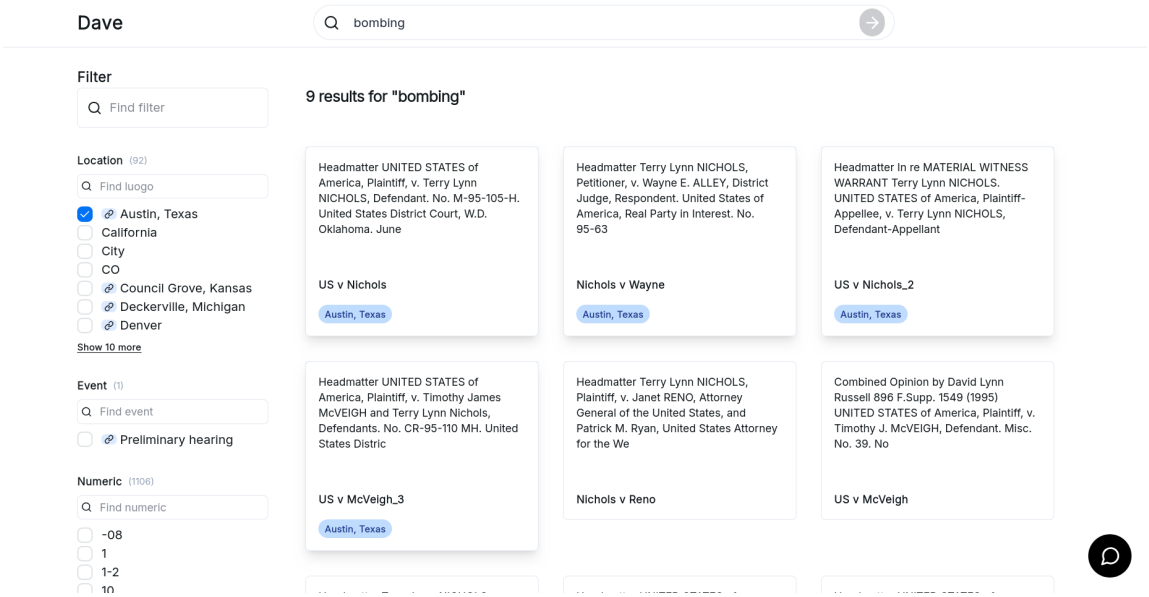


Figure 5.3: DAVE faceted search interface for entity-based filtering and investigative retrieval.

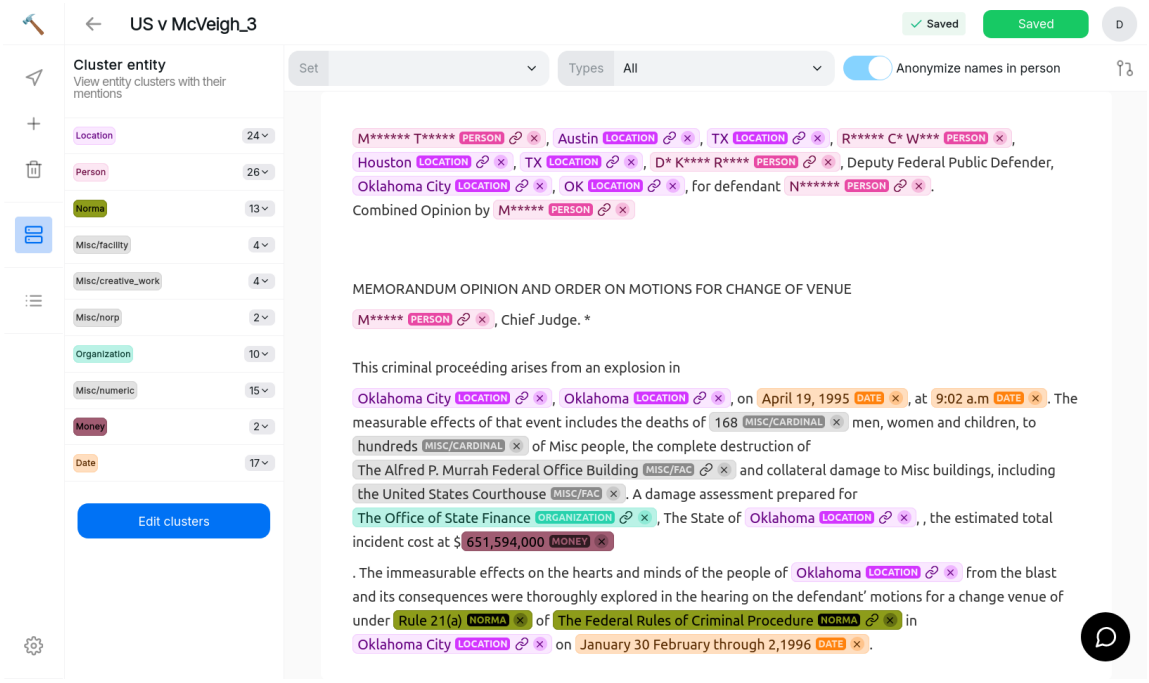


Figure 5.4: DAVE document explorer showing entity types and navigation.

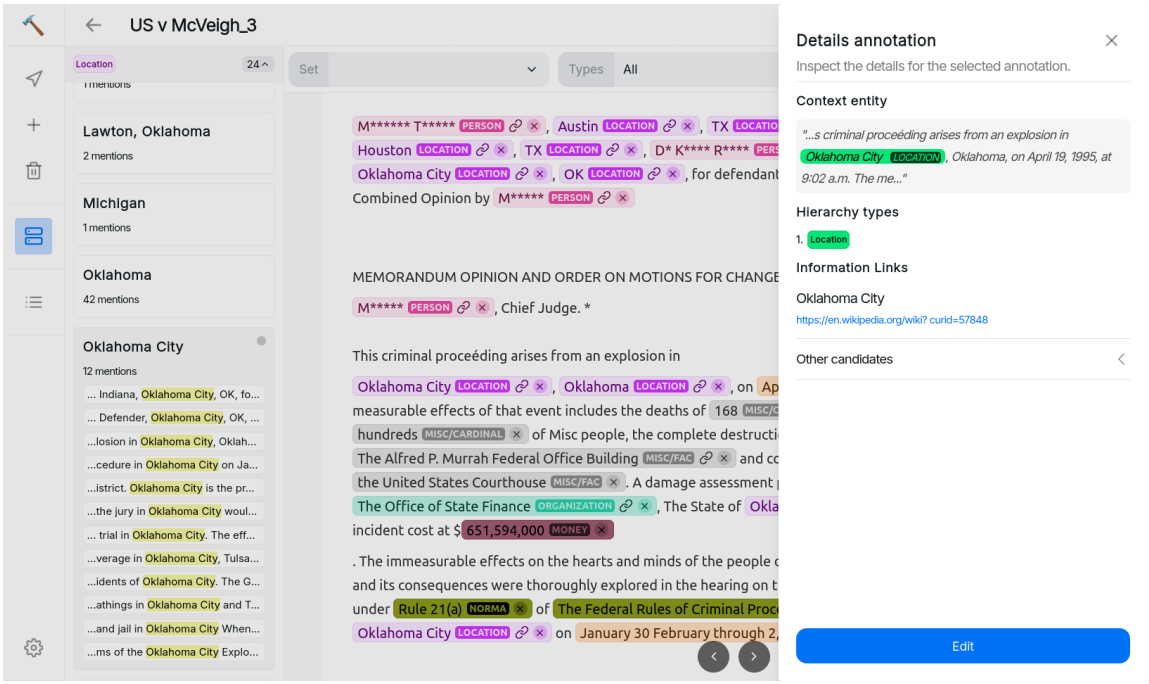


Figure 5.5: DAVE document explorer with instance-level navigation by entities.

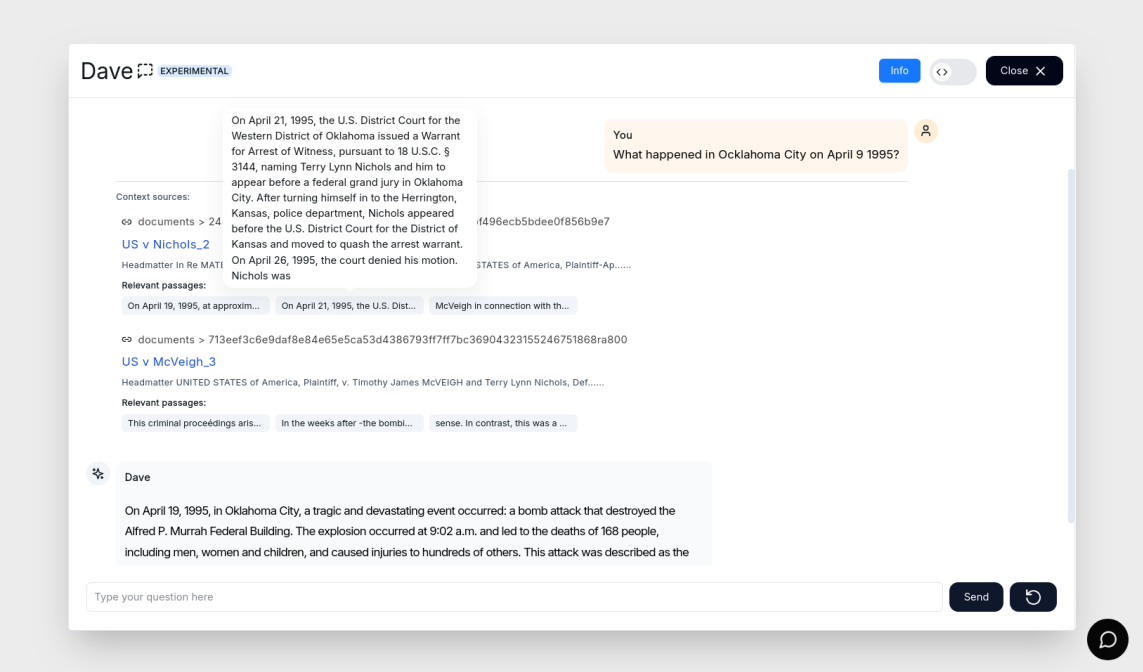


Figure 5.6: DAVE RAG-based conversational question answering interface with grounded answer inspection.

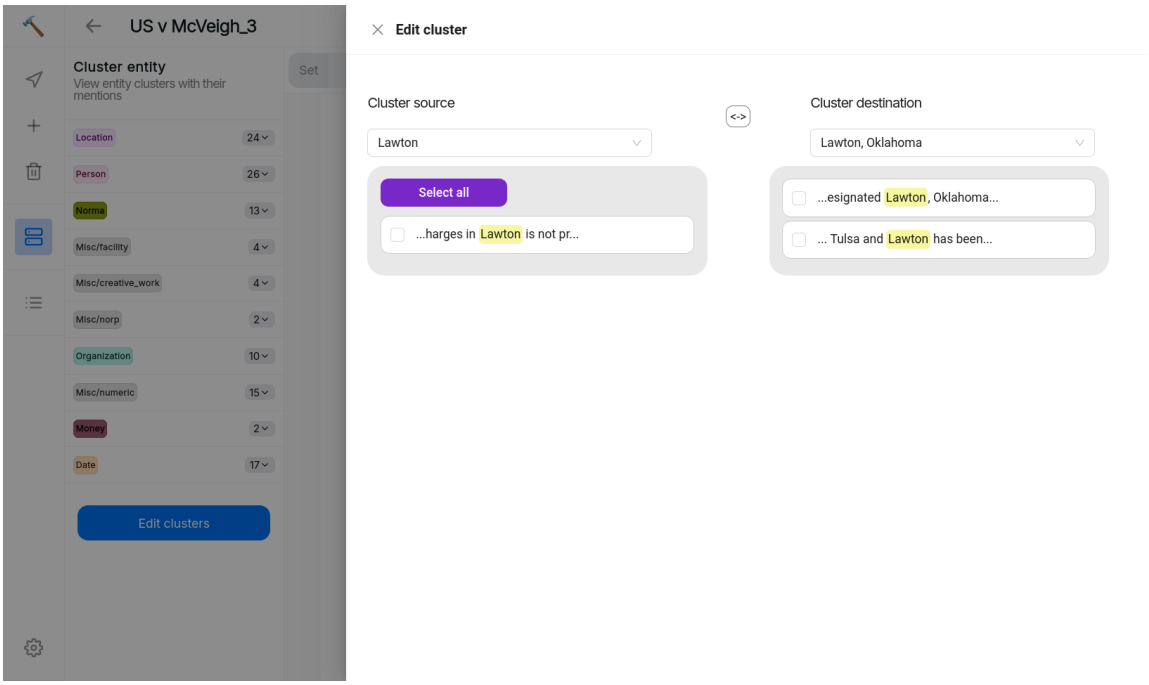


Figure 5.7: DAVE interface for refinement of entity clusters through merge and split operations.

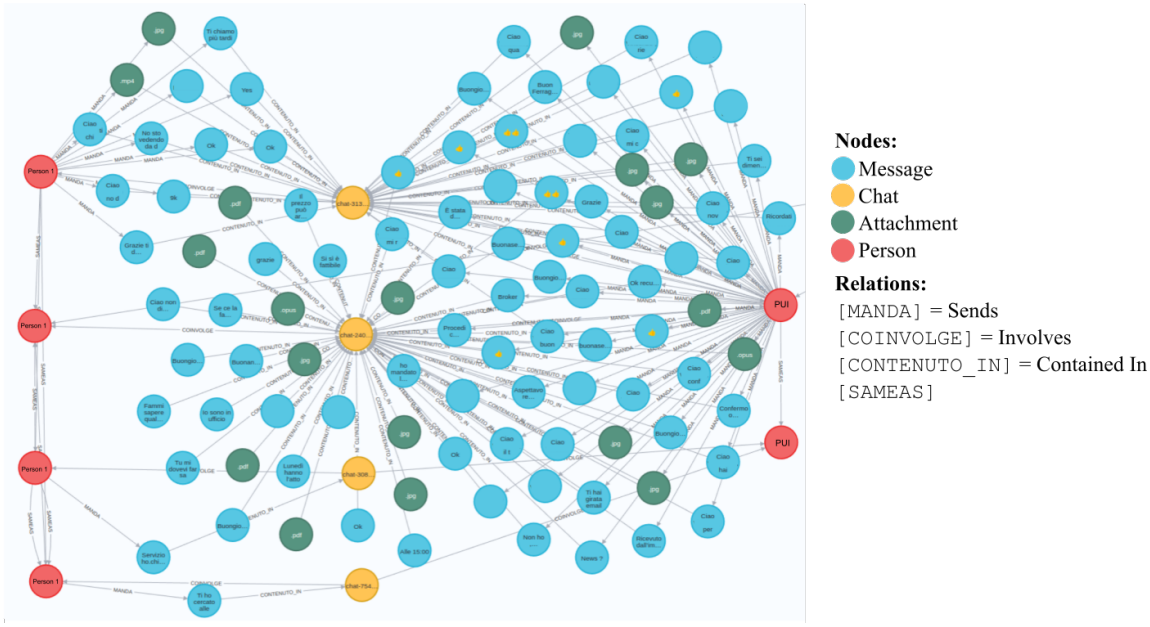


Figure 5.8: Graph-based exploration of chat-derived communication networks supporting complex instances of *UC 3* (investigative retrieval).

5.6 Conclusion

This chapter addressed *Obj. 2* by consolidating the design of an entity-centric data integration architecture for integrating heterogeneous data in knowledge-intensive domains, with a focus on the legal domain. The design was motivated by concrete use cases and by practical experience gained from previous prototypes developed in collaboration with legal professionals. At the same time, the proposal was formulated at a level of generality that can make it applicable beyond the specific domains considered in this thesis.

After recalling the architectural requirements, we presented the proposed data model and overall architecture. We compared the architecture with earlier prototypes to clarify how recurrent limitations informed the present design.

Finally, we proposed advanced UIs that can be paired with an implementation of the architecture to support retrieval uses cases (*UC 1* and *UC 3*), document navigation and exploration (*UC 2*), and question answering (*UC 4*), covering the most of the use cases considered in this thesis.

Chapter 6

Efficient Question Answering over External Knowledge

As described in Sections 1.1 and 2.3, large language models (LLMs) have demonstrated a variety of capabilities, ranging from natural language understanding and generation to reasoning, especially when enhanced with techniques like chain-of-thoughts (CoT) prompting [373]. However, they are prone to hallucinations [226] and their internal knowledge remains limited to their training data (see Section 2.3.2). This complicates their application to knowledge-intensive tasks such as question answering (QA), especially when these tasks necessitate accessing information beyond the parametric knowledge of LLMs—for example, recent data or domain-specific data that is not publicly available.

Solutions to augment LLMs with external knowledge exist, as reviewed in Section 3.3, where we categorized between *input-based* and *latent-interaction* knowledge injection techniques. However, the former rely on external retrievers or pipelines that increase latency, system complexity, and suffer from error propagation. In the latter case, instead, models acquire external knowledge more internally, avoiding the need for external retrievers, but they require architectural modifications complicating the use of recent LLMs.

Question answering plays a central role in knowledge-intensive domains and is therefore highly relevant to this thesis. Moreover, it appears in the use cases considered in Section 1.1 as *UC 4*. However, regulations governing private data may prohibit the use of powerful API-based LLMs, thereby requiring practitioners to rely on locally deployed models subject to hardware constraints. For this reason, objective *Obj. 3* focuses on the development of an efficient and scalable QA system that additionally enables users to trace and verify the produced answers against the original documents or underlying knowledge sources.

In line with objective *Obj. 3*, this chapter introduces Reliable Fact eXtractor (ReFactX), a QA approach that integrates external knowledge without relying on auxiliary models or external retrievers. Instead, it leverages constrained generation supported by a pre-built prefix-tree index, specifically designed to facilitate and accelerate access to external facts. Consequently, ReFactX addresses use case *UC 4*, which concerns question answering in knowledge-intensive domains, while explicitly accounting for the associated challenges, in particular traceability and verifiability (*Ch. 3*), as well as scalability (*Ch. 1*).

A few recent approaches have applied *constrained generation* to QA, achieving promising results [201, 215], however, they did not focus on the scalability to large knowledge bases. This technique restricts the model's output space during decoding to sequences that satisfy predefined

structural, syntactic, or semantic constraints. In our case, we use it for ensuring the LLM generates a fact that really exist in the KG, with the aim of preventing hallucinations.

At inference time, with ReFactX the LLM is instructed via in-context learning (ICL) [96] to invoke the *Fact* command when external facts are required. Once the command is recognized, constrained generation is activated. At this point, the model produces tokens only along valid paths in the prefix tree, guaranteeing that the output matches a fact from the knowledge base, which is derived from Wikidata [377]. Once an entire KB fact is generated, the decoding mechanism reverts back to normal. It is important to note that every generated token is selected based on LLM's probability estimates. Constrained generation only narrows the vocabulary to the tokens that form an existing fact in the KB, but always leaves the final choice to the LLM.

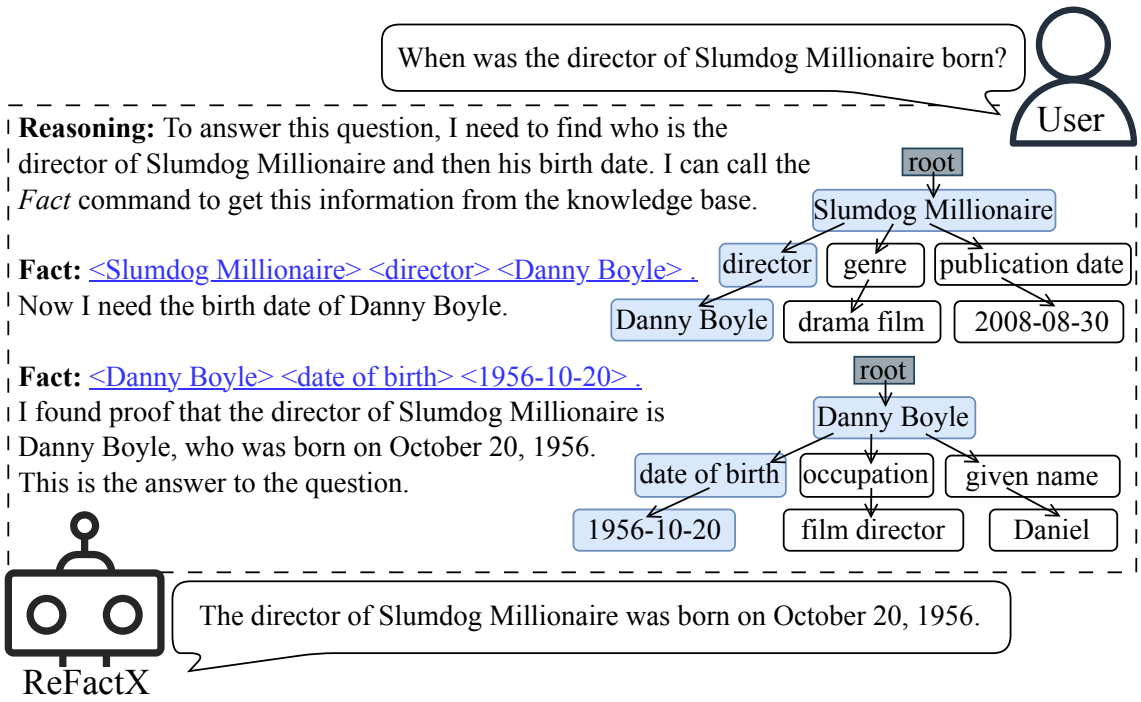


Figure 6.1: ReFactX answering an open-domain question. The LLM sketches a plan, makes two *Fact* calls, and—with constrained generation (blue underline)—inserts valid facts from a Wikidata-based prefix tree before giving the final answer.

Motivating Example. The approach is illustrated in Figure 6.1, by depicting an example with the question *"When was the director of Slumdog Millionaire born?"*. The model is instructed to first reason on how to reach the correct answer, and then it starts to acquire facts. Once the *Fact* command is called, constrained generation is enabled (underlined in blue) and the model is guided to generate an existing fact. This way the model is able to first find that *"Danny Boyle"* is the director of *"Slumdog Millionaire"*, then to find his birthdate, and finally to answer the question correctly.

Furthermore, the facts generated can be highlighted, as in Figure 6.1, so that users are aware that those facts exist in the KB, supporting traceability and verifiability (*Ch. 3*).

Although not enforced, the data integration architecture design proposed in Chapter 5 supports the use of knowledge bases that contain facts, which can be required by advanced systems such as ReFactX.

The main advantages of ReFactX can be summarized as follows:

- Efficiency and scalability: by leveraging a disk-backed prefix tree, ReFactX scales to KBs containing up to 800 million facts, while introducing only a ~1% increase in latency, thereby addressing the scalability challenge of knowledge-intensive domains (*Ch. 1*).
- Traceability and verifiability: by construction, every fact generated through constrained generation is guaranteed to exist in the underlying KB, enabling users to trace and verify whether the provided answers are supported by external knowledge. These properties directly address the traceability and verifiability challenges in knowledge-intensive domains (*Ch. 3*).

Section 6.1 follows by describing the constrained decoding mechanism. Then, to empirically validate ReFactX scalability, we consider Wikidata as a large-scale KB, from which we obtain 800 million facts, and use standard knowledge-graph question answering (KGQA) benchmarks for evaluation. The application of ReFactX to unstructured text from the legal domain, which requires extracting facts from text, is left to future work, but we consider a dataset from the financial domain (a knowledge-intensive domain) to assess the adaptability of ReFactX to domain-specific knowledge. Section 6.2 details how we represent Wikidata facts in a prefix-tree index suitable for constrained generation. Section 6.3 describes how ReFactX is integrated into a QA workflow. Then, Section 6.4 describes the experimental setup, while results are presented in Section 6.5 and discussed in Section 6.6. Finally, Section 6.7 concludes the chapter.

This chapter is related to the publication Pozzi et al. [273] and the corresponding implementation is released under the Apache 2.0 license on GitHub[1].

6.1 Constrained Fact-Generation

The mechanism of constrained generation alters the autoregressive next-token generation process of causal language modeling (CLM). As formulated in Section 2.3, normally an LLM, parameterized by θ, given an initial sequence of tokens $X = [x_0, x_1, \ldots, x_{|X|}]$ and a vocabulary $V = \{v_0, v_1, \ldots, v_{|V|}\}$, estimates the probability distribution of x_i over the entire V:

$$P_\theta(x_i = v \mid x_{<i}) \qquad \forall v \in V. \tag{6.1}$$

The next-token x_i can be chosen according to different sampling strategy; in greedy decoding the most probable token is chosen as follows:

$$x_i = \text{argmax}_{v \in V} P_\theta(x_i = v \mid x_{<i}) \tag{6.2}$$

Intuitively, to constrain this process for generating only existing facts, we need to restrict V to only allow tokens that can form a fact from the KB. We define $V_X^A \subseteq V$, which contains all the tokens that can lead to an existing fact if added to the sequence X. Considering the example in Figure 6.2, at step x_i, when $x_{<i}$ = "<Danny Boyle> <", $V_{x_{<i}}^A$ = {"date", "given"}.

[1] https://github.com/rpo19/ReFactX

Next, we define the *next tokens* function $\mathrm{NT}_{\mathrm{kb}}$ that, given a sequence $x_{<i}$, obtains $V^A_{x_{<i}}$ for the considered KB. Consequently, the token selection formula from Equation (6.2) is updated to:

$$x_i = \mathrm{argmax}_{v \in V^A_{x_{<i}}} P_\theta(x_i = v \mid x_{<i}), \qquad \text{where } V^A_{x_{<i}} = \mathrm{NT}_{\mathrm{kb}}(x_{<i}) = \{v_0, v_1, \ldots\}. \tag{6.3}$$

However, the same result can be achieved by altering the probability distribution, setting the probability of all forbidden tokens to zero—without modifying the vocabulary:

$$P^c_\theta(x_i \mid x_{<i}) = \begin{cases} P_\theta(x_i = v \mid x_{<i}) & \forall v \in V^A_{x_{<i}}, \\ 0 & \text{otherwise.} \end{cases} \tag{6.4}$$

By using P^c_θ, instead of P_θ, we achieve constrained generation, relying on the assumption that the sampling strategy (Equation (6.2)) would never choose any $t_{k+1} \mid P^c(t_{k+1}) = 0$. In practice, since at implementation level the models use log-probabilities, we directly the log-probabilities of forbidden tokens to $-\infty$[2]:

$$\log P^c_\theta(x_i \mid x_{<i}) = \begin{cases} \log P_\theta(x_i = v \mid x_{<i}) & \forall v \in V^A_{x_{<i}}, \\ -\infty & \text{otherwise.} \end{cases} \tag{6.5}$$

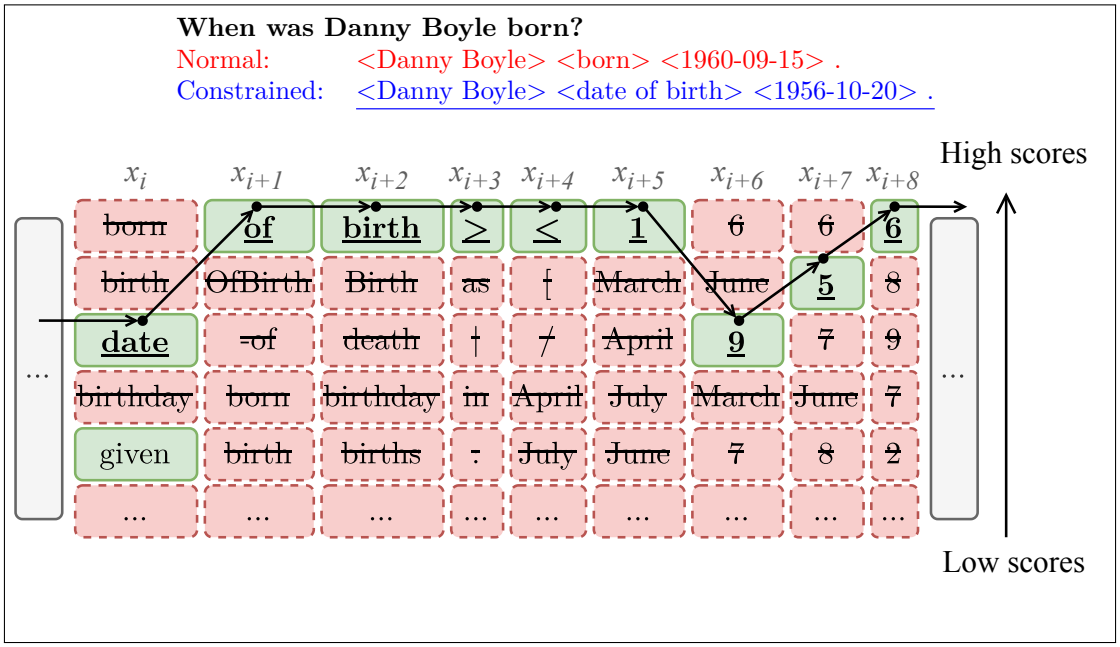


Figure 6.2: Constrained decoding steers the LLM toward the correct fact. At each step, the constrained decoding mechanism chooses the highest-probability token that still completes a Wikidata fact, avoiding invalid branches and yielding *"<Danny Boyle> <date of birth> <1956-10-20> ."*.

Figure 6.2 illustrates the application of constrained generation. It shows two facts produced by the same LLM in response to the question *"When was Danny Boyle born?"*—first using normal

[2]Hugging Face Transformers LogitsProcessor is used to alter the log-probabilities.

(unconstrained) generation, highlighted in red, and then using constrained generation, underlined in blue. While, the fact generated in normal mode is not correct, constrained generation is able to guide the LLM to the correct fact. The lower part of the figure details the mechanism of constrained generation. For each decoding step, allowed tokens are displayed in green boxes, whereas forbidden tokens are displayed in red boxes with dashed borders and a strike-through. Tokens are arranged in ascending order according to $P_\theta(x_k = v)$, from bottom to top.

Starting from the sequence $x_{<i}$ = "<Danny Boyle> <", normal generation would generate

$$x_i = \text{argmax}_{v \in V} P_\theta(x_i = v \mid x_{<i}) = \text{“born”}, \tag{6.6}$$

leading to an incorrect fact, while with constrained generation

$$x_i = \text{argmax}_{v \in V} P^c_\theta(x_i = v \mid x_{<i}) = \text{“date”}. \tag{6.7}$$

This happens because in the KB there is no fact starting with *"<Danny Boyle> <born"*.

Subsequently, when generating x_{i+6}, the constrained generation mechanism guides the model to select "9" that leads to the correct birth year of Danny Boyle *"<Danny Boyle> <born> <1956"*, avoiding the model to generate *"<Danny Boyle> <born> <16"* or *"<Danny Boyle> <born> <196"*, both leading to incorrect information. This mechanism is additionally improved by beam search [154] which allows the model to explore multiple paths in the prefix tree in parallel.

6.2 Scaling to 800 Million Facts from Wikidata

To demonstrate the scalability of ReFactX (*Ch. 1*), we use Wikidata [377] as a large-scale KB. While this study focuses on structured knowledge, applying ReFactX in the legal domains remains an avenue for future work, as it would likely require to extract facts from text.

From the Wikidata truthy dump—which contains the highest-confidence statements while excluding qualifiers[3]—we extract approximately 800 million triples[4]. To filter out uninformative facts only triples whose subject and relation have Wikidata identifiers are retained. For the object, we allow entities with Wikidata identifiers, English literals, numbers, dates, and literals with no language.

Then, for each entity, we obtain a meaningful textual label—because Wikidata IDs lack meaning for LLMs—corresponding to the Wikipedia title if the entity is described in English Wikipedia[5]. Otherwise, since the Wikidata label alone is not unique, the entity description is additionally considered, using the following template.

```
{entity label} ({entity description})
```

For example, the entity Jane Hajduk[6], which is not present in English Wikipedia, is labeled as "Jane Hajduk (American actress)". In this case, the label comes from `rdfs:label` [367] and the description from `schema:description` [314].

With this process more than 800 million facts like the ones underlined in Figure 6.1 were obtained.

[3] https://www.wikidata.org/wiki/Wikidata:Database_download/en#RDF_dumps
[4] The dump from 11 December 2024 was used.
[5] https://en.wikipedia.org/
[6] https://www.wikidata.org/wiki/Q3734827

Additionally, while creating the tree, we calculate the number of reachable leaves from each node to avoid that LLMs generate the same fact repeatedly, a behavior witnessed during preliminary experiments.

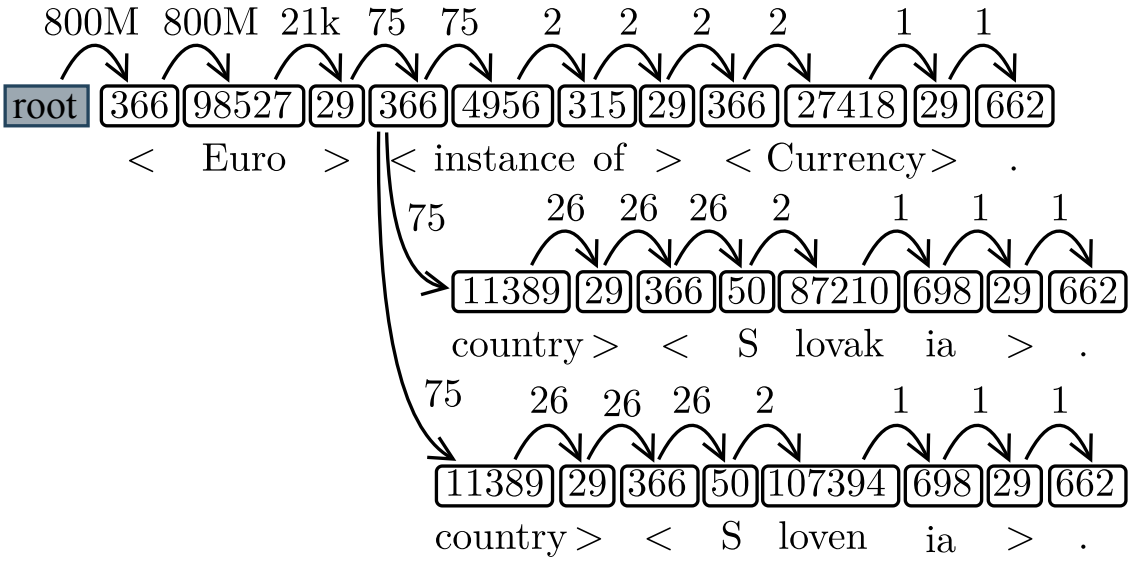


Figure 6.3: Generating facts from the fact tree. Token IDs are surrounded by rounded rectangles. The arrows $X \xrightarrow{n} v_k$ represent the selection of the next token v_k from the current sequence $X = x_{<i}$ and n is the number of leaves reachable from X.

Consider the prefix X = "<Euro> <country> <S" in Figure 6.3, where the number or reachable leaves is $numleaves(X) = 2$, duplicate facts generation is prevented as follows: after generating "<Euro> <country> <Slovakia> .", the number of reachable leaves is decreased to 1 and from X the model is only allowed to select "lovenia> .". At this point, all leaves reachable from X have been generated, thus generating "S" from X' = "<Euro> <country> <" is forbidden by constrained generation. From X' the LLM can generate one of the remaining $numleaves(X') - 2 = 24$ facts (e.g., "<Euro> <country> <Italy> .").

While relatively small trees can be kept in memory, e.g., using Python dictionaries, when dealing with bigger trees, such as the 800-million-fact tree derived from Wikidata, this is often impossible. Therefore, we rely on relational database software. PostgreSQL 17.2 [352] has been chosen and the fact-tree table was designed to ensure that all the information required for token generation is efficiently available. Given a prefix X, this includes the allowed next tokens $V_X^A = \{v_0, v_1, \ldots\}$, the number of leaves reachable from X, and the number of reachable leaves if choosing any $v \in V_X^A$ (to avoid selecting 0-leaves tokens and generating a fact twice). In Table 6.1, they are depicted respectively as *Next Tokens*, *#Lv. (number of leaves)*, and *Child.#Lv.*.

However, this tree-representation schema can quickly grow in disk space usage, as a single path of length L requires storing $L - 1$ rows. To reduce the disk space requirements, two additional measures are applied. First, after a certain prefix length L_c we stop adding rows and instead

directly save the subtree in Pickle format[7], as shown in the last row of Table 6.1. Consequently, at inference time subtrees are manageable in size and ca be directly loaded in memory. In our case, with $L_c = 7$, 99% of the subtrees use less than 116 kilobytes of memory. Secondly, single-leaf sequences are represented in a single row, as visible at the fourth row in Table 6.1, saving the rest of the sequence in *Child.#Lv.* (the token 29 comes after 694, and 662 is the last in the sequence). List items are saved as PostgreSQL arrays, while Pickle data is saved in `BYTEA`. For fast access, the *Prefix* is indexed with a B-Tree[8] index that provides logarithmic search time [73].

Table 6.1: PostgreSQL table content. *#Lv.* represent the number of leaves, reachable from that prefix.

Prefix	**Next Tokens**	**#Lv.**	**Child.#Lv.**	**Subtree**
{`root`}	{366,1134,...}	5M	{5M,3,...}	
{`root`}	{366,8730,...}	5M	{5M,9,...}	
{`root`,366}	{537,7350,...}	2M	{2M,7,...}	
{`root`,694}	{29,...,662}	1	{}	
{`root`,366,...}	{21538,4168}	7	{4,3}	`\x804`

The ingestion process consists of constructing the tree in memory and then persisting it to the database. However, hardware memory constraints prevent keeping the entire tree in memory at once. For this reason, ingestion is performed in batches: a limited-size tree is built, persisted, and then discarded before proceeding to the next batch.

This approach inevitably introduces duplicated prefix rows, since identical prefixes originating from different batches are stored as distinct rows and must be merged at inference time. In the worst case, the number of duplicates is bounded by the number of batches used. Despite this overhead, the proposed mechanisms allow indexing 800 million facts using approximately 95 GB of on-disk storage.

Note that the index stores token IDs and therefore depends on the vocabulary of the underlying LLM. As a consequence, LLMs with different vocabularies require separate indexes.

6.3 Embedding ReFactX into Question-Answering Workflows

ReFactX relies on In-Context Learning (ICL) [96] for instructing the model to access external knowledge. Our prompt, shown in Figure 6.4, instructs the LLM to, first, determine the reasoning path needed for answering, then to use the *Fact* command for getting relevant facts from the KB, and finally to answer based on the proofs acquired with the *Fact* command. Additionally, we instruct the model to:

1. determine the required answer type,
2. respond with "I don't know" when no relevant facts are found,
3. recognize when no useful facts can be retrieved and terminate generation,

[7] https://docs.python.org/3/library/pickle.html
[8] https://www.postgresql.org/docs/current/btree.html

4. strictly adhere to the provided prompt,
5. and be aware of the description predicate, which is often useful when querying the Wikidata KB.

At the end of this prompt, we add two examples to better guide the model behavior. While this number is not sufficient to represent all the possible types of questions, increasing it further may reduce the efficiency of our approach. Therefore, a two-shot prompt is used.

During inference, we detect when the LLM generates the sequence of tokens corresponding to the *Fact* command and we activate constrained generation; at this point, the model is forced to generate an existing fact. After an entire fact is generated, we switch the model back to normal generation, allowing it to either continue reasoning, to call again the *Fact* command, or to answer. An example of ReFactX in action is illustrated in Figure 6.1.

You are a helpful question-answering assistant that bases its answers on facts from a knowledge base and always respects the prompt.

The process to answer questions:

- You receive an input question.
- You determine the reasoning path needed to answer the question based on the information available.
- You determine the kind of answer you are asked. It can be a yes/no, a single entity, or a list of entities. Pay attention to the questions whose answer is a list of entities (e.g. Which countries share a border with Spain?): you need to find all the answer entities and include them all in the final answer.
- You get relevant facts with the **"Fact:"** command. You can rely on these facts and use them as proof for your answer. While getting facts you continue the reasoning explaining it step by step.
- Often description or short description may be useful for answering questions.
- You conclude with a concise answer that depending on the question can be a yes/no, a single entity, or a list of entities. Pay attention to the questions whose answer is a list of entities.
- The answer MUST be based on the proofs you found with **"Fact:"**.

If you didn't find proofs with **"Fact:"** that support an answer you stop and you reply: "I don't know.".

If the question requires to find proof that an event happen and you didn't find any proof, you can assume that event didn't happen.

In case you are taking too long for answering (e.g., you already generated ten facts that are not useful for the question), you stop and you answer based on the proofs you acquired to that point.

You must always follow these instructions precisely and ensure your responses adhere strictly to this prompt.

Figure 6.4: ReFactX system prompt for the Wikidata KB. The prompt instructs the LLM to reason step-by-step, issue *Fact* calls to access facts from the Wikidata fact tree, and deliver an answer only after evidence is gathered, or otherwise respond "I don't know."

6.4 Experimental Setup

In this section, we describe the experimental setup used to evaluate the performance of ReFactX on KGQA tasks and its adaptability to domain-specific data (financial). We detail, in turn, the pretrained LLMs, the datasets, the evaluation metrics, and the selected reference approaches.

In addition, we assess the scalability of ReFactX (*Ch. 1*) by comparing its generation time under constrained decoding with an unconstrained *LLM-only* baseline. This analysis is conducted on the 800-million-fact prefix tree constructed from Wikidata. We measure token generation time

(in seconds) over 4,000 generated tokens using Qwen2.5-3B [281] with PostgreSQL running on the same machine on one NVIDIA Tesla T4[9], and enabling key–value caching [11].

6.4.1 Underlying Models

The models selected for the evaluation are the following:

- *meta-llama/Llama-3.3-70B-Instruct* [281],
- *microsoft/phi-4* [1],
- *Qwen/Qwen2.5-72B-Instruct*,
- *Qwen/Qwen2.5-7B-Instruct* [281]

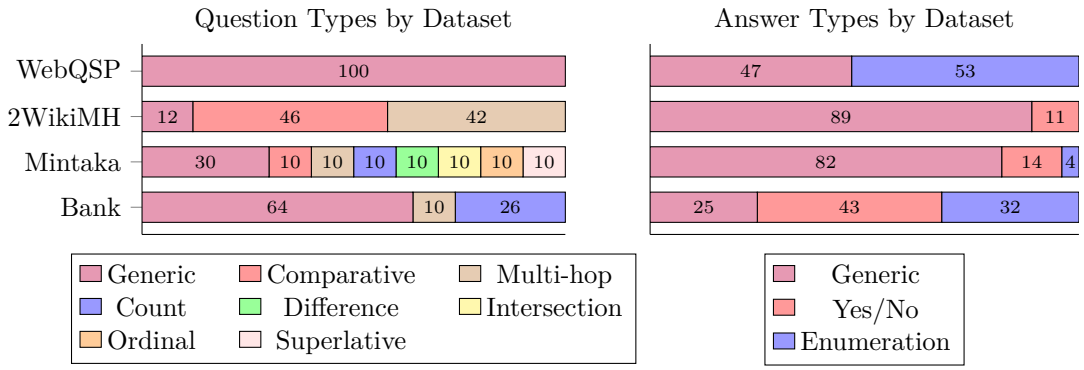


Figure 6.5: Question and answer type distribution across the four evaluation datasets. Stacked bars show how each benchmark (Bank, Mintaka, 2WikiMH, WebQSP) varies in its mix of question categories (generic, comparative, multi-hop, and others) and answer forms (generic, yes/no, enumeration).

6.4.2 Datasets

ReFactX is evaluated on three public and one proprietary benchmark datasets:

- Mintaka [322],
- 2WikiMultiHopQA [146],
- WebQSP [404],
- the "Bank" dataset: an anonymized proprietary financial dataset from Banca d'Italia[10].

[9] https://www.nvidia.com/en-us/data-center/tesla-t4/
[10] https://www.bancaditalia.it/

Mintaka [322] is a multilingual dataset containing nine question types annotated by crowdworkers with Wikidata IDs. In this work, we consider only English questions and merge *Yes/No* questions with *Generic* ones, resulting in eight question types. *Yes/No* is, instead, treated as an answer type, alongside *Generic* and *Enumeration*. This dataset allows us to analyze ReFactX's performance across diverse question types. Figure 6.5 shows the distribution of questions and answer types across all datasets.

2WikiMultiHopQA [146], to which we refer as 2WikiMH, is composed of multi-hop, comparative, and generic questions derived from Wikipedia and Wikidata. For this dataset, the evaluation considers the *development* set, as the ground truth for the test set is not publicly available.

WebQSP [404] contains generic questions annotated with Freebase [47]. Together with 2WikiMH, it is used by the existing KGQA approaches based on constrained generation [201, 215], providing insights into ReFactX's performance relative to prior work.

With these datasets we use the 800-million-fact tree derived from Wikidata indexed in PostgreSQL (Section 6.2), and, for computational efficiency, a limited sample of 200 questions is considered for each public dataset, stratifying by question type.

The proprietary *Bank* dataset comes from Banca d'Italia, the Italian central bank. It covers the financial domain—which is a knowledge-intensive domain—and allows us to study how ReFactX adapts to domain-specific knowledge.

It includes 278 template questions derived from an anonymized corporate knowledge graph containing approximately 10,000 triples and nine relations, such as ownership and control. In this setting, we can evaluate LLMs while minimizing data contamination issues, as the models are unlikely to have been exposed to this data during training. Triples are verbalized, additionally adding inverted facts so that ReFactX can use tail-to-head reasoning.

The choice of public benchmark datasets reflects two main considerations. First, datasets that natively use only or also include Wikidata are prioritized—namely Mintaka and 2WikiMH. Second, we included WebQSP, based on Freebase, to extend the comparison with prior work on constrained generation [201, 215]. Finally, we include the proprietary *Bank* dataset to assess performance on domain-specific data form a knowledge-intensive domain, free from data contamination.

6.4.3 Metrics

We evaluate on *Accuracy* (A) and *Precision* (P), to study, respectively, the overall correctness of our approach and its reliability when it provides an answer.

$$A = \frac{\text{Number of correct answers}}{\text{Number of questions}}, \tag{6.8}$$

$$P = \frac{\text{Number of correct answers}}{\text{Number of given answers}|}. \tag{6.9}$$

Accuracy (A) corresponds to the ratio of questions answered correctly. Precision (P), instead, is normalized by the number of provided answers, excluding "I don't know" responses and cases where no answer is produced because the maximum number of allowed new tokens is reached.

Precision and accuracy are calculated in two settings:

1. exact match: comparing the predicted answer with the ground truth with case-insensitive string equality;

2. LLM-as-a-judge [417]: we ask Llama3.3-70B in 16-bit precision whether the predicted answer is correct and complete with respect to the ground truth.

6.4.4 Reference Approaches

As reference approaches, we first consider the same LLMs used with ReFactX without constrained generation (*LLM-only*). We use the same configurations and prompts, so that models generates not-grounded facts based solely on their parametric-knowledge, giving us clues on how important is to access external knowledge and, furthermore, on how much each dataset can be answered using only the LLM's internal knowledge.

Then, we compare against QA methods that use constrained generation, specifically Decoding on Graphs (DoG) [201] and Graph-Constrained Reasoning (GCR) [215].

Additionally, Hybrid-QA (HQA) [195], a tool-based QA apprach, is included in the comparison. For these approaches, we report the results from the respective papers. HQA is evaluated on a 200-question sample of Mintaka, DoG and GCR evaluate on larger subsets of 2WikiMH and WebQSP: for 2WikiMH DoG uses 6,964 questions; for WebQSP respectively DoG and GCR consider 1,542 and 1,628 questions (filtered during preprocessing).

HQA [195] achieves state-of-the-art results on Mintaka [322]. Given a question, it first selects a limited set of few-shot examples maximizing their relevance for the question and the diversity between the examples, then, it instructs the LLM with ICL [96] to acquire external information with tools, such as a Wikipedia search engine and a Wikidata SPARQL query engine, and to finally answers the question.

DoG [201] and GCR [215] are both based on constrained generation, while they consider smaller question-based prefix trees with respect to ReFactX. DoG [201], incorporates constrained generated triples from a KG inside a reasoning process similar to CoT [373], instructing the model with ICL [96], and alternating normal and constrained generation. The KG, composed of up to 120 triples, is constructed for each question, from the triples within 2-4 hops from question entities. Then, at inference time, a query-centric subgraph of the KG is obtained from the triples containing the question entities (extracted with an entity linking system). The model is allowed to generate only the triples from this query-centric subgraph, and at each generation, the subgraph is expanded with all the adjacent triples, allowing the model to form a reasoning path from question entities to the answer.

GCR [215], instead, considers a pre-built subgraph of Freebase [47] of 8 million triples containing the entities mentioned in the evaluation questions, then for each questions it constructs a smaller prefix tree from the paths obtained with breadth first search within 2-hops from question entities (obtained with entity linking). At this point an LLM, fine-tuned for the task, generates multiple reasoning paths from the prefix tree with constrained decoding and an answer hypothesis for each path. Finally, a powerful LLM, such as GPT-4o-mini, receives all candidate paths and hypothesis answers and produces the final answer.

DoG and HQA use accuracy in the evaluation, while GCR calculates *Hit* and F_1. Considering *enumeration* answer (containing a list of items), *Hit* counts a prediction as correct whenever any item of the ground truth matches the prediction, whereas F_1 is the average of the F_1 of the single predictions. These measures are equivalent to accuracy for generic answers, but not for enumerations.

In the experiments, beam search [154] is used with number of beams = 3 (GCR uses 10 beams, DoG 3), sampling is disabled, and the maximum number of new tokens to generate is set to 1,000.

6.5 Results

6.5.1 Generation-Time Overhead

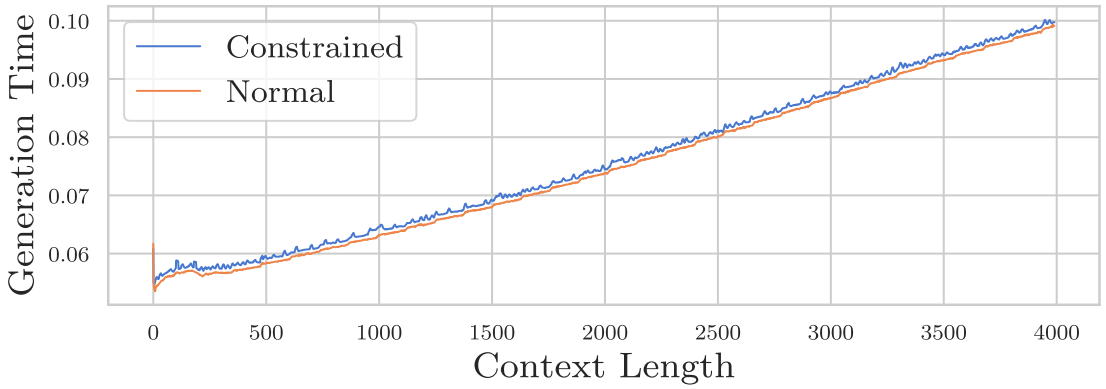


Figure 6.6: Per-token generation time (seconds) of constrained decoding with ReFactX versus unconstrained generation over 4,000 tokens. We apply a moving average with a 10-tokens window to reduce noise.

We compare ReFactX's KB-guided constrained generation time with normal *LLM-only* generation time. Figure 6.6 plots the token generation times calculated in the two settings. The time overhead added by constrained generation is very limited: the total time for generating 4,000 tokens increases by only 1.3%, from 300.14 seconds to 303.89.

6.5.2 Performance Analysis

Tables 6.2 and 6.3 show ReFactX results on benchmark datasets compared to *LLM-only*, calculated with exact match and LLMs-as-a-Judge, respectively. The gap between exact match and LLM-as-a-judge results derives from string comparison: correct answers with different date formats or item orders (for enumeration question) are counted as errors under exact match.

Compared to *LLM-only* models, ReFactX consistently achieves higher precision in the LLM-as-a-judge evaluation. In terms of accuracy, ReFactX performs worse than *LLM-only* on the Mintaka and WebQSP datasets. This reveals these datasets are largely covered by the models' parametric knowledge—reaching 82.0% and 80.5% accuracy respectively. Instead with 2WikiMH, whose questions seem harder for the LLM parametric knowledge, ReFactX improves both accuracy and precision by more than 20% with respect to *LLM-only* models.

The Bank dataset proved especially challenging, with ReFactX achieving precision up to 51.3% and accuracy up to 42.8%. However, a type-wise analysis reveals good results on generic Yes/No questions achieving precision (P) of 85.2% and accuracy (A) of 78.9 with Qwen2.5-72B while with Llama3.3-70B we achieve P = 77.8% and A = 70.6% (using LLM-as-a-judge in both cases). Our approach is, instead, particularly suffering with count questions: both models achieve less than 10% accuracy on these questions. Similarly, enumeration questions pose difficulties, with correct answers achieved in only 30% of cases on the Bank dataset.

Table 6.2: Comparison with *exact match* between ReFactX and *LLM-only* on four datasets, with four different LLMs (details in Section 6.4.1). We report Precision (P) and Accuracy (A).

	Exact Match							
	Bank		Mintaka		2WikiMH		WebQSP	
	P	A	P	A	P	A	P	A
ReFactX								
$textLlama^{3.3}_{70B}$	**40.8**	36.0	**66.4**	40.5	**74.4**	64.0	**22.8**	17.0
$textQwen^{2.5}_{72B}$	39.9	**37.1**	43.8	28.0	71.9	**69.0**	18.5	14.0
$textPhi^{4}_{14B}$	35.8	29.9	33.0	19.0	62.2	46.0	15.4	10.5
$textQwen^{2.5}_{7B}$	24.7	20.9	48.5	24.5	58.5	46.5	17.0	12.0
LLM-only								
$\text{Llama}^{3.3}_{70B}$	23.1	18.3	60.7	**59.5**	46.7	43.0	19.9	**19.5**
$\text{Qwen}^{2.5}_{72B}$	30.0	29.9	52.3	51.5	35.7	35.5	13.6	13.5
Phi^{4}_{14B}	21.8	20.1	43.6	42.5	25.5	24.0	14.4	14.0
$\text{Qwen}^{2.5}_{7B}$	21.7	18.0	45.3	41.0	21.6	18.0	10.9	10.5

Table 6.3: Comparison with *LLM-as-a-judge* between ReFactX and *LLM-only* on four datasets, considering four different LLMs (details in Section 6.4.1). We report Precision (P) and Accuracy (A).

	LLM-as-a-Judge							
	Bank		Mintaka		2WikiMH		WebQSP	
	P	A	P	A	P	A	P	A
ReFactX								
$\text{Llama}^{3.3}_{70B}$	50.0	**42.8**	**91.8**	56.0	93.6	81.0	85.2	63.5
$\text{Qwen}^{2.5}_{72B}$	46.1	**42.8**	82.8	55.5	**96.4**	**92.5**	84.8	65.5
Phi^{4}_{14B}	**51.3**	41.7	88.7	51.0	92.6	69.5	**89.7**	61.0
$\text{Qwen}^{2.5}_{7B}$	44.8	35.3	78.2	39.5	91.2	73.0	78.7	55.5
LLM-only								
$\text{Llama}^{3.3}_{70B}$	23.1	18.3	83.7	**82.0**	61.4	56.5	82.1	**80.5**
$\text{Qwen}^{2.5}_{72B}$	30.0	29.9	81.2	80.0	45.2	45.0	73.7	73.0
Phi^{4}_{14B}	25.7	23.7	76.9	75.0	41.5	39.0	75.3	73.0
$\text{Qwen}^{2.5}_{7B}$	28.1	21.9	65.7	60.0	35.9	30.0	66.7	64.0

On the Mintaka dataset, ReFactX struggles especially with superlative and count questions, with Llama3.3-70B achieving 14.3% and 55.0% accuracy, respectively. Qwen2.5-72B shows a contrasting pattern, achieving higher accuracy on superlatives (47.6%) but lower on counts (35.0%). On comparative and multi-hop questions, instead, Qwen2.5-72B achieves 52.4% and 57.9% accuracy, respectively, while Llama3.3-70B reaches 66.7% and 73.7%, which are still below the performance levels in 2WikiMH, composed mostly by comparative and multi-hop questions.

On WebQSP, enumeration answers prove particularly difficult in terms of precision for ReFactX with both Llama3.3-70B and Qwen2.5-72B, while when using Phi-4 it maintains consistent precision across all answer types.

Table 6.4: Insights from comparison with related work on Precision (P) and Accuracy (A). Results for related work are taken from their papers. ReFactX is evaluated using LLM-as-a-judge†, HQA via manual annotation‡. DoG and GCR use exact match*. GCR results are calculated with different metrics§ (see Section 6.4.3).

	Mintaka		**2WikiMH**		**WebQSP**	
	P	A	P	A	P	A
ReFactX $\text{Llama}_{70B}^{3.3}$ †	91.8	56.0	93.6	81.0	85.2	63.5
ReFactX $\text{Qwen}_{72B}^{2.5}$ †	82.8	55.5	96.4	**92.5**	84.8	65.5
ReFactX $\text{Qwen}_{7B}^{2.5}$ †	78.2	39.5	91.2	73.0	78.7	55.5
DoG[201]* $\text{Qwen}_{7B}^{2.5}$	–	–	–	84.2	–	**92.7**
GCR[215]* $\text{Llama}_{8B}^{3.1}$ + $\text{GPT}^{\text{4o-mini}}$	–	–	–	–	92.2§	74.1§
HQA[195] $\text{GPT}^{3.5}$ ‡	–	85.9	–	–	–	–
HQA[195] GPT^{4} ‡	–	**95.9**	–	–	–	–

Results from Table 6.4, showing ReFactX with reference approaches, suggest our approach can achieve competitive results. In fact, compared to DoG on 2WikiMH—although evaluated on different dataset samples—ReFactX, even if considering a large KB, achieves 73.0% accuracy with Qwen2.5-7B and 92.5% with Qwen2.5-72B, while DoG stands in the middle with 84.2%.

6.6 Discussion

The generation-time overhead measurements along with the results on the benchmark QA datasets demonstrate that LLMs with constrained generation and a prefix tree can effectively access external knowledge without any additional model, retriever, or external service. Additionally, by storing the prefix tree on disk using a database service, we are able to scale to a large knowledge base of 800 million facts derived from Wikidata. These advantages come with only a ~1% increase in generation time.

In particular, ReFactX shows greater effectiveness on the 2WikiMH dataset. This can be explained by the nature of the dataset, which requires only simple answers and contains only generic, comparative, and multi-hop questions (see Figure 6.5), easier to address with our approach. Indeed, answering these question types generally requires fewer facts with respect to count or enumeration questions. While for Mintaka multi-hop questions, we notice that in some cases they require ordinal reasoning, such as *"Where was the 16th president of the United States born?"*, making them harder for ReFactX.

On Mintaka and WebQSP, the datasets widely covered by the internal knowledge of the tested LLMs, when comparing ReFactX behavior with *LLM-only* on the same questions, ReFactX still demonstrates interesting reasoning patterns. In some cases, it uses facts to prove an answer coming from LLM parametric knowledge. In others, where *LLM-only* fails, ReFactX is able to acquire correct facts which lead to correcting model parametric knowledge.

However, ReFactX has some intrinsic limitations. The main one derives from the autoregressive left-to-right nature of LLMs. Indeed, ReFactX requires left-to-right facts, that start with known-information and lead to desired information. Furthermore, while we believe ReFactX is particularly useful to access point-wise factual information, count questions like *"How many movies have been directed by Danny Boyle?"* or *"Which NHL team has the most Stanley Cup wins?"* are particularly challenging, because they require ReFactX to enumerate a large list of facts and to understand when all the required facts have been generated.

This limitation likely explains why our approach is obtaining worse performance on count and superlative question types. Such question types could be better handled by using additional tools like SPARQL engines, which support counts or other set operations, or by investigating mechanisms to control ReFactX generation similarly as we did for preventing fact repetition.

The obtained results are particularly promising considering that we did not fine-tune the models to improve their ability to leverage ReFactX during reasoning. We plan to address this limitation in future work.

6.7 Findings and Open Directions

This chapter addressed *Obj. 3* by presenting ReFactX, a knowledge-base-constrained decoding mechanism that enables any autoregressive LLM to perform QA by exploiting large, external, or domain-specific KBs without relying on external retrievers or architectural modifications—an important use case in knowledge-intensive domains (*UC 4*). By grounding generation in explicit external facts, ReFactX inherently supports traceability and verifiability (*Ch. 3*), as every generated statement can be traced back to the underlying KB. In our experiments, we indexed more than 800 million Wikidata facts in a 95 GB on-disk prefix tree. By restricting decoding to valid continuations within this structure, ReFactX injects factual evidence during generation while incurring only about a 1.3% latency overhead, thereby directly addressing the scalability challenge (*Ch. 1*). Empirically, ReFactX improves QA performance over LLMs relying solely on their parametric knowledge, and comparisons with reference approaches show that it can achieve competitive results.

We further assessed the generalizability of ReFactX in a domain-specific, knowledge-intensive setting, namely corporate finance, by adapting the instruction prompt and indexing reversed triples to enable tail-to-head reasoning. Although overall performance on this dataset remains modest, simpler questions are handled substantially better, indicating headroom for improvement through stronger domain adaptation strategies, such as fine-tuning.

Future work includes enhancing the reasoning capabilities of ReFactX-enabled models through targeted fine-tuning, as well as extending support for count questions, long enumerations, and other set-based queries. In such scenarios, ReFactX could serve as a first-stage mechanism for factual acquisition, complemented by additional tools (e.g., a SPARQL engine) when more complex operations are required.

Finally, ReFactX operates on left-to-right verbalized facts and, although in this work such facts are derived from a structured KB (Wikidata), the proposed mechanism is, in principle, compatible with facts extracted from unstructured documents via relation extraction [172]. An especially

relevant direction for future work concerns, indeed, the extraction of facts from legal texts, where traceability to the original sources is critical. Evaluating this setting, and studying methods to guarantee traceability and verifiability with respect to the originating text passages, constitutes an important direction for future research, further addressing the traceability and verifiability challenges in knowledge-intensive domains (*Ch. 3*).

Chapter 7

Thesis Conclusion

This dissertation investigated how an entity-centric architecture can support the fulfillment of information needs in knowledge-intensive legal and investigative contexts. Building on the hypothesis that organizing knowledge around entities enables advanced and composable forms of information access, including document navigation or question answering, the work pursued three main objectives. First, it assessed the applicability and limitations of existing entity extraction methods in these domains, focusing on challenges such as the presence of *novel entities*, which are not represented in public knowledge repositories (*Obj. 1*). Second, it designed an entity-centric data integration architecture capable of integrating heterogeneous data sources and ensuring traceability, while supporting advanced information access applications such as faceted search and question answering (*Obj. 2*). Third, it developed an efficient and scalable question answering system, able to collect information from external knowledge bases and to generate verifiable answers grounded in the collected information (*Obj. 3*). Together, these objectives address the broader challenge of accessing and consolidating knowledge from heterogeneous, high-risk settings while preserving scalability, traceability, and human oversight.

Chapter 4 showed that incremental entity extraction—in which unknown entities, such as the persons mentioned in chat messages, are recognized and added to a knowledge repository as novel entities—suffers from error propagation and that NIL prediction (i.e., the detection of novel entity mentions) is a major source of errors. Moreover, while existing entity extraction models can be applied to legal documents and investigative chat data, in-domain fine-tuning or human-in-the-loop correction may be required for achieving satisfactory extraction quality.

Chapter 5 introduced an entity-centric architecture for integrating heterogeneous data sources—including judgments, investigative chats, and attachments—that remains effective even under imperfect extraction. It enables users to trace automatically extracted information back to its source and to correct it within a human-in-the-loop workflow. The architecture is grounded in real tasks and prior prototypes, and defines the abstractions required to support information access functionalities such as retrieval, navigation, conversational question answering, and statistical analysis over legal and investigative data. In addition, the chapter presented advanced user interfaces coupled with an implementation of the architecture, which operationalize the use cases discussed in this thesis, eventually combining them—for instance, combining chat-based question answering with faceted search by first filtering relevant documents and then asking questions.

Chapter 6 introduced ReFactX, showing that efficient and scalable question answering with traceable and verifiable answers can be achieved using an LLM with constrained generation sup-

ported by a disk-backed prefix tree. This approach avoids reliance on retrievers, complex pipelines, or architectural changes to the LLM, which may be impractical under budget or deployment constraints. ReFactX, with constrained generation, produces valid facts from large KBs, enabling users to verify that the answer is consistent with the grounded facts. This method scales to hundreds of millions of facts with negligible latency and yields substantial gains over LLMs relying solely on parametric knowledge.

Taken together, these results address the information access use cases considered for satisfying users' information needs in knowledge-intensive domains and, in particular, for sensitive contexts such as law and investigations, which require additional warranties, including traceability, verifiability, and error-correction capability. The data integration architecture presented in Chapter 5 supports the integration of extraction services, such as those evaluated in Chapter 4. If combined with a knowledge base that stores entity facts, it also supports ReFactX to answer questions over extracted facts, including queries related to the suspect's communication network described in Section 4.4.

Several directions remain open. First, extraction could be improved by advancing NIL prediction and clustering. Further gains may be obtained by adapting knowledge consolidation models with in-domain data and by evaluating newer approaches and LLMs, with particular interest in adaptation techniques such as in-context learning [172], which require minimal manual operations. Second, the architectural design could be implemented and evaluated. This includes the development of advanced UIs for currently uncovered use cases, such as statistical analysis (*UC 5*). Third, richer human-in-the-loop feedback loops could be explored, in which user corrections are not only recorded but also incorporated to update models or constraints, thereby reducing recurrent errors. Finally, methods for extracting facts from unstructured text while preserving traceability could be investigated. This would facilitate the integration of ReFactX into the data integration architecture in scenarios where data are predominantly unstructured.

7.1 Ethical and regulatory considerations

The design choices of this thesis align with the constraints governing the use of artificial intelligence (AI) in sensitive domains. In legal and investigative contexts, data often contain personal or confidential information and cannot be transferred to third-party API-based services without risking non-compliance with the GDPR (Regulation (EU) 2016/679) [107]. The GDPR requires lawful and controlled processing [107, Art. 5] and restricts international data transfers [107, Art. 44–49]. The AI Act (Regulation (EU) 2024/1689) [106] further classifies AI systems used in law enforcement and justice as high-risk [106, Annex III]. As a consequence, it mandates risk management, data governance, logging, and traceability [106, Art. 9–12]. Using LLMs through external APIs introduces additional compliance complexities. For example, a data processing agreement must be established between the controller and the processor [107, Art. 28(3)]. By contrast, the approaches developed in this thesis support local execution of question answering by explicitly considering efficiency and scalability. Together with the traceability, verifiability, and error-correction mechanisms ensured by the data integration architecture, they can keep processing within the secure perimeter of the deploying institution and preserve human oversight [106, Art. 14].

Bibliography

[1] Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, et al. *Phi-4 Technical Report*. arXiv preprint arXiv:2412.08905. 2024. DOI: 10.48550/arXiv.2412.08905. arXiv: 2412.08905 [cs.CL].

[2] Dhruv Agarwal, Rico Angell, Nicholas Monath, and Andrew McCallum. "Entity Linking and Discovery via Arborescence-based Supervised Clustering". In: *CoRR* abs/2109.01242 (2021). arXiv: 2109.01242. URL: https://arxiv.org/abs/2109.01242.

[3] Dhruv Agarwal, Rico Angell, Nicholas Monath, and Andrew McCallum. "Entity Linking via Explicit Mention-Mention Coreference Modeling". In: *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Seattle, United States: Association for Computational Linguistics, July 2022, pp. 4644–4658. DOI: 10.18653/v1/2022.naacl-main.343. URL: https://aclanthology.org/2022.naacl-main.343/.

[4] Ruben Agazzi, Renzo Alva Principe, Riccardo Pozzi, Marco Ripamonti, and Matteo Palmonari. "DAVE: A Framework for Assisted Analysis of Document Collections in Knowledge-Intensive Domains". In: *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, IJCAI-25*. Demo Track. International Joint Conferences on Artificial Intelligence Organization, Aug. 2025, pp. 10984–10988. DOI: 10.24963/ijcai.2025/1246. URL: https://doi.org/10.24963/ijcai.2025/1246.

[5] Microsoft Research AI4Science and Microsoft Azure Quantum. *The Impact of Large Language Models on Scientific Discovery: a Preliminary Study using GPT-4*. 2023. arXiv: 2311.07361 [cs.CL]. URL: https://arxiv.org/abs/2311.07361.

[6] Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. "FLAIR: An Easy-to-Use Framework for State-of-the-Art NLP". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 54–59. DOI: 10.18653/v1/N19-4010. URL: https://aclanthology.org/N19-4010/.

[7] Rana Salal Ali, Benjamin Zi Hao Zhao, Hassan Jameel Asghar, Tham Nguyen, Ian David Wood, and Mohamed Ali Kaafar. "Unintended Memorization and Timing Attacks in Named Entity Recognition Models". In: *Proceedings on Privacy Enhancing Technologies (PoPETs)* 2023.2 (2023), pp. 329–346. DOI: 10.56553/popets-2023-0056.

[8] Zeyuan Allen-Zhu and Yuanzhi Li. "Physics of language models: part 3.1, knowledge storage and extraction". In: ICML'24. Vienna, Austria: JMLR.org, 2024. DOI: 10.5555/3692070.3692115. URL: https://openreview.net/pdf?id=5x788rqbcj.

[9] Nasser Alshammari and Saad Alanazi. "The impact of using different annotation schemes on named entity recognition". In: *Egyptian Informatics Journal* 22.3 (2021), pp. 295–302. ISSN: 1110-8665. DOI: https://doi.org/10.1016/j.eij.2020.10.004. URL: https://www.sciencedirect.com/science/article/pii/S1110866520301596.

[10] Flora Amato, Antonino Mazzeo, Antonio Penta, and Antonio Picariello. "Using NLP and Ontologies for Notary Document Management Systems". In: *Database and Expert Systems Application, 2008. DEXA '08.* 2008, pp. 67–71. DOI: 10.1109/DEXA.2008.86.

[11] Sotiris Anagnostidis, Dario Pavllo, Luca Biggio, Lorenzo Noci, Aurelien Lucchi, and Thomas Hofmann. "Dynamic Context Pruning for Efficient and Interpretable Autoregressive Transformers". In: *Advances in Neural Information Processing Systems*. Vol. 36. Curran Associates, Inc., 2023, pp. 65202–65223. URL: https://proceedings.neurips.cc/paper_files/paper/2023/file/cdaac2a02c4fdcae77ba083b110efcc3-Paper-Conference.pdf.

[12] Carl F. Andersen et al. "KB Construction and Hypothesis Generation Using SAMSON". In: *Proceedings of the 12th Text Analysis Conference (TAC 2019)*. NIST, 2019. URL: https://tac.nist.gov/publications/2019/participant.papers/TAC2019.SAMSON.proceedings.pdf.

[13] James G Apple and Robert P Deyling. *A primer on the civil-law system*. Federal Judicial Center, 1995. URL: https://www.govinfo.gov/content/pkg/GOVPUB-JU13-PURL-LPS55055/pdf/GOVPUB-JU13-PURL-LPS55055.pdf.

[14] Negar Arabzadeh, Xinyi Yan, and Charles L. A. Clarke. "Predicting Efficiency/Effectiveness Trade-offs for Dense vs. Sparse Retrieval Strategy Selection". In: *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. CIKM '21. Virtual Event, Queensland, Australia: Association for Computing Machinery, 2021, pp. 2862–2866. ISBN: 9781450384469. DOI: 10.1145/3459637.3482159. URL: https://doi.org/10.1145/3459637.3482159.

[15] SM Archana, Jay Prakash, Pramod Kumar Singh, and Waquar Ahmed. "An effective biomedical named entity recognition by handling imbalanced data sets using deep learning and rule-based methods". In: *SN Computer Science* 4.5 (2023), p. 650. DOI: 10.1007/s42979-023-02068-6.

[16] Rosana Ardila et al. "Common Voice: A Massively-Multilingual Speech Corpus". eng. In: *Proceedings of the Twelfth Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association, May 2020, pp. 4218–4222. ISBN: 979-10-95546-34-4. URL: https://aclanthology.org/2020.lrec-1.520/.

[17] Marcelo G. Armentano, Daniela Godoy, Marcelo Campo, and Analia Amandi. "NLP-based faceted search: Experience in the development of a science and technology search engine". In: *Expert Systems with Applications* 41.6 (2014), pp. 2886–2896. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2013.10.023. URL: https://www.sciencedirect.com/science/article/pii/S0957417413008397.

[18] Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. "Self-RAG: Self-reflective Retrieval Augmented Generation". In: *NeurIPS 2023 Workshop on Instruction Tuning and Instruction Following*. 2023. URL: https://openreview.net/forum?id=jbNjgmE0OP.

[19] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. "DBpedia: A Nucleus for a Web of Open Data". In: *The Semantic Web*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 722–735. ISBN: 978-3-540-76298-0. DOI: 10.1007/978-3-540-76298-0_52.

[20] Gizem Aydin, Seyed Amin Tabatabaei, George Tsatsaronis, and Faegheh Hasibi. "Find the Funding: Entity Linking with Incomplete Funding Knowledge Bases". In: *Proceedings of the 29th International Conference on Computational Linguistics*. Gyeongju, Republic of Korea: International Committee on Computational Linguistics, Oct. 2022, pp. 1937–1942. URL: https://aclanthology.org/2022.coling-1.168/.

[21] Tom Ayoola, Shubhi Tyagi, Joseph Fisher, Christos Christodoulopoulos, and Andrea Pierleoni. "ReFinED: An Efficient Zero-shot-capable Approach to End-to-End Entity Linking". In: *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Industry Track*. Hybrid: Seattle, Washington + Online: Association for Computational Linguistics, July 2022, pp. 209–220. DOI: 10.18653/v1/2022.naacl-industry.24. URL: https://aclanthology.org/2022.naacl-industry.24/.

[22] Amit Bagga and Breck Baldwin. "Entity-Based Cross-Document Coreferencing Using the Vector Space Model". In: *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 1*. Montreal, Quebec, Canada: Association for Computational Linguistics, Aug. 1998, pp. 79–85. DOI: 10.3115/980845.980859. URL: https://aclanthology.org/P98-1012/.

[23] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. "Neural Machine Translation by Jointly Learning to Align and Translate". In: *3rd International Conference on Learning Representations (ICLR)*. 2015. DOI: https://doi.org/10.48550/arXiv.1409.0473.

[24] Krisztian Balog. *Entity-Oriented Search*. Vol. 39. The Information Retrieval Series. Springer, 2018. ISBN: 978-3-319-93933-9. DOI: 10.1007/978-3-319-93935-3. URL: https://doi.org/10.1007/978-3-319-93935-3.

[25] Debayan Banerjee, Debanjan Chaudhuri, Mohnish Dubey, and Jens Lehmann. "PNEL: Pointer Network Based End-To-End Entity Linking over Knowledge Graphs". In: *The Semantic Web – ISWC 2020*. Cham: Springer International Publishing, 2020, pp. 21–38. ISBN: 978-3-030-62419-4. DOI: 10.1007/978-3-030-62419-4_2.

[26] Edoardo Barba, Luigi Procopio, and Roberto Navigli. "ExtEnD: Extractive Entity Disambiguation". In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 2478–2488. DOI: 10.18653/v1/2022.acl-long.177. URL: https://aclanthology.org/2022.acl-long.177/.

[27] Pierpaolo Basile, Annalina Caputo, Anna Lisa Gentile, and Giuseppe Rizzo. "Overview of the EVALITA 2016 Named Entity rEcognition and Linking in Italian Tweets (NEEL-IT) Task". In: *Proceedings of the Third Italian Conference on Computational Linguistics (CLiC-it 2016) & Fifth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2016)*. Vol. 1749. Napoli, Italy, Dec. 2016. URL: https://ceur-ws.org/Vol-1749/paper_007.pdf.

[28] Madeleine Bates. "Models of natural language understanding." In: *Proceedings of the National Academy of Sciences* 92.22 (1995), pp. 9977–9982. URL: https://www.pnas.org/doi/pdf/10.1073/pnas.92.22.9977.

[29] Carlo Batini, Valerio Bellandi, Paolo Ceravolo, Federico Moiraghi, Matteo Palmonari, and Stefano Siccardi. "Semantic Data Integration for Investigations: Lessons Learned and Open Challenges". In: *2021 IEEE International Conference on Smart Data Services (SMDS)*. IEEE. 2021, pp. 173–183. DOI: 10.1109/SMDS53860.2021.00031.

[30] Nicole Lang Beebe and Jan Guynes Clark. "Digital forensic text string searching: Improving information retrieval effectiveness by thematically clustering search results". In: *Digital Investigation* 4 (2007), pp. 49–54. ISSN: 1742-2876. DOI: https://doi.org/10.1016/j.diin.2007.06.005. URL: https://www.sciencedirect.com/science/article/pii/S1742287607000412.

[31] Valerio Bellandi, Silvana Castano, Stefano Montanelli, and Stefano Siccardi. "Streamlining Legal Document Management: A Knowledge-Driven Service Platform". In: *SN Computer Science* 6.2 (2025), pp. 1–17. DOI: 10.1007/s42979-025-03694-y.

[32] Valerio Bellandi et al. "An entity-centric approach to manage court judgments based on Natural Language Processing". In: *Computer Law & Security Review* 52 (2024). All authors contributed equally, p. 105904. ISSN: 0267-3649. DOI: https://doi.org/10.1016/j.clsr.2023.105904. URL: https://www.sciencedirect.com/science/article/pii/S0267364923001140.

[33] Iz Beltagy, Matthew E. Peters, and Arman Cohan. *Longformer: The Long-Document Transformer*. 2020. arXiv: 2004.05150 [cs.CL]. URL: https://arxiv.org/abs/2004.05150.

[34] Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. "On the Dangers of Stochastic Parrots: Can Language Models Be Too Big? ". In: *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*. FAccT '21. Virtual Event, Canada: Association for Computing Machinery, 2021, pp. 610–623. ISBN: 9781450383097. DOI: 10.1145/3442188.3445922. URL: https://doi.org/10.1145/3442188.3445922.

[35] Yoshua Bengio, Aaron Courville, and Pascal Vincent. "Representation Learning: A Review and New Perspectives". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 35.8 (Aug. 2013), pp. 1798–1828. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2013.50. URL: https://doi.org/10.1109/TPAMI.2013.50.

[36] Tim Berners-Lee, Roy Fielding, and Larry Masinter. *Uniform Resource Identifier (URI): Generic Syntax*. Internet STD 66, RFC 3986. 2005. URL: http://www.ietf.org/rfc/rfc2396.txt.

[37] Gabriel Bernier-Colborne, Caroline Barrière, and P. Ménard. "CRIM's Systems for the Trilingual Entity Detection and Linking Task". In: *Proceedings of the 10th Text Analysis Conference (TAC 2017)*. NIST, 2017. URL: https://tac.nist.gov/publications/2017/participant.papers/TAC2017.CRIM.proceedings.pdf.

[38] Michele Bevilacqua, Giuseppe Ottaviano, Patrick Lewis, Scott Yih, Sebastian Riedel, and Fabio Petroni. "Autoregressive Search Engines: Generating Substrings as Document Identifiers". In: *Advances in Neural Information Processing Systems*. Vol. 35. Curran Associates, Inc., 2022, pp. 31668–31683. URL: https://proceedings.neurips.cc/paper_files/paper/2022/file/cd88d62a2063fdaf7ce6f9068fb15dcd-Paper-Conference.pdf.

[39] G P Shrivatsa Bhargav et al. "Zero-shot Entity Linking with Less Data". In: *Findings of the Association for Computational Linguistics: NAACL 2022*. Seattle, United States: Association for Computational Linguistics, July 2022, pp. 1681–1697. DOI: 10.18653/v1/2022.findings-naacl.127. URL: https://aclanthology.org/2022.findings-naacl.127/.

[40] Joanna Biega, Erdal Kuzey, and Fabian M. Suchanek. "Inside YAGO2s: a transparent information extraction architecture". In: *Proceedings of the 22nd International Conference on World Wide Web*. WWW '13 Companion. Rio de Janeiro, Brazil: Association for Computing Machinery, 2013, pp. 325–328. ISBN: 9781450320382. DOI: 10.1145/2487788.2487935. URL: https://doi.org/10.1145/2487788.2487935.

[41] Christian Bizer et al. "DBpedia - A crystallization point for the Web of Data". In: *Journal of Web Semantics* 7.3 (2009). The Web of Data, pp. 154–165. ISSN: 1570-8268. DOI: https://doi.org/10.1016/j.websem.2009.07.002. URL: https://www.sciencedirect.com/science/article/pii/S1570826809000225.

[42] Kevin Blissett and Heng Ji. "Cross-lingual NIL Entity Clustering for Low-resource Languages". In: *Proceedings of the Second Workshop on Computational Models of Reference, Anaphora and Coreference*. Minneapolis, USA: Association for Computational Linguistics, June 2019, pp. 20–25. DOI: 10.18653/v1/W19-2804. URL: https://aclanthology.org/W19-2804/.

[43] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. "Fast unfolding of communities in large networks". In: *Journal of Statistical Mechanics: Theory and Experiment* 2008.10 (2008). DOI: 10.1088/1742-5468/2008/10/P10008.

[44] Guido Boella, Luigi Di Caro, Llio Humphreys, Livio Robaldo, Piercarlo Rossi, and Leendert Van Der Torre. "Eunomos, a legal document and knowledge management system for the web to provide relevant, reliable and up-to-date information on the law". In: *Artificial Intelligence and Law* 24.3 (2016), pp. 245–283. DOI: 10.1007/s10506-016-9184-3. URL: https://link.springer.com/content/pdf/10.1007/s10506-016-9184-3.pdf.

[45] Guido Boella, Luigi Di Caro, and Valentina Leone. "Semi-automatic knowledge population in a legal document management system". In: *Artificial intelligence and Law* 27.2 (2019), pp. 227–251. DOI: 10.1007/s10506-018-9239-8. URL: https://link.springer.com/content/pdf/10.1007/s10506-018-9239-8.pdf.

[46] Sergei Bogdanov, Alexandre Constantin, Timothée Bernard, Benoit Crabbé, and Etienne P Bernard. "NuNER: Entity Recognition Encoder Pretraining via LLM-Annotated Data". In: Miami, Florida, USA: Association for Computational Linguistics, Nov. 2024, pp. 11829–11841. DOI: 10.18653/v1/2024.emnlp-main.660. URL: https://aclanthology.org/2024.emnlp-main.660/.

[47] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. "Freebase: a collaboratively created graph database for structuring human knowledge". In: SIGMOD '08. Vancouver, Canada: Association for Computing Machinery, 2008, pp. 1247–1250. ISBN: 9781605581026. DOI: 10.1145/1376616.1376746. URL: https://doi.org/10.1145/1376616.1376746.

[48] Sebastian Borgeaud et al. "Improving Language Models by Retrieving from Trillions of Tokens". In: *Proceedings of the 39th International Conference on Machine Learning*. Vol. 162. Proceedings of Machine Learning Research. PMLR, 17–23 Jul 2022, pp. 2206–2240. URL: https://proceedings.mlr.press/v162/borgeaud22a.html.

[49] Tim Bray. *The JavaScript Object Notation (JSON) Data Interchange Format*. RFC 8259. Dec. 2017. DOI: 10.17487/RFC8259. URL: https://www.rfc-editor.org/info/rfc8259.

[50] Alexander Braylan, Omar Alonso, and Matthew Lease. "Measuring Annotator Agreement Generally across Complex Structured, Multi-object, and Free-text Annotation Tasks". In: *Proceedings of the ACM Web Conference 2022*. WWW '22. Virtual Event, Lyon, France: Association for Computing Machinery, 2022, pp. 1720–1730. ISBN: 9781450390965. DOI: 10.1145/3485447.3512242. URL: https://doi.org/10.1145/3485447.3512242.

[51] Anna Breit, Laura Waltersdorfer, Fajar J. Ekaputra, and Marta Sabou. "An Architecture for Extracting Key Elements from Legal Permits". In: *2020 IEEE International Conference on Big Data (Big Data)*. 2020, pp. 2105–2110. DOI: 10.1109/BigData50022.2020.9378375.

[52] Joost Breuker, André Valente, and Radboud Winkels. "Legal ontologies in knowledge engineering and information management". In: *Artificial intelligence and law* 12.4 (2004), pp. 241–277. DOI: 10.1007/s10506-006-0002-1.

[53] Samuel Broscheit. "Investigating Entity Knowledge in BERT with Simple Neural End-To-End Entity Linking". In: *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)* (2019). DOI: 10.18653/v1/k19-1063. URL: http://dx.doi.org/10.18653/v1/K19-1063.

[54] Tom Brown et al. "Language Models are Few-Shot Learners". In: *Advances in Neural Information Processing Systems*. Vol. 33. Curran Associates, Inc., 2020, pp. 1877–1901. DOI: 10.5555/3495724.3495883. URL: https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

[55] Maria G. Buey, Angel Luis Garrido, Carlos Bobed, and Sergio Ilarri. "The AIS Project: Boosting Information Extraction from Legal Documents by using Ontologies". In: *Proceedings of the 8th International Conference on Agents and Artificial Intelligence (ICAART 2016)*. Vol. 2. 2016, pp. 438–445. DOI: 10.5220/0005757204380445.

[56] Razvan Bunescu and Marius Paca. "Using Encyclopedic Knowledge for Named entity Disambiguation". In: *11th Conference of the European Chapter of the Association for Computational Linguistics*. Trento, Italy: Association for Computational Linguistics, Apr. 2006, pp. 9–16. URL: https://aclanthology.org/E06-1002/.

[57] Michael Caballero. "A brief survey of question answering systems". In: *International Journal of Artificial Intelligence & Applications (IJAIA)* 12.5 (2021). URL: https://papers.ssrn.com/sol3/Delivery.cfm/SSRN_ID3996229_code3122004.pdf?abstractid=3996229&mirid=1.

[58] Cambridge Dictionary. *Knowledge-intensive*. URL: https://dictionary.cambridge.org/us/dictionary/english/knowledge-intensive (visited on 10/08/2025).

[59] Cambridge Dictionary. *Nil*. Accessed September 28, 2025. 2025. URL: https://dictionary.cambridge.org/dictionary/english/nil.

[60] Nicola De Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni. "Autoregressive Entity Retrieval". In: *International Conference on Learning Representations*. 2021. URL: https://openreview.net/forum?id=5k8F6UU39V.

[61] Cristian Cardellino, Milagro Teruel, Laura Alonso Alemany, and Serena Villata. "A low-cost, high-coverage legal named entity recognizer, classifier and linker". In: ICAIL '17. London, United Kingdom: Association for Computing Machinery, 2017, pp. 9–18. ISBN: 9781450348911. DOI: 10.1145/3086512.3086514. URL: https://hal.science/hal-01541446v1/document.

[62] Scott Carlson. *Introduction to Civil Law Legal Systems*. Report. 2009. URL: https://www.fjc.gov/sites/default/files/2015/Introduction%20to%20Civil%20Law%20Legal%20Systems.pdf.

[63] Imdat Celeste. *M-AILABS Speech Dataset*. URL: https://github.com/imdatceleste/m-ailabs-dataset (visited on 10/14/2025).

[64] Can Çetinda, Berkay Yazcolu, and Aykut Koç. "Named-entity recognition in Turkish legal texts". In: *Natural Language Engineering* 29.3 (2023), pp. 615–642. DOI: 10.1017/S1351324922000304.

[65] Mohamed Chabchoub, Michel Gagnon, and Amal Zouaq. "FICLONE: Improving DBpedia Spotlight Using Named Entity Recognition and Collective Disambiguation". In: *Open Journal of Semantic Web (OJSW)* 5.1 (2018), pp. 12–28. ISSN: 2199-336X. URL: http://nbn-resolving.de/urn:nbn:de:101:1-201808051930147807663.

[66] Ilias Chalkidis, Ion Androutsopoulos, and Achilleas Michos. "Extracting contract elements". In: ICAIL '17. London, United Kingdom: Association for Computing Machinery, 2017, pp. 19–28. ISBN: 9781450348911. DOI: 10.1145/3086512.3086515. URL: https://doi.org/10.1145/3086512.3086515.

[67] Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. "LEGAL-BERT: The Muppets straight out of Law School". In: *Findings of the Association for Computational Linguistics: EMNLP 2020*. Online: Association for Computational Linguistics, Nov. 2020, pp. 2898–2904. DOI: 10.18653/v1/2020.findings-emnlp.261. URL: https://aclanthology.org/2020.findings-emnlp.261/.

[68] Hoyeon Chang et al. "How Do Large Language Models Acquire Factual Knowledge During Pretraining?" In: *Advances in Neural Information Processing Systems*. Vol. 37. Curran Associates, Inc., 2024, pp. 60626–60668. URL: https://proceedings.neurips.cc/paper_files/paper/2024/file/6fdf57c71bc1f1ee29014b8dc52e723f-Paper-Conference.pdf.

[69] Tianqi Chen and Carlos Guestrin. "XGBoost: A Scalable Tree Boosting System". In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '16. San Francisco, California, USA: Association for Computing Machinery, 2016, pp. 785–794. ISBN: 9781450342322. DOI: 10.1145/2939672.2939785. URL: https://doi.org/10.1145/2939672.2939785.

[70] Nancy A. Chinchor. "Overview of MUC-7". In: *Seventh Message Understanding Conference (MUC-7): Proceedings of a Conference Held in Fairfax, Virginia, April 29 - May 1, 1998*. 1998. URL: https://aclanthology.org/M98-1001/.

[71] Roi Cohen, Eden Biran, Ori Yoran, Amir Globerson, and Mor Geva. "Evaluating the Ripple Effects of Knowledge Editing in Language Models". In: 12 (2024), pp. 283–298. DOI: 10.1162/tacl_a_00644. URL: https://aclanthology.org/2024.tacl-1.16/.

[72] Common Crawl. *Common Crawl Corpus*. Common Crawl. 2025. URL: https://commoncrawl.org/overview (visited on 10/05/2025).

[73] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms, 3rd Edition*. MIT Press, 2009. ISBN: 978-0-262-03384-8. URL: http://mitpress.mit.edu/books/introduction-algorithms.

[74] *Costituzione della Repubblica Italiana (Constitution of the Italian Republic)*. https://www.normattiva.it/uri-res/N2Ls?urn:nir:stato:costituzione. 1947.

[75] Council of Europe, European Commission for the Efficiency of Justice (CEPEJ). *European Ethical Charter on the Use of Artificial Intelligence in Judicial Systems and Their Environment*. Adopted at the 31st plenary meeting of the CEPEJ (Strasbourg, 3–4 December 2018). Council of Europe. Dec. 2018. URL: https://rm.coe.int/ethical-charter-en-for-publication-4-december-2018/16808f699c (visited on 10/05/2025).

[76] Marialuisa Cugno, Silvia Giacomelli, Laura Malgieri, Sauro Mocetti, and Giuliana Palumbo. *No. 715 - Civil justice in Italy, length of proceedings, productivity of the courts and stability of judgments*. Banca d'Italia, Questioni di Economia e Finanza, n. 715. 2022. URL: https://www.bancaditalia.it/pubblicazioni/qef/2022-0715/index.html?com.dotmarketing.htmlpage.language=1&dotcache=refresh (visited on 10/02/2025).

[77] Hamish Cunningham, Diana Maynard, Kalina Bontcheva, and Valentin Tablan. "GATE: an Architecture for Development of Robust HLT applications". In: *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. Philadelphia, Pennsylvania, USA: Association for Computational Linguistics, July 2002, pp. 168–175. DOI: 10.3115/1073083.1073112. URL: https://aclanthology.org/P02-1022/.

[78] *Cypher Query Language Manual — Introduction*. Neo4j, Inc. URL: https://neo4j.com/docs/cypher-manual/current/introduction/ (visited on 10/19/2025).

[79] Joachim Daiber, Max Jakob, Chris Hokamp, and Pablo N. Mendes. "Improving efficiency and accuracy in multilingual entity extraction". In: *Proceedings of the 9th International Conference on Semantic Systems*. I-SEMANTICS '13. Graz, Austria: Association for Computing Machinery, 2013, pp. 121–124. ISBN: 9781450319720. DOI: 10.1145/2506182.2506198. URL: https://jodaiber.de/doc/entity.pdf.

[80] Fred J. Damerau. "A technique for computer detection and correction of spelling errors". In: *Commun. ACM* 7.3 (Mar. 1964), pp. 171–176. ISSN: 0001-0782. DOI: 10.1145/363958.363994. URL: https://doi.org/10.1145/363958.363994.

[81] DBpedia Community. *DBpedia*. 2025. URL: https://www.dbpedia.org/ (visited on 08/25/2025).

[82] Nicola De Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni. "Autoregressive Entity Retrieval". In: *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

[83] Nicola De Cao et al. "Multilingual Autoregressive Entity Linking". In: *Transactions of the Association for Computational Linguistics* 10 (2022), pp. 274–290. DOI: 10.1162/tacl_a_00460. URL: https://aclanthology.org/2022.tacl-1.16/.

[84] S. Decker et al. "The Semantic Web: the roles of XML and RDF". In: *IEEE Internet Computing* 4.5 (2000), pp. 63–73. DOI: 10.1109/4236.877487.

[85] *Decreto del Presidente della Repubblica 22 settembre 1988, n. 447, Codice di Procedura Penale (Italian Code of Criminal Procedure)*. https://www.normattiva.it/uri-res/N2Ls?urn:nir:stato:decreto.del.presidente.della.repubblica:1988-09-22;447. 1988.

[86] DeepSeek-AI et al. *DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning*. 2025. arXiv: 2501.12948 [cs.CL]. URL: https://arxiv.org/abs/2501.12948.

[87] DeepSeek-AI et al. *DeepSeek-V3 Technical Report*. 2025. arXiv: 2412.19437 [cs.CL]. URL: https://arxiv.org/abs/2412.19437.

[88] Mohammad Dehghan et al. "EWEK-QA : Enhanced Web and Efficient Knowledge Graph Retrieval for Citation-based Question Answering Systems". In: *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Bangkok, Thailand: Association for Computational Linguistics, Aug. 2024, pp. 14169–14187. DOI: 10.18653/v1/2024.acl-long.764. URL: https://aclanthology.org/2024.acl-long.764/.

[89] Louise Deleger et al. "Building gold standard corpora for medical natural language processing tasks". In: *AMIA Annual Symposium Proceedings*. Vol. 2012. American Medical Informatics Association. 2012. URL: https://pmc.ncbi.nlm.nih.gov/articles/PMC3540456/pdf/amia_2012_symp_0144.pdf.

[90] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. "BERT: Pretraining of Deep Bidirectional Transformers for Language Understanding". In: Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 4171–4186. DOI: 10.18653/v1/N19-1423.

[91] Beniamino Di Martino et al. "Semantic Based Knowledge Management in e-Government Document Workflows: A Case Study for Judiciary Domain in Road Accident Trials". In: *Complex, Intelligent and Software Intensive Systems*. Cham: Springer International Publishing, 2022, pp. 435–445. ISBN: 978-3-031-08812-4. DOI: 10.1007/978-3-031-08812-4_42.

[92] Yifan Ding, Qingkai Zeng, and Tim Weninger. "ChatEL: Entity Linking with Chatbots". In: *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*. Torino, Italia: ELRA and ICCL, May 2024, pp. 3086–3097. URL: https://aclanthology.org/2024.lrec-main.275/.

[93] George Doddington, Alexis Mitchell, Mark Przybocki, Lance Ramshaw, Stephanie Strassel, and Ralph Weischedel. "The Automatic Content Extraction (ACE) Program – Tasks, Data, and Evaluation". In: *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*. Lisbon, Portugal: European Language Resources Association (ELRA), May 2004. URL: https://aclanthology.org/L04-1011/.

[94] Elham Dolatabadi et al. "Natural Language Processing for Clinical Laboratory Data Repository Systems: Implementation and Evaluation for Respiratory Viruses". In: *JMIR AI* 2 (June 2023), e44835. ISSN: 2817-1705. DOI: 10.2196/44835. URL: https://doi.org/10.2196/44835.

[95] Hang Dong, Jiaoyan Chen, Yuan He, Yinan Liu, and Ian Horrocks. "Reveal the Unknown: Out-of-Knowledge-Base Mention Discovery with Entity Linking". In: *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*. CIKM '23. Birmingham, United Kingdom: Association for Computing Machinery, 2023, pp. 452–462. ISBN: 9798400701245. DOI: 10.1145/3583780.3615036. URL: https://doi.org/10.1145/3583780.3615036.

[96] Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, et al. "A Survey on In-context Learning". In: *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. 2024, pp. 1107–1128. DOI: 10.18653/v1/2024.emnlp-main.64.

[97] Kelvin Du, Yazhi Zhao, Rui Mao, Frank Xing, and Erik Cambria. "Natural language processing in finance: A survey". In: *Information Fusion* 115 (2025), p. 102755. ISSN: 1566-2535. DOI: https://doi.org/10.1016/j.inffus.2024.102755. URL: https://www.sciencedirect.com/science/article/pii/S1566253524005335.

[98] Mohnish Dubey, Debayan Banerjee, Debanjan Chaudhuri, and Jens Lehmann. "EARL: Joint Entity and Relation Linking for Question Answering over Knowledge Graphs". In: *The Semantic Web – ISWC 2018*. Cham: Springer International Publishing, 2018, pp. 108–126. ISBN: 978-3-030-00671-6. DOI: 10.1007/978-3-030-00671-6_7.

[99] Martin Dürst and Michel Suignard. *Internationalized Resource Identifiers (IRIs)*. RFC 3987. 2005. URL: http://www.ietf.org/rfc/rfc3987.

[100] Darren Edge et al. *From Local to Global: A Graph RAG Approach to Query-Focused Summarization*. 2025. arXiv: 2404.16130 [cs.CL]. URL: https://arxiv.org/abs/2404.16130.

[101] Alfonso Egea-de Haro. "How Does Case Law Shape Civil Law Systems? An Analysis of Spanish Administrative Courts". In: *Liverpool Law Review* 45.1 (2024), pp. 1–23. ISSN: 1572-8625. DOI: 10.1007/s10991-023-09325-x. URL: https://doi.org/10.1007/s10991-023-09325-x.

[102] Ramez Elmasri and Shamkant B. Navathe. *Fundamentals of database systems*. Vol. 7. Pearson, 2014.

[103] Ahmed Elnaggar, Robin Otto, and Florian Matthes. "Deep Learning for Named-Entity Linking with Transfer Learning for Legal Documents". In: *Proceedings of the 2018 Artificial Intelligence and Cloud Computing Conference*. AICCC '18. Tokyo, Japan: Association for Computing Machinery, 2018, pp. 23–28. ISBN: 9781450366236. DOI: 10.1145/3299819.3299846. URL: https://doi.org/10.1145/3299819.3299846.

[104] Encyclopedia.com. *Information Access*. 2025. URL: https://www.encyclopedia.com/computing/news-wires-white-papers-and-books/information-access (visited on 08/23/2025).

[105] Patrick Th. Eugster, Pascal A. Felber, Rachid Guerraoui, and Anne-Marie Kermarrec. "The many faces of publish/subscribe". In: *ACM Comput. Surv.* 35.2 (June 2003), pp. 114–131. ISSN: 0360-0300. DOI: 10.1145/857076.857078. URL: https://doi.org/10.1145/857076.857078.

[106] European Union. *Artificial Intelligence Act (Regulation (EU) 2024/1689)*. 2024. URL: https://eur-lex.europa.eu/eli/reg/2024/1689/oj.

[107] European Union. *General Data Protection Regulation (GDPR, Regulation (EU) 2016/679)*. 2016. URL: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A32016R0679.

[108] EXFILES - Extract Forensic Information for Law Enforcement Agencies from Encrypted Smart Phones. *State of the Art in Mobile Forensics*. Tech. rep. D1.1. EU Horizon 2020 Project No. 883156. 2020. URL: https://exfiles.eu/wp-content/uploads/2022/07/EXFILES-D1.1-State-of-the-art-in-mobile-forensics-PU-M03.pdf.

[109] Explosion AI. *spaCy en_core_web_sm NER Label Scheme*. 2025. URL: https://spacy.io/models/en#en_core_web_sm-labels (visited on 08/21/2025).

[110] Angela Fahrni, Thierry Göckel, and M. Strube. "HITS' Monolingual and Cross-lingual Entity Linking System at TAC 2012: A Joint Approach". In: *Proceedings of the Fifth Text Analysis Conference (TAC 2012)*. NIST, Nov. 2012. URL: https://tac.nist.gov/publications/2012/participant.papers/HITS.proceedings.pdf.

[111] Nicolas Rodolfo Fauceglia, Yiu-Chang Lin, Xuezhe Ma, and Eduard H. Hovy. "CMU System for Entity Discovery and Linking at TAC-KBP 2015". In: *Proceedings of the 8th Text Analysis Conference (TAC 2015)*. NIST, 2015. URL: http://www.cs.cmu.edu/~xuezhem/publications/TAC2015.pdf.

[112] Yi Feng, Chuanyi Li, and Vincent Ng. "Legal Case Retrieval: A Survey of the State of the Art". In: *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Bangkok, Thailand: Association for Computational Linguistics, Aug. 2024, pp. 6472–6485. DOI: 10.18653/v1/2024.acl-long.350. URL: https://aclanthology.org/2024.acl-long.350/.

[113] Flair NLP. *Flair OntoNotes English NER Model: Entity Types*. 2025. URL: https://huggingface.co/flair/ner-english-ontonotes (visited on 08/21/2025).

[114] Raymond Fok et al. "Scim: Intelligent Skimming Support for Scientific Papers". In: *Proceedings of the 28th International Conference on Intelligent User Interfaces*. IUI '23. New York, NY, USA: Association for Computing Machinery, 2023, pp. 476–490. ISBN: 9798400701061. DOI: 10.1145/3581641.3584034. URL: https://doi.org/10.1145/3581641.3584034.

[115] Matthew Francis-Landau, Greg Durrett, and Dan Klein. "Capturing Semantic Similarity for Entity Linking with Convolutional Neural Networks". In: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. San Diego, California: Association for Computational Linguistics, June 2016, pp. 1256–1261. DOI: 10.18653/v1/N16-1150. URL: https://aclanthology.org/N16-1150/.

[116] Federico Galli. *L'analisi automatica delle decisioni giudiziali*. Giappichelli, 2025. URL: https://www.giappichelli.it/media/catalog/product/openaccess/9791221162769.pdf.

[117] Federico Galli, Alessia Fidelangeli, Piera Santin, Galileo Sartor, et al. "Analytics for Deciding Legal Cases: The ADELE Project". In: *Artificial Intelligence, Judicial Decision-Making and Fundamental Rights*. Scuola Superiore della Magistratura, 2024, pp. 107–121. URL: https://hdl.handle.net/11585/1002472.

[118] Octavian-Eugen Ganea and Thomas Hofmann. "Deep Joint Entity Disambiguation with Local Neural Attention". In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Copenhagen, Denmark: Association for Computational Linguistics, Sept. 2017, pp. 2619–2629. DOI: 10.18653/v1/D17-1277. URL: https://aclanthology.org/D17-1277/.

[119] Leo Gao et al. *The Pile: An 800GB Dataset of Diverse Text for Language Modeling*. 2020. arXiv: 2101.00027 [cs.CL]. URL: https://arxiv.org/abs/2101.00027.

[120] Silin Gao et al. "Efficient Tool Use with Chain-of-Abstraction Reasoning". In: Abu Dhabi, UAE: Association for Computational Linguistics, Jan. 2025, pp. 2727–2743. URL: https://aclanthology.org/2025.coling-main.185/.

[121] Yunfan Gao et al. *Retrieval-Augmented Generation for Large Language Models: A Survey*. arXiv preprint arXiv:2312.10997. 2024. arXiv: 2312.10997 [cs.CL].

[122] GateNLP Project. *Python GateNLP*. Accessed: 2025-10-16. 2025. URL: https://gatenlp.github.io/python-gatenlp/.

[123] Saibo Geng, Martin Josifoski, Maxime Peyrard, and Robert West. "Grammar-Constrained Decoding for Structured NLP Tasks without Finetuning". In: Singapore: Association for Computational Linguistics, Dec. 2023, pp. 10932–10952. DOI: 10.18653/v1/2023.emnlp-main.674.

[124] Daniel Gillick et al. "Learning Dense Representations for Entity Retrieval". In: *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 528–537. DOI: 10.18653/v1/K19-1049. URL: https://aclanthology.org/K19-1049/.

[125] Mireia Artigot Golobardes et al. *Artificial Intelligence, Judicial Decision-Making and Fundamental Rights*. Scuola Superiore Magistratura, 2024. URL: https://www.scuolamagistratura.it/documents/20126/1750902/JuLIA_handbook%20Justice_final.pdf.

[126] Google. *How Google's Knowledge Graph works*. 2025. URL: https://support.google.com/knowledgepanel/answer/9787176 (visited on 08/20/2025).

[127] Google LLC. *Explore – Google Trends*. 2025. URL: https://trends.google.com/trends/explore?date=2025-09-25%202025-10-25&geo=US (visited on 10/25/2025).

[128] Aaron Grattafiori et al. *The Llama 3 Herd of Models*. 2024. arXiv: 2407.21783 [cs.AI]. URL: https://arxiv.org/abs/2407.21783.

[129] Cyril Grouin, Sophie Rosset, Pierre Zweigenbaum, Karën Fort, Olivier Galibert, and Ludovic Quintard. "Proposal for an Extension of Traditional Named Entities: From Guidelines to Evaluation, an Overview". In: *Proceedings of the 5th Linguistic Annotation Workshop*. Portland, Oregon, USA: Association for Computational Linguistics, June 2011, pp. 92–100. URL: https://aclanthology.org/W11-0411/.

[130] Albert Gu and Tri Dao. "Mamba: Linear-Time Sequence Modeling with Selective State Spaces". In: *First Conference on Language Modeling*. 2024. URL: https://openreview.net/forum?id=tEYskw1VY2.

[131] Ziwei Gu et al. "AbstractExplorer: Leveraging Structure-Mapping Theory to Enhance Comparative Close Reading at Scale". In: *Proceedings of the 38th Annual ACM Symposium on User Interface Software and Technology*. UIST '25. Association for Computing Machinery, 2025. ISBN: 9798400720376. DOI: 10.1145/3746059.3747773. URL: https://doi.org/10.1145/3746059.3747773.

[132] Jiarui Guan et al. "Designing Human-AI System for Legal Research: A Case Study of Precedent Search in Chinese Law". In: *Proceedings of the Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*. CHI EA '25. Association for Computing Machinery, 2025. ISBN: 9798400713958. DOI: 10.1145/3706599.3720167. URL: https://doi.org/10.1145/3706599.3720167.

[133] Nitish Gupta, Sameer Singh, and Dan Roth. "Entity Linking via Joint Encoding of Types, Descriptions, and Context". In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Copenhagen, Denmark: Association for Computational Linguistics, Sept. 2017, pp. 2681–2690. DOI: 10.18653/v1/D17-1284. URL: https://aclanthology.org/D17-1284/.

[134] Wenjuan Han et al. "LegalAsst: Human-centered and AI-empowered machine to enhance court productivity and legal assistance". In: *Information Sciences* 679 (2024), p. 121052. ISSN: 0020-0255. DOI: https://doi.org/10.1016/j.ins.2024.121052. URL: https://www.sciencedirect.com/science/article/pii/S0020025524009666.

[135] Daniel Hanisch, Katrin Fundel, Heinz-Theodor Mevissen, Ralf Zimmer, and Juliane Fluck. "ProMiner: rule-based protein and gene entity recognition". In: *BMC bioinformatics* 6.Suppl 1 (2005), S14. DOI: 10.1186/1471-2105-6-S1-S14.

[136] Zellig S. Harris. "Distributional Structure". In: *WORD* 10.2-3 (1954), pp. 146–162. DOI: 10.1080/00437956.1954.11659520.

[137] Faegheh Hasibi, Krisztian Balog, and Svein Erik Bratsberg. "Entity Linking in Queries: Tasks and Evaluation". In: ICTIR '15. Northampton, Massachusetts, USA: Association for Computing Machinery, 2015, pp. 171–180. ISBN: 9781450338332. DOI: 10.1145/2808194.2809473. URL: https://doi.org/10.1145/2808194.2809473.

[138] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. 2nd. 12th printing with corrections, Jan 2017. New York, NY: Springer, 2009. DOI: https://doi.org/10.1007/978-0-387-84858-7. URL: https://www.springer.com/gp/book/9780387848570.

[139] Andrew F. Hayes and Klaus Krippendorff. "Answering the Call for a Standard Reliability Measure for Coding Data". In: *Communication Methods and Measures* 1.1 (2007), pp. 77–89. DOI: 10.1080/19312450709336664. URL: https://www.asc.upenn.edu/sites/default/files/2021-03/Answering%20the%20Call%20for%20a%20Standard%20Reliability%20Measure%20for%20Coding%20Data.pdf.

[140] Pengcheng He, Jianfeng Gao, and Weizhu Chen. "DeBERTaV3: Improving DeBERTa using ELECTRA-Style Pre-Training with Gradient-Disentangled Embedding Sharing". In: *ICLR 2023*. Poster. 2023. URL: https://openreview.net/forum?id=sE7-XhLxHA.

[141] Marti A. Hearst. "Clustering versus faceted categories for information exploration". In: *Commun. ACM* 49.4 (Apr. 2006), pp. 59–61. ISSN: 0001-0782. DOI: 10.1145/1121949.1121983. URL: https://doi.org/10.1145/1121949.1121983.

[142] Nicolas Heist and Heiko Paulheim. "NASTyLinker: NIL-Aware Scalable Transformer-Based Entity Linker". In: *The Semantic Web*. Springer Nature Switzerland, 2023. ISBN: 978-3-031-33455-9. DOI: 10.1007/978-3-031-33455-9_11.

[143] William R. Hersh, Chris Buckley, Tony J. Leone, and David Hickam. "TREC 2003 Genomics Track Overview". In: *Text Retrieval Conference (TREC)*. 2003. URL: https://trec.nist.gov/pubs/trec12/papers/GENOMICS.OVERVIEW3.pdf.

[144] L Hirschman. "The Evolution of evaluation: Lessons from the Message Understanding Conferences". In: *Computer Speech & Language* 12.4 (1998), pp. 281–305. ISSN: 0885-2308. DOI: https://doi.org/10.1006/csla.1998.0102. URL: https://www.sciencedirect.com/science/article/pii/S0885230898901024.

[145] Emna Hkiri, Souheyl Mallat, and Mounir Zrigui. "Arabic-English Text Translation Leveraging Hybrid NER". In: The National University (Phillippines), Nov. 2017, pp. 124–131. URL: https://aclanthology.org/Y17-1019/.

[146] Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. "Constructing A Multi-hop QA Dataset for Comprehensive Evaluation of Reasoning Steps". In: *Proceedings of the 28th International Conference on Computational Linguistics*. International Committee on Computational Linguistics, 2020, pp. 6609–6625. DOI: 10.18653/v1/2020.coling-main.580.

[147] Johannes Hoffart, Yasemin Altun, and Gerhard Weikum. "Discovering emerging entities with ambiguous names". In: *Proceedings of the 23rd International Conference on World Wide Web*. WWW '14. Seoul, Korea: Association for Computing Machinery, 2014, pp. 385–396. ISBN: 9781450327442. DOI: 10.1145/2566486.2568003. URL: https://doi.org/10.1145/2566486.2568003.

[148] Johannes Hoffart, Stephan Seufert, Dat Ba Nguyen, Martin Theobald, and Gerhard Weikum. "KORE: keyphrase overlap relatedness for entity disambiguation". In: *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*. CIKM '12. Maui, Hawaii, USA: Association for Computing Machinery, 2012, pp. 545–554. ISBN: 9781450311564. DOI: 10.1145/2396761.2396832. URL: https://doi.org/10.1145/2396761.2396832.

[149] Johannes Hoffart, Fabian M. Suchanek, Klaus Berberich, and Gerhard Weikum. "YAGO2: A spatially and temporally enhanced knowledge base from Wikipedia". In: *Artificial Intelligence* 194 (2013). Artificial Intelligence, Wikipedia and Semi-Structured Resources, pp. 28–61. ISSN: 0004-3702. DOI: https://doi.org/10.1016/j.artint.2012.06.001. URL: https://www.sciencedirect.com/science/article/pii/S0004370212000719.

[150] Hongkang Yang Hongkang Yang et al. "Memory^3: Language Modeling with Explicit Memory". In: *Journal of Machine Learning* 3.3 (Jan. 2024), pp. 300–346. ISSN: 2790-203X. DOI: 10.4208/jml.240708. URL: http://dx.doi.org/10.4208/jml.240708.

[151] Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. "spaCy: Industrial-strength Natural Language Processing in Python". In: (2020). If you use spaCy, please cite it as below. DOI: 10.5281/zenodo.1212303.

[152] Christoph Hoppe, David Pelkmann, Nico Migenda, Daniel Hötte, and Wolfram Schenck. "Towards Intelligent Legal Advisors for Document Retrieval and Question-Answering in German Legal Documents". In: *2021 IEEE Fourth International Conference on Artificial Intelligence and Knowledge Engineering (AIKE)*. 2021, pp. 29–32. DOI: 10.1109/AIKE52691.2021.00011.

[153] George Hripcsak and Adam S. Rothschild. "Agreement, the F-Measure, and Reliability in Information Retrieval". In: *Journal of the American Medical Informatics Association* 12.3 (2005), pp. 296–298. ISSN: 1067-5027. DOI: https://doi.org/10.1197/jamia.M1733. URL: https://www.sciencedirect.com/science/article/pii/S1067502705000253.

[154] Xiaoguang Hu, Wei Li, Xiang Lan, Hua Wu, and Haifeng Wang. "Improved beam search with constrained softmax for NMT". In: *Proceedings of Machine Translation Summit XV: Papers*. Miami, USA, 2015. URL: https://aclanthology.org/2015.mtsummit-papers.23/.

[155] Oz Huly, Idan Pogrebinsky, David Carmel, Oren Kurland, and Yoelle Maarek. "Old IR Methods Meet RAG". In: *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR '24. Washington DC, USA: Association for Computing Machinery, 2024, pp. 2559–2563. ISBN: 9798400704314. DOI: 10.1145/3626772.3657935. URL: https://doi.org/10.1145/3626772.3657935.

[156] Samuel Humeau, Kurt Shuster, Marie-Anne Lachaux, and Jason Weston. "Poly-encoders: Architectures and Pretraining Strategies for Fast and Accurate Multi-sentence Scoring". In: *International Conference on Learning Representations*. 2020. URL: https://openreview.net/forum?id=SkxgnnNFvH.

[157] Filip Ilievski, Eduard Hovy, Piek Vossen, Stefan Schlobach, and Qizhe Xie. "The role of knowledge in determining identity of long-tail entities". In: *Journal of Web Semantics* 61-62 (2020), p. 100565. ISSN: 1570-8268. DOI: https://doi.org/10.1016/j.websem.2020.100565.

[158] Italian Republic. *Intelligenza artificiale (Artificial Intelligence)*. Law n. 1146/2024. 2025. URL: https://www.senato.it/leggi-e-documenti/disegni-di-legge/scheda-ddl?tab=testiEmendamenti&did=59313.

[159] Anastasiia Iurshina, Jiaxin Pan, Rafika Boutalbi, and Steffen Staab. "NILK: Entity Linking Dataset Targeting NIL-linking Cases". In: *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. CIKM '22. Atlanta, GA, USA: Association for Computing Machinery, 2022, pp. 4069–4073. ISBN: 9781450392365. DOI: 10.1145/3511808.3557659. URL: https://doi.org/10.1145/3511808.3557659.

[160] Gautier Izacard and Edouard Grave. "Leveraging Passage Retrieval with Generative Models for Open Domain Question Answering". In: *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. Online: Association for Computational Linguistics, Apr. 2021, pp. 874–880. DOI: 10.18653/v1/2021.eacl-main.74. URL: https://aclanthology.org/2021.eacl-main.74/.

[161] Paul Jaccard. *Etude comparative de la distribution florale dans une portion des Alpes et du Jura*. Vol. 37. Bulletin de la Société vaudoise des sciences naturelles. Impr. Corbaz, 1901, pp. 547–579. URL: https://books.google.it/books?id=JCNdmgEACAAJ.

[162] Nihal Jain et al. "ContraCLM: Contrastive Learning For Causal Language Model". In: Toronto, Canada: Association for Computational Linguistics, July 2023, pp. 6436–6459. DOI: 10.18653/v1/2023.acl-long.355. URL: https://aclanthology.org/2023.acl-long.355/.

[163] Aaron Jarrett and Kim-Kwang Raymond Choo. "The impact of automation and artificial intelligence on digital forensics". In: *Wiley Interdisciplinary Reviews: Forensic Science* 3.6 (2021), e1418. DOI: 10.1002/wfs2.1418.

[164] Kunal Jha, Michael Röder, and Axel-Cyrille Ngonga Ngomo. "All that glitters is not gold–rule-based curation of reference datasets for named entity recognition and entity linking". In: *The Semantic Web: 14th International Conference, ESWC 2017, Portoro, Slovenia, May 28–June 1, 2017, Proceedings, Part I 14*. Springer. 2017. DOI: 10.1007/978-3-319-58068-5_19.

[165] Heng Ji and Ralph Grishman. "Overview of the TAC2011 Knowledge Base Population Track". In: *Proceedings of the Fourth Text Analysis Conference (TAC 2011)*. NIST, 2011. URL: https://blender.cs.illinois.edu/paper/kbp2011.pdf.

[166] Heng Ji, Ralph Grishman, Hoa Trang Dang, Kira Griffitt, and Joe Ellis. "Overview of the TAC 2010 knowledge base population track". In: *Third text analysis conference (TAC 2010)*. Vol. 3. 2. 2010, pp. 3–3. URL: https://blender.cs.illinois.edu/paper/kbp2010overview.pdf.

[167] Ziwei Ji et al. "Survey of Hallucination in Natural Language Generation". In: *ACM Comput. Surv.* 55.12 (Mar. 2023). ISSN: 0360-0300. DOI: 10.1145/3571730. URL: https://doi.org/10.1145/3571730.

[168] Shanshan Jiang, Yihan Li, Tianyi Qin, Qian Meng, and Bin Dong. "SRCB Entity Discovery and Linking (EDL) and Event Nugget Systems for TAC 2017". In: (2017). URL: https://tac.nist.gov/publications/2017/participant.papers/TAC2017.srcb.proceedings.pdf.

[169] Jeff Johnson, Matthijs Douze, and Hervé Jégou. "Billion-Scale Similarity Search with GPUs". In: *IEEE Transactions on Big Data* 7.3 (2021), pp. 535–547. DOI: 10.1109/TBDATA.2019.2921572.

[170] Michiel de Jong, Yury Zemlyanskiy, Nicholas FitzGerald, Fei Sha, and William W. Cohen. "Mention Memory: incorporating textual knowledge into Transformers through entity mention attention". In: *Proceedings of the International Conference on Learning Representations (ICLR)*. 2022. URL: https://openreview.net/pdf?id=OY1A8ejQgEX.

[171] *Judgments of the Court of Justice in Case C-634/21 | SCHUFA Holding (Scoring) and in Joined Cases C-26/22 and C-64/22 | SCHUFA Holding (Discharge from remaining debts)*. 2023. URL: https://curia.europa.eu/jcms/upload/docs/application/pdf/2023-12/cp230186en.pdf (visited on 09/19/2025).

[172] Daniel Jurafsky and James H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition, with Language Models*. 3rd. Online manuscript released August 24, 2025. 2025. URL: https://web.stanford.edu/~jurafsky/slp3/.

[173] Rijula Kar, Susmija Reddy, Sourangshu Bhattacharya, Anirban Dasgupta, and Soumen Chakrabarti. "Task-specific representation learning for web-scale entity disambiguation". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 32. 1. 2018. DOI: 10.1609/aaai.v32i1.12066.

[174] Vladimir Karpukhin et al. "Dense Passage Retrieval for Open-Domain Question Answering". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, Nov. 2020, pp. 6769–6781. DOI: 10.18653/v1/2020.emnlp-main.550. URL: https://aclanthology.org/2020.emnlp-main.550/.

[175] Nora Kassner, Fabio Petroni, Mikhail Plekhanov, Sebastian Riedel, and Nicola Cancedda. "EDIN: An End-to-end Benchmark and Pipeline for Unknown Entity Discovery and Indexing". In: *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022, pp. 8659–8673. DOI: 10.18653/v1/2022.emnlp-main.593. URL: https://aclanthology.org/2022.emnlp-main.593/.

[176] Keshav Kaushik and Yash Katara. "Forensic Analysis of WhatsApp chat data". In: *2022 10th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)(ICRITO)*. IEEE. 2022, pp. 1–6. DOI: 10.1109/ICRITO56286.2022.9965028. URL: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9965028.

[177] Mayank Kejriwal and Pedro Szekely. "An Investigative Search Engine for the Human Trafficking Domain". In: *The Semantic Web – ISWC 2017*. Cham: Springer International Publishing, 2017, pp. 247–262. ISBN: 978-3-319-68204-4. DOI: 10.1007/978-3-319-68204-4_25.

[178] Mayank Kejriwal, Pedro Szekely, and Craig Knoblock. "Investigative Knowledge Discovery for Combating Illicit Activities". In: *IEEE Intelligent Systems* 33 (Jan. 2018), pp. 53–63. DOI: 10.1109/MIS.2018.111144556.

[179] Imed Keraghel, Stanislas Morbieu, and Mohamed Nadif. *Recent Advances in Named Entity Recognition: A Comprehensive Survey and Comparative Study*. 2024. arXiv: 2401.10825 [cs.CL]. URL: https://arxiv.org/abs/2401.10825.

[180] Hossein Keshavarz et al. "Named Entity Recognition in Long Documents: An End-to-end Case Study in the Legal Domain". In: *2022 IEEE International Conference on Big Data (Big Data)*. 2022, pp. 2024–2033. DOI: 10.1109/BigData55660.2022.10020873.

[181] Daniel Keszthelyi, Christophe Gaudet-Blavignac, Mina Bjelogrlic, and Christian Lovis. "Patient Information Summarization in Clinical Settings: Scoping Review". In: *JMIR Med Inform* 11 (Nov. 2023), e44639. ISSN: 2291-9694. DOI: 10.2196/44639. URL: http://www.ncbi.nlm.nih.gov/pubmed/38015588.

[182] Mohammad Ebrahim Khademi and Mohammad Fakhredanesh. "Persian automatic text summarization based on named entity recognition". In: *Iranian Journal of Science and Technology, Transactions of Electrical Engineering* (2020), pp. 1–12. DOI: 10.1007/s40998-020-00352-2.

[183] Sammy Khalife and Michalis Vazirgiannis. "Scalable graph-based method for individual named entity identification". In: *Proceedings of the Thirteenth Workshop on Graph-Based Methods for Natural Language Processing (TextGraphs-13)*. Hong Kong: Association for Computational Linguistics, Nov. 2019, pp. 17–25. DOI: 10.18653/v1/D19-5303. URL: https://aclanthology.org/D19-5303/.

[184] Jiho Kim, Yeonsu Kwon, Yohan Jo, and Edward Choi. "KG-GPT: A General Framework for Reasoning on Knowledge Graphs Using Large Language Models". In: *Findings of the Association for Computational Linguistics: EMNLP 2023*. Singapore: Association for Computational Linguistics, Dec. 2023, pp. 9410–9421. DOI: 10.18653/v1/2023.findings-emnlp.631. URL: https://aclanthology.org/2023.findings-emnlp.631/.

[185] Jan-Christoph Klie, Richard Eckart de Castilho, and Iryna Gurevych. "From Zero to Hero: Human-In-The-Loop Entity Linking in Low Resource Domains". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, July 2020, pp. 6982–6993. DOI: 10.18653/v1/2020.acl-main.624. URL: https://aclanthology.org/2020.acl-main.624/.

[186] Graham Klyne. *Resource Description Framework (RDF): Concepts and Abstract Syntax*. W3C Recommendation. 2004. URL: http://www.w3.org/TR/rdf-concepts/.

[187] Nikolaos Kolitsas, Octavian-Eugen Ganea, and Thomas Hofmann. "End-to-End Neural Entity Linking". In: *Proceedings of the 22nd Conference on Computational Natural Language Learning*. Brussels, Belgium: Association for Computational Linguistics, Oct. 2018, pp. 519–529. DOI: 10.18653/v1/K18-1050. URL: https://aclanthology.org/K18-1050/.

[188] John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data". In: *Proceedings of the Eighteenth International Conference on Machine Learning*. ICML '01. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2001, pp. 282–289. ISBN: 1558607781. URL: http://www.cs.columbia.edu/~jebara/6772/papers/crf.pdf.

[189] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. "Neural Architectures for Named Entity Recognition". In: San Diego, California: Association for Computational Linguistics, June 2016, pp. 260–270. DOI: 10.18653/v1/N16-1030. URL: https://aclanthology.org/N16-1030/.

[190] Yunshi Lan, Gaole He, Jinhao Jiang, Jing Jiang, Wayne Xin Zhao, and Ji-Rong Wen. "Complex knowledge base question answering: A survey". In: *IEEE Transactions on Knowledge and Data Engineering* 35.11 (2022), pp. 11196–11215. DOI: 10.1109/TKDE.2022.3223858. URL: https://ieeexplore.ieee.org/document/9960856.

[191] Snezana Lazovic. "Perspectives on AI, Machine Learning and Predictability in the Italian Legal System. Data Set Management, Accuracy and Reliability of the "Predictive Justice" Project". PhD in Social Sciences and Humanities, Curriculum: Law, Psychology and Education, Cycle XXXV. Niccolò Cusano University - Telematic, Rome, 2022. URL: https://tesidottorato.depositolegale.it/bitstream/20.500.14242/192401/2/DOTTORATO%20LAZOVIC%20SNEZANA.pdf.

[192] Phong Le and Ivan Titov. "Boosting Entity Linking Performance by Leveraging Unlabeled Documents". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, July 2019, pp. 1935–1945. DOI: 10.18653/v1/P19-1187. URL: https://aclanthology.org/P19-1187/.

[193] Minh Lê and Antske Fokkens. "Tackling Error Propagation through Reinforcement Learning: A Case of Greedy Dependency Parsing". In: *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*. 2017, pp. 677–687.

[194] *Legge 10 aprile 1951, n. 287, Riordinamento dei giudizi di Assise (Reorganization of Assize Trials)*. https://www.normattiva.it/uri-res/N2Ls?urn:nir:legge:1951-04-10;287. 1951.

[195] Jens Lehmann, Dhananjay Bhandiwad, Preetam Gattogi, and Sahar Vahdati. "Beyond Boundaries: A Human-like Approach for Question Answering over Structured and Unstructured Information Sources". In: 12 (June 2024), pp. 786–802. ISSN: 2307-387X. DOI: 10.1162/tacl_a_00671. eprint: https://direct.mit.edu/tacl/article-pdf/doi/10.1162/tacl_a_00671/2383373/tacl_a_00671.pdf. URL: https://doi.org/10.1162/tacl%5C_a%5C_00671.

[196] Jens Lehmann et al. "Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia". In: *Semantic web* 6.2 (2015), pp. 167–195. DOI: 10.3233/SW-140134.

[197] Vladimir I. Levenshtein. "Binary codes capable of correcting deletions, insertions and reversals". In: *Soviet Physics Doklady* 10 (1966).

[198] Patrick Lewis et al. "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks". In: *Advances in Neural Information Processing Systems*. Vol. 33. Curran Associates, Inc., 2020, pp. 9459–9474. URL: https://proceedings.neurips.cc/paper_files/paper/2020/file/6b493230205f780e1bc26945df7481e5-Paper.pdf.

[199] Belinda Z. Li, Sewon Min, Srinivasan Iyer, Yashar Mehdad, and Wen-tau Yih. "Efficient One-Pass End-to-End Entity Linking for Questions". In: Online: Association for Computational Linguistics, Nov. 2020, pp. 6433–6441. DOI: 10.18653/v1/2020.emnlp-main.522. URL: https://aclanthology.org/2020.emnlp-main.522/.

[200] Jonathan Li, Rohan Bhambhoria, and Xiaodan Zhu. "Parameter-Efficient Legal Domain Adaptation". In: Abu Dhabi, United Arab Emirates (Hybrid): Association for Computational Linguistics, Dec. 2022, pp. 119–129. DOI: 10.18653/v1/2022.nllp-1.10. URL: https://aclanthology.org/2022.nllp-1.10/.

[201] Kun Li, Tianhua Zhang, Xixin Wu, Hongyin Luo, James R. Glass, and Helen M. Meng. "Decoding on Graphs: Faithful and Sound Reasoning on Knowledge Graphs through Generation of Well-Formed Chains". In: Vienna, Austria: Association for Computational Linguistics, July 2025, pp. 24349–24364. ISBN: 979-8-89176-251-0. DOI: 10.18653/v1/2025.acl-long.1186. URL: https://aclanthology.org/2025.acl-long.1186/.

[202] Peng Li et al. "CodeIE: Large Code Generation Models are Better Few-Shot Information Extractors". In: Toronto, Canada: Association for Computational Linguistics, July 2023, pp. 15339–15353. DOI: 10.18653/v1/2023.acl-long.855. URL: https://aclanthology.org/2023.acl-long.855/.

[203] Zhenzhen Li, Qun Zhang, Ting Li, Jun Xu, and Dawei Feng. "A Hybrid Model for Trilingual Entity Detection and Linking Tasks at TAC KBP 2017". In: *Proceedings of the 10th Text Analysis Conference (TAC 2017)*. NIST, 2017. URL: https://tac.nist.gov/publications/2017/participant.papers/TAC2017.NUDT_PDL2017.proceedings.pdf.

[204] Zhoubo Li, Ningyu Zhang, Yunzhi Yao, Mengru Wang, Xi Chen, and Huajun Chen. "Unveiling the Pitfalls of Knowledge Editing for Large Language Models". In: *The Twelfth International Conference on Learning Representations*. 2024. URL: https://openreview.net/forum?id=fNktD3ib16.

[205] Zixuan Li et al. "The Open Knowledge System for TAC KBP 2017". In: *Proceedings of the 10th Text Analysis Conference (TAC 2017)*. NIST, 2017. URL: https://tac.nist.gov/publications/2017/participant.papers/TAC2017.ICTCAS_OKN.proceedings.pdf.

[206] Daniele Licari and Giovanni Comandè. "ITALIAN-LEGAL-BERT: A Pre-trained Transformer Language Model for Italian Law". In: *Companion Proceedings of the 23rd International Conference on Knowledge Engineering and Knowledge Management*. Vol. 3256. CEUR Workshop Proceedings. Bozen-Bolzano, Italy: CEUR, Sept. 2022. URL: https://ceur-ws.org/Vol-3256/#km4law3.

[207] Peerat Limkonchotiwat, Weiwei Cheng, Christos Christodoulopoulos, Amir Saffari, and Jens Lehmann. "mReFinED: An Efficient End-to-End Multilingual Entity Linking System". In: *Findings of the Association for Computational Linguistics: EMNLP 2023*. Singapore: Association for Computational Linguistics, Dec. 2023, pp. 15080–15089. DOI: 10.18653/v1/2023.findings-emnlp.1007. URL: https://aclanthology.org/2023.findings-emnlp.1007/.

[208] Jessica Lin and Amir Zeldes. "GUMsley: Evaluating Entity Salience in Summarization for 12 English Genres". In: St. Julian's, Malta: Association for Computational Linguistics, Mar. 2024, pp. 2575–2588. DOI: 10.18653/v1/2024.eacl-long.158. URL: https://aclanthology.org/2024.eacl-long.158/.

[209] Xukai Liu, Ye Liu, Kai Zhang, Kehang Wang, Qi Liu, and Enhong Chen. "OneNet: A Fine-Tuning Free Framework for Few-Shot Entity Linking via Large Language Model Prompting". In: *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. Miami, Florida, USA: Association for Computational Linguistics, Nov. 2024, pp. 13634–13651. DOI: 10.18653/v1/2024.emnlp-main.756. URL: https://aclanthology.org/2024.emnlp-main.756/.

[210] Yinhan Liu et al. *Ro{BERT}a: A Robustly Optimized {BERT} Pretraining Approach*. 2020. URL: https://openreview.net/forum?id=SyxS0T4tvS.

[211] Robert L Logan IV, Andrew McCallum, Sameer Singh, and Dan Bikel. "Benchmarking Scalable Methods for Streaming Cross Document Entity Coreference". In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Online: Association for Computational Linguistics, Aug. 2021, pp. 4717–4731. DOI: 10.18653/v1/2021.acl-long.364. URL: https://aclanthology.org/2021.acl-long.364/.

[212] Lajanugen Logeswaran, Ming-Wei Chang, Kenton Lee, Kristina Toutanova, Jacob Devlin, and Honglak Lee. "Zero-Shot Entity Linking by Reading Entity Descriptions". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, July 2019, pp. 3449–3460. DOI: 10.18653/v1/P19-1335. URL: https://aclanthology.org/P19-1335/.

[213] Ilya Loshchilov and Frank Hutter. "Decoupled Weight Decay Regularization". In: *International Conference on Learning Representations*. 2019. URL: https://openreview.net/forum?id=Bkg6RiCqY7.

[214] Ling Luo, Chih-Hsuan Wei, Po-Ting Lai, Robert Leaman, Qingyu Chen, and Zhiyong Lu. "AIONER: all-in-one scheme-based biomedical named entity recognition using deep learning". In: *Bioinformatics* 39.5 (May 2023). ISSN: 1367-4811. DOI: 10.1093/bioinformatics/btad310. eprint: https://academic.oup.com/bioinformatics/article-pdf/39/5/btad310/50453589/btad310.pdf. URL: https://doi.org/10.1093/bioinformatics/btad310.

[215] Linhao Luo, Zicheng Zhao, Chen Gong, Gholamreza Haffari, and Shirui Pan. "Graph-constrained Reasoning: Faithful Reasoning on Knowledge Graphs with Large Language Models". In: *Forty-second International Conference on Machine Learning*. 2025. arXiv: 2410.13080 [cs.CL]. URL: https://arxiv.org/abs/2410.13080.

[216] Xiaoqiang Luo. "On Coreference Resolution Performance Metrics". In: *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*. Vancouver, British Columbia, Canada: Association for Computational Linguistics, Oct. 2005, pp. 25–32. URL: https://aclanthology.org/H05-1004/.

[217] Xuezhe Ma and Eduard Hovy. "End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF". In: Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 1064–1074. DOI: 10.18653/v1/P16-1101. URL: https://aclanthology.org/P16-1101/.

[218] B. Magnini et al. "I-CAB: the Italian Content Annotation Bank". In: *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*. Genoa, Italy: European Language Resources Association (ELRA), May 2006. URL: https://aclanthology.org/L06-1313/.

[219] Farzaneh Mahdisoltani, Joanna Biega, and Fabian M. Suchanek. "YAGO3: A Knowledge Base from Multilingual Wikipedias". In: *Seventh Biennial Conference on Innovative Data Systems Research, CIDR 2015, Asilomar, CA, USA, January 4-7, 2015, Online Proceedings*. www.cidrdb.org, 2015. URL: http://cidrdb.org/cidr2015/Papers/CIDR15%5C_Paper1.pdf.

[220] Yu A. Malkov and D. A. Yashunin. "Efficient and Robust Approximate Nearest Neighbor Search Using Hierarchical Navigable Small World Graphs". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 42.4 (Apr. 2020), pp. 824–836. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2018.2889473. URL: https://doi.org/10.1109/TPAMI.2018.2889473.

[221] Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. "The Stanford CoreNLP Natural Language Processing Toolkit". In: *Association for Computational Linguistics (ACL) System Demonstrations*. 2014, pp. 55–60. URL: http://www.aclweb.org/anthology/P/P14/P14-5010.

[222] James Martin. *Managing the Data-Base Environment*. Englewood Cliffs, New Jersey: Prentice-Hall, 1983, p. 381. ISBN: 0135505828.

[223] Pedro Henrique Martins, Zita Marinho, and André F. T. Martins. "Joint Learning of Named Entity Recognition and Entity Linking". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*. Florence, Italy: Association for Computational Linguistics, July 2019. DOI: 10.18653/v1/P19-2026. URL: https://aclanthology.org/P19-2026.

[224] Mattia Marzocchi, Marco Cremaschi, Riccardo Pozzi, Roberto Avogadro, and Matteo Palmonari. "MammoTab: A Giant and Comprehensive Dataset for Semantic Table Interpretation". In: *SemTab@ISWC*. 2022, pp. 28–33. URL: https://ceur-ws.org/Vol-3320/paper3.pdf.

[225] Costas Mavromatis and George Karypis. "GNN-RAG: Graph Neural Retrieval for Efficient Large Language Model Reasoning on Knowledge Graphs". In: *Findings of the Association for Computational Linguistics: ACL 2025*. Vienna, Austria: Association for Computational Linguistics, July 2025, pp. 16682–16699. ISBN: 979-8-89176-256-5. DOI: 10.18653/v1/2025.findings-acl.856. URL: https://aclanthology.org/2025.findings-acl.856/.

[226] Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. "On Faithfulness and Factuality in Abstractive Summarization". In: 2020, pp. 1906–1919. DOI: 10.18653/v1/2020.acl-main.173.

[227] Paul McNamee and Hoa Trang Dang. "Overview of the TAC 2009 knowledge base population track". In: *Proceedings of the Second Text Analysis Conference (TAC 2009)*. NIST, 2009. URL: https://tac.nist.gov/publications/2009/presentations/TAC2009_KBP_overview.pdf.

[228] Pablo N. Mendes, Max Jakob, Andrés Garca-Silva, and Christian Bizer. "DBpedia spotlight: shedding light on the web of documents". In: *Proceedings of the 7th International Conference on Semantic Systems*. I-Semantics '11. Graz, Austria: Association for Computing Machinery, 2011, pp. 1–8. ISBN: 9781450306218. DOI: 10.1145/2063518.2063519. URL: https://www.dbpedia-spotlight.org/docs/spotlight.pdf.

[229] Xupeng Miao et al. "X-former Elucidator: Reviving Efficient Attention for Long Context Language Modeling". In: *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*. Survey Track. International Joint Conferences on Artificial Intelligence Organization, Aug. 2024, pp. 8179–8187. DOI: 10.24963/ijcai.2024/904. URL: https://doi.org/10.24963/ijcai.2024/904.

[230] Tomas Mikolov, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. "Efficient Estimation of Word Representations in Vector Space". In: 2013. DOI: https://doi.org/10.48550/arXiv.1301.3781.

[231] Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. "Linguistic Regularities in Continuous Space Word Representations". In: Atlanta, Georgia: Association for Computational Linguistics, June 2013, pp. 746–751. URL: https://aclanthology.org/N13-1090/.

[232] Dan Milmo and Agency. *Two US lawyers fined for submitting fake court citations from ChatGPT*. June 2023. URL: https://www.theguardian.com/technology/2023/jun/23/two-us-lawyers-fined-submitting-fake-court-citations-chatgpt (visited on 02/24/2025).

[233] Marie-Francine Moens. "Innovative techniques for legal text retrieval". In: *Artificial Intelligence and Law* 9.1 (2001), pp. 29–57. DOI: 10.1023/A:1011297104922.

[234] Diego Mollá, Menno van Zaanen, and Daniel Smith. "Named Entity Recognition for Question Answering". In: *Proceedings of the Australasian Language Technology Workshop 2006*. Sydney, Australia, Nov. 2006, pp. 51–58. URL: https://aclanthology.org/U06-1009/.

[235] Cedric Möller and Ricardo Usbeck. "Entity Linking with Out-of-Knowledge-Graph Entity Detection and Clustering Using Only Knowledge Graphs". In: *Knowledge Graphs in the Age of Language Models and Neuro-Symbolic AI - Proceedings of the 20th International Conference on Semantic Systems, 17-19 September 2024, Amsterdam, The Netherlands*. Vol. 60. Studies on the Semantic Web. IOS Press, 2024, pp. 88–105. DOI: 10.3233/SSW240009.

[236] Sean Monahan, John Lehmann, Timothy Nyberg, Jesse Plymale, and Arnold Jung. "Cross-Lingual Cross-Document Coreference with Entity Linking". In: *Proceedings of the Fourth Text Analysis Conference (TAC 2011)*. NIST, 2011. URL: https://tac.nist.gov/publications/2011/participant.papers/lcc.proceedings.pdf.

[237] *MongoDB Query Language (MQL) Reference*. MongoDB, Inc. URL: https://www.mongodb.com/docs/manual/reference/mql/ (visited on 10/19/2025).

[238] Jose G. Moreno et al. "Combining Word and Entity Embeddings for Entity Linking". In: *The Semantic Web*. Cham: Springer International Publishing, 2017, pp. 337–352. ISBN: 978-3-319-58068-5.

[239] Julián Moreno-Schneider and Georg Rehm. *Lynx D4.5 Final implementation and report of Data and Content Curation Services*. en. Creative Commons Attribution 4.0 International. 2021. DOI: 10.5281/zenodo.4651358. URL: https://zenodo.org/record/4651358.

[240] Matteo Muffo and Enrico Bertino. "BERTino: An Italian DistilBERT Model". In: *Proceedings of the Seventh Italian Conference on Computational Linguistics (CLiC-it 2020)*. Vol. 2769. CEUR Workshop Proceedings. Online: CEUR-WS.org, 2020. DOI: 10.48550/arXiv.2303.18121. URL: https://ceur-ws.org/Vol-2769/paper_09.pdf.

[241] In Jae Myung. "Tutorial on maximum likelihood estimation". In: *Journal of Mathematical Psychology* 47.1 (2003), pp. 90–100. ISSN: 0022-2496. DOI: https://doi.org/10.1016/S0022-2496(02)00028-7. URL: https://www.sciencedirect.com/science/article/pii/S0022249602000287.

[242] David Nadeau and Satoshi Sekine. "A survey of named entity recognition and classification". In: *Lingvisticae Investigationes* 30.1 (2007). URL: https://nlp.cs.nyu.edu/sekine/papers/li07.pdf.

[243] Hiroki Nakayama, Takahiro Kubo, Junya Kamura, Yasufumi Taniguchi, and Xu Liang. *doccano: Text Annotation Tool for Human*. 2018. URL: https://github.com/doccano/doccano.

[244] Deepak Nathani et al. *MLGym: A New Framework and Benchmark for Advancing AI Research Agents*. 2025. arXiv: 2502.14499 [cs.CL]. URL: https://arxiv.org/abs/2502.14499.

[245] Humza Naveed et al. "A Comprehensive Overview of Large Language Models". In: *ACM Trans. Intell. Syst. Technol.* 16.5 (Aug. 2025). ISSN: 2157-6904. DOI: 10.1145/3744746. URL: https://doi.org/10.1145/3744746.

[246] *Neo4j — Graph Database Platform*. 2025. URL: https://neo4j.com/ (visited on 10/19/2025).

[247] Giang Nguyen, tefan Dlugolinský, Michal Laclavík, Martin eleng, and Viet Tran. "Next Improvement Towards Linear Named Entity Recognition Using Character Gazetteers". In: *Advanced Computational Methods for Knowledge Engineering*. Cham: Springer International Publishing, 2014, pp. 255–265. ISBN: 978-3-319-06569-4.

[248] Thien Huu Nguyen, Nicolas Fauceglia, Mariano Rodriguez Muro, Oktie Hassanzadeh, Alfio Massimiliano Gliozzo, and Mohammad Sadoghi. "Joint Learning of Local and Global Features for Entity Linking via Neural Networks". In: *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*. Osaka, Japan: The COLING 2016 Organizing Committee, Dec. 2016, pp. 2310–2320. URL: https://aclanthology.org/C16-1218/.

[249] Songjie Niu et al. "Tree-KG: An Expandable Knowledge Graph Construction Framework for Knowledge-intensive Domains". In: *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vienna, Austria: Association for Computational Linguistics, July 2025, pp. 18516–18529. ISBN: 979-8-89176-251-0. DOI: 10.18653/v1/2025.acl-long.907. URL: https://aclanthology.org/2025.acl-long.907/.

[250] Joel Nothman, Nicky Ringland, Will Radford, Tara Murphy, and James R. Curran. "Learning multilingual named entity recognition from Wikipedia". In: *Artificial Intelligence* 194 (2013). Artificial Intelligence, Wikipedia and Semi-Structured Resources. ISSN: 0004-3702. DOI: https://doi.org/10.1016/j.artint.2012.03.006.

[251] Yasumasa Onoe and Greg Durrett. "Fine-Grained Entity Typing for Domain Independent Entity Linking". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 34.05 (Apr. 2020), pp. 8576–8583. DOI: 10.1609/aaai.v34i05.6380. URL: https://ojs.aaai.org/index.php/AAAI/article/view/6380.

[252] OpenAI. *ChatGPT*. Large language model. 2022. URL: https://chatgpt.com/ (visited on 10/10/2025).

[253] OpenAI. "GPT-4 Technical Report". In: *CoRR* abs/2303.08774 (2023). DOI: 10.48550/ARXIV.2303.08774. arXiv: 2303.08774. URL: https://doi.org/10.48550/arXiv.2303.08774.

[254] OpenAI. *Introducing ChatGPT Search*. 2024. URL: https://openai.com/index/introducing-chatgpt-search/ (visited on 10/25/2025).

[255] OpenAI et al. *OpenAI o1 System Card*. 2024. arXiv: 2412.16720 [cs.AI]. URL: https://arxiv.org/abs/2412.16720.

[256] Charles E. Osgood, George J. Suci, and Percy H. Tannenbaum. *The Measurement of Meaning*. University of Illinois Press, 1957.

[257] Long Ouyang et al. "Training language models to follow instructions with human feedback". In: *Advances in Neural Information Processing Systems*. Vol. 35. Curran Associates, Inc., 2022, pp. 27730–27744. URL: https://proceedings.neurips.cc/paper_files/paper/2022/file/b1efde53be364a73914f58805a001731-Paper-Conference.pdf.

[258] Teresa Paccosi and Alessio Palmero Aprosio. "KIND: an Italian Multi-Domain Dataset for Named Entity Recognition". In: Marseille, France: European Language Resources Association, June 2022, pp. 501–507. URL: https://aclanthology.org/2022.lrec-1.52/.

[259] Alessio Palmero Aprosio and Giovanni Moretti. "Tint 2.0: an All-inclusive Suite for NLP in Italian". In: *Proceedings of the Fifth Italian Conference on Computational Linguistics (CLiC-it 2018)*. Turin, Italy: CEUR Workshop Proceedings, Dec. 2018, pp. 312–318. ISBN: 978-88-31978-41-5. URL: https://aclanthology.org/2018.clicit-1.55/.

[260] Shirui Pan, Linhao Luo, Yufei Wang, Chen Chen, Jiapu Wang, and Xindong Wu. "Unifying Large Language Models and Knowledge Graphs: A Roadmap". In: *IEEE Transactions on Knowledge and Data Engineering* 36.7 (2024), pp. 3580–3599. DOI: 10.1109/TKDE.2024.3352100.

[261] David Patterson et al. "The Carbon Footprint of Machine Learning Training Will Plateau, Then Shrink". In: *Computer* 55.7 (2022), pp. 18–28. DOI: 10.1109/MC.2022.3148714.

[262] Thomas Pellissier Tanon, Denny Vrandei, Sebastian Schaffert, Thomas Steiner, and Lydia Pintscher. "From Freebase to Wikidata: The Great Migration". In: *Proceedings of the 25th International Conference on World Wide Web*. WWW '16. Montréal, Québec, Canada: International World Wide Web Conferences Steering Committee, 2016, pp. 1419–1428. ISBN: 9781450341431. DOI: 10.1145/2872427.2874809.

[263] Thomas Pellissier Tanon, Gerhard Weikum, and Fabian Suchanek. "YAGO 4: A Reason-able Knowledge Base". In: *The Semantic Web*. Cham: Springer International Publishing, 2020, pp. 583–596. ISBN: 978-3-030-49461-2. DOI: 10.1007/978-3-030-49461-2_34.

[264] Boci Peng et al. "Graph Retrieval-Augmented Generation: A Survey". In: *ACM Trans. Inf. Syst.* 44.2 (Dec. 2025). ISSN: 1046-8188. DOI: 10.1145/3777378. URL: https://doi.org/10.1145/3777378.

[265] Francisco J. Pérez et al. "Multimedia analysis platform for crime prevention and investigation". In: *Multimedia Tools and Applications* (Feb. 2021). ISSN: 1573-7721. DOI: 10.1007/s11042-020-10206-y. URL: http://dx.doi.org/10.1007/s11042-020-10206-y.

[266] Fabio Petroni et al. "Language Models as Knowledge Bases?" In: Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 2463–2473. DOI: 10.18653/v1/D19-1250. URL: https://aclanthology.org/D19-1250/.

[267] Shahnaz Pirzada, Nurul Hidayah Ab Rahman, Niken Dwi Wahyu Cahyani, and Muhammad Fakri Othman. "A Framework of Forensic Analysis and Visualization: Using WhatsApp Chat Data as a Case Study". In: *JOIV: International Journal on Informatics Visualization* 8.3-2 (2024), pp. 1834–1848. DOI: 10.62527/joiv.8.3-2.2868. URL: https://www.joiv.org/index.php/joiv/article/viewFile/2868/1103.

[268] Aske Plaat, Max van Duijn, Niki van Stein, Mike Preuss, Peter van der Putten, and Kees Joost Batenburg. *Agentic Large Language Models, a survey*. 2025. arXiv: 2503.23037 [cs.AI]. URL: https://arxiv.org/abs/2503.23037.

[269] Mikhail Plekhanov et al. *Multilingual End to End Entity Linking*. 2023. arXiv: 2306.08896 [cs.CL]. URL: https://arxiv.org/abs/2306.08896.

[270] Thiago Dal Pont, Federico Galli, Andrea Loreggia, Giuseppe Pisano, Riccardo Rovatti, and Giovanni Sartor. *Legal Summarisation through LLMs: The PRODIGIT Project*. 2023. arXiv: 2308.04416 [cs.CL]. URL: https://arxiv.org/abs/2308.04416.

[271] Riccardo Pozzi, Valentina Barbera, Renzo Alva Principe, Davide Giardini, Riccardo Rubini, and Matteo Palmonari. "Combining Knowledge Graphs and NLP to Analyze Instant Messaging Data in Criminal Investigations". In: *Web Information Systems Engineering – WISE 2024*. Springer Nature Singapore, 2025, pp. 427–442. ISBN: 978-981-96-0567-5. DOI: 10.1007/978-981-96-0567-5_30. arXiv: 2509.26487 [cs.AI]. URL: https://link.springer.com/content/pdf/10.1007/978-981-96-0567-5.pdf.

[272] Riccardo Pozzi, Federico Moiraghi, Fausto Lodi, and Matteo Palmonari. "Evaluation of Incremental Entity Extraction with Background Knowledge and Entity Linking". In: *Proceedings of the 11th International Joint Conference on Knowledge Graphs*. IJCKG '22. Hangzhou, China: Association for Computing Machinery, 2023, pp. 30–38. ISBN: 9781450399876. DOI: 10.1145/3579051.3579063. URL: https://doi.org/10.1145/3579051.3579063.

[273] Riccardo Pozzi, Matteo Palmonari, Andrea Coletta, Luigi Bellomarini, Jens Lehmann, and Sahar Vahdati. "ReFactX: Scalable Reasoning with Reliable Facts via Constrained Generation". In: *The Semantic Web – ISWC 2025*. Cham: Springer Nature Switzerland, 2026, pp. 290–308. ISBN: 978-3-032-09527-5. DOI: 10.1007/978-3-032-09527-5_16. URL: https://doi.org/10.1007/978-3-032-09527-5_16.

[274] Riccardo Pozzi, Riccardo Rubini, Christian Bernasconi, and Matteo Palmonari. "Named Entity Recognition and Linking for Entity Extraction from Italian Civil Judgements". In: *AIxIA 2023 – Advances in Artificial Intelligence*. Cham: Springer Nature Switzerland, 2023, pp. 187–201. ISBN: 978-3-031-47546-7. DOI: 10.1007/978-3-031-47546-7_13. URL: https://link.springer.com/content/pdf/10.1007/978-3-031-47546-7.pdf.

[275] Soumajit Pramanik, Jesujoba Alabi, Rishiraj Saha Roy, and Gerhard Weikum. "Uniqorn: Unified question answering over RDF knowledge graphs and natural language text". In: *Journal of Web Semantics* 83 (2024), p. 100833. ISSN: 1570-8268. DOI: https://doi.org/10.1016/j.websem.2024.100833. URL: https://www.sciencedirect.com/science/article/pii/S1570826824000192.

[276] Luigi Procopio, Simone Conia, Edoardo Barba, and Roberto Navigli. "Entity Disambiguation with Entity Definitions". In: *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*. Dubrovnik, Croatia: Association for Computational Linguistics, May 2023, pp. 1297–1303. DOI: 10.18653/v1/2023.eacl-main.93. URL: https://aclanthology.org/2023.eacl-main.93/.

[277] Eric Prud'hommeaux and Gavin Carothers. *RDF 1.1 Turtle: Terse RDF Triple Language*. W3C Recommendation. 2014. URL: http://www.w3.org/TR/2014/REC-turtle-20140225/.

[278] Eric Prud'hommeaux and Andy Seaborne. *SPARQL Query Language for RDF*. W3C Recommendation. 2008. URL: http://www.w3.org/TR/rdf-sparql-query/.

[279] Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. "Stanza: A Python Natural Language Processing Toolkit for Many Human Languages". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. Online: Association for Computational Linguistics, July 2020, pp. 101–108. DOI: 10.18653/v1/2020.acl-demos.14. URL: https://aclanthology.org/2020.acl-demos.14/.

[280] Muhammad Reza Qorib, Geonsik Moon, and Hwee Tou Ng. "Are Decoder-Only Language Models Better than Encoder-Only Language Models in Understanding Word Meaning?" In: Bangkok, Thailand: Association for Computational Linguistics, Aug. 2024, pp. 16339–16347. DOI: 10.18653/v1/2024.findings-acl.967. URL: https://aclanthology.org/2024.findings-acl.967/.

[281] Qwen, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, et al. *Qwen2.5 Technical Report*. arXiv preprint arXiv:2412.15115. 2025. DOI: 10.48550/arXiv.2412.15115. arXiv: 2412.15115 [cs.CL].

[282] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine Mcleavey, and Ilya Sutskever. "Robust Speech Recognition via Large-Scale Weak Supervision". In: *ICML 2023*. PMLR, 2023, pp. 28492–28518. URL: https://proceedings.mlr.press/v202/radford23a.html.

[283] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. *Improving Language Understanding by Generative Pre-Training*. Technical Report. OpenAI, 2018. URL: https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf.

[284] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. *Language models are unsupervised multitask learners*. Technical Report. OpenAI, 2019. URL: https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf.

[285] Colin Raffel et al. "Exploring the limits of transfer learning with a unified text-to-text transformer". In: *J. Mach. Learn. Res.* 21.1 (Jan. 2020). ISSN: 1532-4435. URL: https://dl.acm.org/doi/pdf/10.5555/3455716.3455856.

[286] Jonathan Raiman and Olivier Raiman. "DeepType: Multilingual Entity Linking by Neural Type System Evolution". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 32.1 (Apr. 2018). DOI: 10.1609/aaai.v32i1.12008. URL: https://ojs.aaai.org/index.php/AAAI/article/view/12008.

[287] Lance Ramshaw and Mitch Marcus. "Text Chunking using Transformation-Based Learning". In: *Third Workshop on Very Large Corpora*. 1995. URL: https://aclanthology.org/W95-0107/.

[288] Priyanka Ranade and Anupam Joshi. "FABULA: Intelligence Report Generation Using Retrieval-Augmented Narrative Construction". In: ASONAM '23. Kusadasi, Turkiye: Association for Computing Machinery, 2024, pp. 603–610. ISBN: 9798400704093. DOI: 10.1145/3625007.3627505. URL: https://doi.org/10.1145/3625007.3627505.

[289] Delip Rao, Paul McNamee, and Mark Dredze. "Entity Linking: Finding Extracted Entities in a Knowledge Base". In: *Multi-source, Multilingual Information Extraction and Summarization*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 93–115. ISBN: 978-3-642-28569-1. DOI: 10.1007/978-3-642-28569-1_5. URL: https://doi.org/10.1007/978-3-642-28569-1_5.

[290] Giulio Ravasio and Leonardo Di Perna. *GilBERTo: A pretrained language model based on RoBERTa for Italian*. GitHub repository. 2020. URL: https://github.com/idb-ita/GilBERTo (visited on 10/15/2025).

[291] M. Recasens and E. Hovy. "Blanc: Implementing the rand index for coreference evaluation". In: *Nat. Lang. Eng.* 17.4 (Oct. 2011), pp. 485–510. ISSN: 1351-3249. DOI: 10.1017/S135132491000029X. URL: https://doi.org/10.1017/S135132491000029X.

[292] *Regio Decreto 16 marzo 1942, n. 262, Codice Civile Italiano (Italian Civil Code)*. https://www.normattiva.it/uri-res/N2Ls?urn:nir:stato:codice.civile:1942-03-16;262. 1942.

[293] *Regio Decreto 28 ottobre 1940, n. 1443, Codice di Procedura Civile (Italian Code of Civil Procedure)*. https://www.normattiva.it/uri-res/N2Ls?urn:nir:stato:regio.decreto:1940-10-28;1443. 1940.

[294] *Regio Decreto 30 gennaio 1941, n. 12, Ordinamento Giudiziario (Judicial Organization)*. https://www.normattiva.it/uri-res/N2Ls?urn:nir:stato:regio.decreto:1941-01-30;12~art115-com3. 1941.

[295] Marta Regis, Paulo Serra, and Edwin R. van den Heuvel. "Random autoregressive models: A structured overview". In: *Econometric Reviews* 41.2 (2022), pp. 207–230. DOI: 10.1080/07474938.2021.1899504. eprint: https://doi.org/10.1080/07474938.2021.1899504. URL: https://doi.org/10.1080/07474938.2021.1899504.

[296] Elizabeth Reid. *Generative AI in Search: Let Google Do the Searching for You*. Google Blog. Accessed: 2025-10-11. May 2024. URL: https://blog.google/products/search/generative-ai-google-search-may-2024/.

[297] Emily Reif et al. "Visualizing and Measuring the Geometry of BERT". In: *Advances in Neural Information Processing Systems*. Vol. 32. Curran Associates, Inc., 2019. URL: https://proceedings.neurips.cc/paper_files/paper/2019/file/159c1ffe5b61b41b3c4d8f4c2150f6c4-Paper.pdf.

[298] Martin Riedl and Sebastian Padó. "A Named Entity Recognition Shootout for German". In: Melbourne, Australia: Association for Computational Linguistics, July 2018, pp. 120–125. DOI: 10.18653/v1/P18-2020. URL: https://aclanthology.org/P18-2020/.

[299] Giuseppe Rizzo, Bianca Pereira, Andrea Varga, Marieke Van Erp, and Amparo Elizabeth Cano Basave. "Lessons learnt from the Named Entity rEcognition and Linking (NEEL) challenge series: Emerging Trends in Mining Semantics from Tweets". In: *Semantic Web* 8.5 (2017), pp. 667–700. DOI: 10.3233/SW-170276.

[300] Adam Roberts, Colin Raffel, and Noam Shazeer. "How Much Knowledge Can You Pack Into the Parameters of a Language Model?" In: Online: Association for Computational Linguistics, Nov. 2020, pp. 5418–5426. DOI: 10.18653/v1/2020.emnlp-main.437. URL: https://aclanthology.org/2020.emnlp-main.437/.

[301] Kirk Roberts, Dina Demner-Fushman, Ellen M. Voorhees, and William R. Hersh. "Overview of the TREC 2016 Clinical Decision Support Track". In: *Text Retrieval Conference (TREC)*. 2016. URL: https://trec.nist.gov/pubs/trec25/papers/Overview-CL.pdf.

[302] Víctor Rodríguez-Doncel, Socorro Bernardos, Rebeca Varela, and Patricia Martín-Chozas. *Lynx D2.4 Data Management Plan*. en. Creative Commons Attribution 4.0 International. 2019. DOI: 10.5281/zenodo.3236320. URL: https://zenodo.org/record/3236320.

[303] Vishal Singh Roha, Naveen Saini, Sriparna Saha, and Jose G. Moreno. "MOO-CMDS+NER: Named Entity Recognition-Based Extractive Comment-Oriented Multi-document Summarization". In: *Advances in Information Retrieval*. Cham: Springer Nature Switzerland, 2023, pp. 580–588. ISBN: 978-3-031-28238-6. DOI: 10.1007/978-3-031-28238-6_49.

[304] Lior Rokach and Oded Maimon. "Clustering Methods". In: *Data Mining and Knowledge Discovery Handbook*. Boston, MA: Springer US, 2005, pp. 321–352. ISBN: 978-0-387-25465-4. DOI: 10.1007/0-387-25465-X_15. URL: https://doi.org/10.1007/0-387-25465-X_15.

[305] Henry Rosales-Méndez, Aidan Hogan, and Barbara Poblete. "VoxEL: a benchmark dataset for multilingual entity linking". In: *The Semantic Web–ISWC 2018: 17th International Semantic Web Conference, Monterey, CA, USA, October 8–12, 2018, Proceedings, Part II 17*. Springer. 2018. DOI: 10.1007/978-3-030-00668-6_11.

[306] Stuart J. Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. 4th. Pearson, 2020. ISBN: 9780134610993.

[307] Adam Aron Rynkiewicz, Raul Palma, and Piotr Formanowicz. "Universal entity linking". In: *Engineering Applications of Artificial Intelligence* 161 (2025), p. 112185. ISSN: 0952-1976. DOI: https://doi.org/10.1016/j.engappai.2025.112185. URL: https://www.sciencedirect.com/science/article/pii/S0952197625021931.

[308] Oscar Sainz, Iker Garca-Ferrero, Rodrigo Agerri, Oier Lopez de Lacalle, German Rigau, and Eneko Agirre. "GoLLIE: Annotation Guidelines improve Zero-Shot Information-Extraction". In: *The Twelfth International Conference on Learning Representations*. 2024. URL: https://openreview.net/forum?id=Y3wpuxd7u9.

[309] Carlo Sansone and Giancarlo Sperlí. "Legal Information Retrieval systems: State-of-the-art and open issues". In: *Information Systems* 106 (2022), p. 101967. ISSN: 0306-4379. DOI: https://doi.org/10.1016/j.is.2021.101967. URL: https://www.sciencedirect.com/science/article/pii/S0306437921001551.

[310] Sunita Sarawagi. “Information Extraction”. In: *Foundations and Trendső in Databases* 1.3 (2008), pp. 261–377. ISSN: 1931-7883. DOI: 10.1561/1900000003. URL: http://dx.doi.org/10.1561/1900000003.

[311] Bhaskarjit Sarmah, Dhagash Mehta, Benika Hall, Rohan Rao, Sunil Patel, and Stefano Pasquali. “HybridRAG: Integrating Knowledge Graphs and Vector Retrieval Augmented Generation for Efficient Information Extraction”. In: ICAIF ’24. Brooklyn, NY, USA: Association for Computing Machinery, 2024, pp. 608–616. ISBN: 9798400710810. DOI: 10.1145/3677052.3698671. URL: https://doi.org/10.1145/3677052.3698671.

[312] Denis Savenkov and Eugene Agichtein. “When a Knowledge Base Is Not Enough: Question Answering over Knowledge Bases with External Text Data”. In: SIGIR ’16. Pisa, Italy: Association for Computing Machinery, 2016, pp. 235–244. ISBN: 9781450340694. DOI: 10.1145/2911451.2911536. URL: https://doi.org/10.1145/2911451.2911536.

[313] Uma Sawant, Saurabh Garg, Soumen Chakrabarti, and Ganesh Ramakrishnan. “Neural architecture for question answering using a knowledge graph and web corpus”. In: *Inf. Retr.* 22.34 (Aug. 2019), pp. 324–349. ISSN: 1386-4564. DOI: 10.1007/s10791-018-9348-8. URL: https://doi.org/10.1007/s10791-018-9348-8.

[314] Schema.org Community. *Schema.org Vocabulary: description*. URL: https://schema.org/description (visited on 10/20/2025).

[315] Timo Schick et al. *Toolformer: Language Models Can Teach Themselves to Use Tools*. 2023. arXiv: 2302.04761 [cs.CL]. URL: https://arxiv.org/abs/2302.04761.

[316] Torsten Scholak, Nathan Schucher, and Dzmitry Bahdanau. “PICARD: Parsing Incrementally for Constrained Auto-Regressive Decoding from Language Models”. In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2021, pp. 9895–9901. DOI: 10.18653/v1/2021.emnlp-main.779.

[317] Stefan Schweter. *Italian BERT and ELECTRA models*. Version 1.0.1. Nov. 2020. DOI: 10.5281/zenodo.4263142. URL: https://doi.org/10.5281/zenodo.4263142.

[318] Stefan Schweter and Johannes Baiter. “Towards Robust Named Entity Recognition for Historic German”. In: Florence, Italy: Association for Computational Linguistics, Aug. 2019, pp. 96–103. DOI: 10.18653/v1/W19-4312. URL: https://aclanthology.org/W19-4312/.

[319] Andy Seaborne et al. *SPARQL/Update: A Language for Updating RDF Graphs*. W3C Member Submission. July 2008. URL: http://www.w3.org/Submission/2008/SUBM-SPARQL-Update-20080715/.

[320] Satoshi Sekine and Chikashi Nobata. “Definition, Dictionaries and Tagger for Extended Named Entity Hierarchy”. In: *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC’04)*. Lisbon, Portugal: European Language Resources Association (ELRA), May 2004. URL: https://aclanthology.org/L04-1051/.

[321] Satoshi Sekine, Kiyoshi Sudo, and Chikashi Nobata. “Extended Named Entity Hierarchy”. In: *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC’02)*. Las Palmas, Canary Islands - Spain: European Language Resources Association (ELRA), May 2002. URL: https://aclanthology.org/L02-1120/.

[322] Priyanka Sen, Alham Fikri Aji, and Amir Saffari. "Mintaka: A Complex, Natural, and Multilingual Dataset for End-to-End Question Answering". In: *Proceedings of the 29th International Conference on Computational Linguistics*. International Committee on Computational Linguistics, 2022, pp. 1604–1619. URL: https://aclanthology.org/2022.coling-1.138.

[323] Rico Sennrich, Barry Haddow, and Alexandra Birch. "Neural Machine Translation of Rare Words with Subword Units". In: Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 1715–1725. DOI: 10.18653/v1/P16-1162. URL: https://aclanthology.org/P16-1162/.

[324] Özge Sevgili, Artem Shelmanov, Mikhail Arkhipov, Alexander Panchenko, and Chris Biemann. "Neural entity linking: A survey of models based on deep learning". In: *Semantic Web* 13.3 (2022), pp. 527–570. DOI: 10.3233/SW-222986. URL: https://journals.sagepub.com/doi/abs/10.3233/SW-222986.

[325] Thomas Shafee, Daniel Mietchen, Tiago Lubiana, Dariusz Jemielniak, and Andra Waagmeester. "Ten quick tips for editing Wikidata". In: *PLOS Computational Biology* 19.7 (2023), e1011235. DOI: 10.1371/journal.pcbi.1011235.

[326] Zhihong Shao, Yeyun Gong, Yelong Shen, Minlie Huang, Nan Duan, and Weizhu Chen. "Enhancing Retrieval-Augmented Large Language Models with Iterative Retrieval-Generation Synergy". In: *Findings of the Association for Computational Linguistics: EMNLP 2023*. Singapore: Association for Computational Linguistics, Dec. 2023, pp. 9248–9274. DOI: 10.18653/v1/2023.findings-emnlp.620. URL: https://aclanthology.org/2023.findings-emnlp.620/.

[327] Alok Sharma, Artem Lysenko, Shangru Jia, Keith A Boroevich, and Tatsuhiko Tsunoda. "Advances in AI and machine learning for predictive medicine". In: *Journal of Human Genetics* 69.10 (2024), pp. 487–497. DOI: 10.1038/s10038-024-01231-y.

[328] Wei Shen, Jianyong Wang, and Jiawei Han. "Entity Linking with a Knowledge Base: Issues, Techniques, and Solutions". In: *IEEE Transactions on Knowledge and Data Engineering* 27.2 (2015), pp. 443–460. DOI: 10.1109/TKDE.2014.2327028.

[329] Amit Sheth, Manas Gaur, Kaushik Roy, and Keyur Faldu. "Knowledge-Intensive Language Understanding for Explainable AI". In: *IEEE Internet Computing* 25.5 (2021), pp. 19–24. DOI: 10.1109/MIC.2021.3101919.

[330] Luke Shrimpton, Victor Lavrenko, and Miles Osborne. "Sampling Techniques for Streaming Cross Document Coreference Resolution". In: *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Denver, Colorado: Association for Computational Linguistics, May 2015, pp. 1391–1396. DOI: 10.3115/v1/N15-1158. URL: https://aclanthology.org/N15-1158/.

[331] Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. "Retrieval Augmentation Reduces Hallucination in Conversation". In: *Findings of the Association for Computational Linguistics: EMNLP 2021*. Association for Computational Linguistics, 2021, pp. 3784–3803. DOI: 10.18653/v1/2021.findings-emnlp.320.

[332] Marco Siino, Mariana Falco, Daniele Croce, and Paolo Rosso. "Exploring LLMs Applications in Law: A Literature Review on Current Legal NLP Approaches". In: *IEEE Access* 13 (2025), pp. 18253–18276. DOI: 10.1109/ACCESS.2025.3533217.

[333] A. Singhal. *Introducing the Knowledge Graph: things, not strings*. May 2012. URL: https://googleblog.blogspot.co.at/2012/05/introducing-knowledge-graph-things-not.html (visited on 08/23/2025).

[334] Jennifer Sor. "ChatGPT is now being used by 10% of the world's adult population". In: *Business Insider* (Oct. 2025). Accessed: 2025-10-11. URL: https://www.businessinsider.com/chatgpt-users-growth-openai-growth-sam-altman-ai-llm-2025-10.

[335] Daniil Sorokin and Iryna Gurevych. "Mixing Context Granularities for Improved Entity Linking on Question Answering Data across Entity Categories". In: *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*. New Orleans, Louisiana: Association for Computational Linguistics, June 2018, pp. 65–75. DOI: 10.18653/v1/S18-2007. URL: https://aclanthology.org/S18-2007/.

[336] Francesco Sovrano, Monica Palmirani, Fabio Vitali, et al. "Legal knowledge extraction for knowledge graph based question-answering". In: *Frontiers in Artificial Intelligence and Applications* 334 (2020), pp. 143–153. URL: https://cris.unibo.it/retrieve/e1dcb336-1cbb-7715-e053-1705fe0a6cc9/FAIA-334-FAIA200858.pdf.

[337] Reuters Staff. *Italy enacts AI law covering privacy, oversight, child access*. Reuters. Sept. 2025. URL: https://www.reuters.com/technology/italy-enacts-ai-law-covering-privacy-oversight-child-access-2025-09-17/ (visited on 09/19/2025).

[338] Stanford NLP Group. *Stanford CoreNLP Named Entity Recognizer (NER) Pipeline Overview*. 2025. URL: https://stanfordnlp.github.io/CoreNLP/ner.html#ner-pipeline-overview (visited on 08/21/2025).

[339] Emma Strubell, Ananya Ganesh, and Andrew McCallum. "Energy and Policy Considerations for Deep Learning in NLP". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, July 2019, pp. 3645–3650. DOI: 10.18653/v1/P19-1355. URL: https://aclanthology.org/P19-1355/.

[340] Chia-Yi Su, Aakash Bansal, Vijayanta Jain, Sepideh Ghanavati, and Collin McMillan. "A Language Model of Java Methods with Train/Test Deduplication". In: ESEC/FSE 2023. San Francisco, CA, USA: Association for Computing Machinery, 2023, pp. 2152–2156. ISBN: 9798400703270. DOI: 10.1145/3611643.3613090. URL: https://doi.org/10.1145/3611643.3613090.

[341] N. Subbulakshmi, S. Ariffa Begum, Adarsha Kumar, Shivam Kumar, Shivam Satyarthi, and Aman Singh. "Forensic Investigation and WhatsApp Chat Analysis Using Named Entity Recognition with Web Based Visualization". In: *2024 International Conference on Sustainable Communication Networks and Application (ICSCNA)*. 2024, pp. 1683–1688. DOI: 10.1109/ICSCNA63714.2024.10864198.

[342] Fabian M. Suchanek, Mehwish Alam, Thomas Bonald, Lihu Chen, Pierre-Henri Paris, and Jules Soria. "YAGO 4.5: A Large and Clean Knowledge Base with a Rich Taxonomy". In: *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR '24. Washington DC, USA: Association for Computing Machinery, 2024, pp. 131–140. ISBN: 9798400704314. DOI: 10.1145/3626772.3657876. URL: https://doi.org/10.1145/3626772.3657876.

[343] Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. "Yago: a core of semantic knowledge". In: *Proceedings of the 16th International Conference on World Wide Web*. WWW '07. Banff, Alberta, Canada: Association for Computing Machinery, 2007, pp. 697–706. ISBN: 9781595936547. DOI: 10.1145/1242572.1242667. URL: https://doi.org/10.1145/1242572.1242667.

[344] Li Sun, Florian Luisier, Kayhan Batmanghelich, Dinei Florencio, and Cha Zhang. "From Characters to Words: Hierarchical Pre-trained Language Model for Open-vocabulary Language Understanding". In: Toronto, Canada: Association for Computational Linguistics, July 2023, pp. 3605–3620. DOI: 10.18653/v1/2023.acl-long.200. URL: https://aclanthology.org/2023.acl-long.200/.

[345] Yaming Sun, Lei Lin, Duyu Tang, Nan Yang, Zhenzhou Ji, and Xiaolong Wang. "Modeling mention, context and entity with neural networks for entity disambiguation". In: *Proceedings of the 24th International Conference on Artificial Intelligence*. IJCAI'15. Buenos Aires, Argentina: AAAI Press, 2015, pp. 1333–1339. ISBN: 9781577357384. URL: https://www.ijcai.org/Proceedings/15/Papers/192.pdf.

[346] Beth M. Sundheim. "Overview of results of the MUC-6 evaluation". In: MUC6 '95. Columbia, Maryland: Association for Computational Linguistics, 1995, pp. 13–31. ISBN: 1558604022. DOI: 10.3115/1072399.1072402. URL: https://doi.org/10.3115/1072399.1072402.

[347] Chelse Swoopes, Ziwei Gu, and Elena L. Glassman. *Interface Design to Support Legal Reading and Writing: Insights from Interviews with Legal Experts*. 2025. DOI: 10.48550/arXiv.2509.24854. arXiv: 2509.24854 [cs.HC]. URL: https://arxiv.org/abs/2509.24854.

[348] Pedro Szekely et al. "Building and Using a Knowledge Graph to Combat Human Trafficking". In: *The Semantic Web - ISWC 2015*. Cham: Springer International Publishing, 2015, pp. 205–221. ISBN: 978-3-319-25010-6. DOI: 10.1007/978-3-319-25010-6_12.

[349] Minna Tamper, Arttu Oksanen, Jouni Tuominen, Aki Hietanen, and Eero Hyvönen. "Automatic Annotation Service APPI: Named Entity Linking in Legal Domain". In: *The Semantic Web: ESWC 2020 Satellite Events: ESWC 2020 Satellite Events, Heraklion, Crete, Greece, May 31 June 4, 2020, Revised Selected Papers*. Heraklion, Crete, Greece: Springer-Verlag, 2020, pp. 208–213. ISBN: 978-3-030-62326-5. DOI: 10.1007/978-3-030-62327-2_36. URL: https://doi.org/10.1007/978-3-030-62327-2_36.

[350] Yongmei Tan, Di Zheng, Maolin Li, and Xiaojie Wang. "BUPTTeam Participation at TAC 2015 Knowledge Base Population". In: Nov. 2015. URL: https://tac.nist.gov/publications/2015/participant.papers/TAC2015.BUPTTeam.proceedings.pdf.

[351] The PostgreSQL Global Development Group. *PostgreSQL 18 Documentation: Aggregate Functions*. 2025. URL: https://www.postgresql.org/docs/18/tutorial-agg.html (visited on 10/18/2025).

[352] The PostgreSQL Global Development Group. *PostgreSQL Documentation*. 2025. URL: https://www.postgresql.org/ (visited on 10/18/2025).

[353] Guiyao Tie et al. *A Survey on Post-training of Large Language Models*. 2025. arXiv: 2503.06072 [cs.CL]. URL: https://arxiv.org/abs/2503.06072.

[354] Erik F. Tjong Kim Sang and Fien De Meulder. "Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition". In: *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*. 2003, pp. 142–147. URL: https://aclanthology.org/W03-0419/.

[355] Lisa Torrey and Jude Shavlik. "Transfer learning". In: *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques*. IGI Global Scientific Publishing, 2010, pp. 242–264. URL: https://ftp.cs.wisc.edu/machine-learning/shavlik-group/torrey.handbook09.pdf.

[356] Hugo Touvron et al. *Llama 2: Open Foundation and Fine-Tuned Chat Models*. 2023. arXiv: 2307.09288 [cs.CL]. URL: https://arxiv.org/abs/2307.09288.

[357] Hugo Touvron et al. *LLaMA: Open and Efficient Foundation Language Models*. 2023. arXiv: 2302.13971 [cs.CL]. URL: https://arxiv.org/abs/2302.13971.

[358] Richard Tzong-Han Tsai et al. "Various criteria in the evaluation of biomedical named entity recognition". In: *BMC bioinformatics* 7 (2006). DOI: 10.1186/1471-2105-7-92.

[359] Alan Turing. "Computing Machinery and Intelligence". In: *Mind* LIX.236 (Oct. 1950), pp. 433–460. DOI: 10.1093/mind/LIX.236.433. URL: https://courses.cs.umbc.edu/471/papers/turing.pdf.

[360] Rutger Van Oest. "A new coefficient of interrater agreement: The challenge of highly unequal category proportions." In: *Psychological Methods* 24.4 (2019), p. 439. DOI: 10.1037/met0000183. URL: https://biopen.bi.no/bi-xmlui/bitstream/handle/11250/2590643/Van%20Oest.pdf?sequence=4&isAllowed=y.

[361] Ashish Vaswani et al. "Attention is All you Need". In: *Advances in Neural Information Processing Systems*. Vol. 30. Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[362] Sonakshi Vij, Vidur Agarwal, Vinay Aggarwal, and Vaibhav Goyal. "WhatsApp Forensic Analysis for Group Chats Using Exploratory Data Analysis and Natural Language Processing". In: *Data Science and Applications*. Singapore: Springer Nature Singapore, 2025, pp. 31–49. ISBN: 978-981-96-2299-3. DOI: 10.1007/978-981-96-2299-3_3.

[363] Marc Vilain, John Burger, John Aberdeen, Dennis Connolly, and Lynette Hirschman. "A Model-Theoretic Coreference Scoring Scheme". In: *Sixth Message Understanding Conference (MUC-6): Proceedings of a Conference Held in Columbia, Maryland, November 6-8, 1995*. 1995. URL: https://aclanthology.org/M95-1005/.

[364] Antti Virtanen et al. *Multilingual is not enough: BERT for Finnish*. 2019. arXiv: 1912.07076 [cs.CL]. URL: https://arxiv.org/abs/1912.07076.

[365] VoxForge.org. *VoxForge*. URL: https://www.voxforge.org/ (visited on 10/14/2025).

[366] Denny Vrandei and Markus Krötzsch. "Wikidata: a free collaborative knowledgebase". In: *Commun. ACM* 57.10 (Sept. 2014), pp. 78–85. ISSN: 0001-0782. DOI: 10.1145/2629489.

[367] W3C. *RDF Schema 1.1*. 2014. URL: https://www.w3.org/TR/rdf-schema/ (visited on 10/20/2025).

[368] Meiling Wang, Min Li, Kewei Sun, and Zhirong Hou. "Entity Difference Modeling Based Entity Linking for Question Answering over Knowledge Graphs". In: *Natural Language Processing and Chinese Computing*. Cham: Springer International Publishing, 2022, pp. 221–233. ISBN: 978-3-031-17120-8. DOI: 10.1007/978-3-031-17120-8_18.

[369] Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. "Knowledge Graph Embedding: A Survey of Approaches and Applications". In: *IEEE Transactions on Knowledge and Data Engineering* 29.12 (2017), pp. 2724–2743. DOI: 10.1109/TKDE.2017.2754499.

[370] Ruili Wang, Feng Hou, Steven F. Cahan, Li Chen, Xiaoyun Jia, and Wanting Ji. "Fine-Grained Entity Typing With a Type Taxonomy: A Systematic Review". In: *IEEE Transactions on Knowledge and Data Engineering* 35.5 (2023), pp. 4794–4812. DOI: 10.1109/TKDE.2022.3148980.

[371] Shuhe Wang et al. "GPT-NER: Named Entity Recognition via Large Language Models". In: Albuquerque, New Mexico: Association for Computational Linguistics, Apr. 2025, pp. 4257–4275. ISBN: 979-8-89176-195-7. DOI: 10.18653/v1/2025.findings-naacl.239. URL: https://aclanthology.org/2025.findings-naacl.239/.

[372] Song Wang, Yaochen Zhu, Haochen Liu, Zaiyi Zheng, Chen Chen, and Jundong Li. "Knowledge Editing for Large Language Models: A Survey". In: *ACM Comput. Surv.* 57.3 (Nov. 2024). ISSN: 0360-0300. DOI: 10.1145/3698590. URL: https://doi.org/10.1145/3698590.

[373] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, et al. "Chain-of-Thought Prompting Elicits Reasoning in Large Language Models". In: *Advances in Neural Information Processing Systems*. Vol. 35. Curran Associates, Inc., 2022, pp. 24824–24837. URL: https://proceedings.neurips.cc/paper%5C_files/paper/2022/file/9d5609613524ecf4f15af0f7b31abca4-Paper-Conference.pdf.

[374] Laura Weidinger et al. "Taxonomy of Risks posed by Language Models". In: *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*. FAccT '22. Seoul, Republic of Korea: Association for Computing Machinery, 2022, pp. 214–229. ISBN: 9781450393522. DOI: 10.1145/3531146.3533088. URL: https://doi.org/10.1145/3531146.3533088.

[375] Robert West, Evgeniy Gabrilovich, Kevin Murphy, Shaohua Sun, Rahul Gupta, and Dekang Lin. "Knowledge base completion via search-based question answering". In: WWW '14. Seoul, Korea: Association for Computing Machinery, 2014, pp. 515–526. ISBN: 9781450327442. DOI: 10.1145/2566486.2568032. URL: https://doi.org/10.1145/2566486.2568032.

[376] Wikidata contributors. *Wikidata: Statistics*. https://www.wikidata.org/wiki/Wikidata:Statistics. Accessed: 2025-09-24. 2025.

[377] Wikimedia Foundation. *Wikidata*. 2025. URL: https://www.wikidata.org/ (visited on 08/25/2025).

[378] Wikimedia Foundation. *Wikipedia*. 2025. URL: https://www.wikipedia.org/ (visited on 08/25/2025).

[379] Wikimedia Foundation. *Wikipedia*. 2025. URL: https://en.wikipedia.org/wiki/Wikipedia (visited on 09/24/2025).

[380] Wikipedia contributors. *Wikidata*. https://en.wikipedia.org/wiki/Wikidata. Accessed: 2025-09-24. 2025.

[381] Wikipedia contributors. *Wikipedia: Size of Wikipedia*. https://en.wikipedia.org/wiki/Wikipedia:Size_of_Wikipedia. Accessed: 2025-09-24. 2025.

[382] William E. Winkler. "String comparator metrics and enhanced decision rules in the Fellegi-Sunter model of record linkage". In: (1990). URL: http://files.eric.ed.gov/fulltext/ED325505.pdf.

[383] Thomas Wolf et al. "Transformers: State-of-the-Art Natural Language Processing". In: Online: Association for Computational Linguistics, Oct. 2020, pp. 38–45. DOI: 10.18653/v1/2020.emnlp-demos.6. URL: https://aclanthology.org/2020.emnlp-demos.6/.

[384] Rini Wongso, Meiliana, and Derwin Suhartono. "A literature review of question answering system using Named Entity Recognition". In: *2016 3rd International Conference on Information Technology, Computer, and Electrical Engineering (ICITACEE)*. 2016, pp. 274–277. DOI: 10.1109/ICITACEE.2016.7892454.

[385] Ledell Wu, Fabio Petroni, Martin Josifoski, Sebastian Riedel, and Luke Zettlemoyer. "Scalable Zero-shot Entity Linking with Dense Entity Retrieval". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, Nov. 2020, pp. 6397–6407. DOI: 10.18653/v1/2020.emnlp-main.519. URL: https://aclanthology.org/2020.emnlp-main.519/.

[386] Shijie Wu and Mark Dredze. "Are All Languages Created Equal in Multilingual BERT?" In: *Proceedings of the 5th Workshop on Representation Learning for NLP*. Online: Association for Computational Linguistics, July 2020, pp. 120–130. DOI: 10.18653/v1/2020.repl4nlp-1.16. URL: https://aclanthology.org/2020.repl4nlp-1.16/.

[387] Yuxiang Wu, Yu Zhao, Baotian Hu, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. "An Efficient Memory-Augmented Transformer for Knowledge-Intensive NLP Tasks". In: Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022, pp. 5184–5196. DOI: 10.18653/v1/2022.emnlp-main.346. URL: https://aclanthology.org/2022.emnlp-main.346/.

[388] Zhiheng Xi et al. "The rise and potential of large language model based agents: A survey". In: *Science China Information Sciences* 68.2 (2025), p. 121101. DOI: 10.1007/s11432-024-4222-0. URL: https://link.springer.com/content/pdf/10.1007/s11432-024-4222-0.pdf.

[389] Tianyun Xiao, Shanshan Kong, Zichen Zhang, Dianbo Hua, and Fengchun Liu. "A review of big data technology and its application in cancer care". In: *Computers in Biology and Medicine* 176 (2024), p. 108577. ISSN: 0010-4825. DOI: https://doi.org/10.1016/j.compbiomed.2024.108577. URL: https://www.sciencedirect.com/science/article/pii/S0010482524006620.

[390] Zilin Xiao, Ming Gong, Jie Wu, Xingyao Zhang, Linjun Shou, and Daxin Jiang. "Instructed Language Models with Retrievers Are Powerful Entity Linkers". In: *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Singapore: Association for Computational Linguistics, Dec. 2023, pp. 2267–2282. DOI: 10.18653/v1/2023.emnlp-main.139. URL: https://aclanthology.org/2023.emnlp-main.139/.

[391] Shufang Xie, Yingce Xia, Lijun Wu, Yiqing Huang, Yang Fan, and Tao Qin. "End-to-end entity-aware neural machine translation". In: *Machine Learning* 111.3 (2022), pp. 1181–1203.

[392] Tingyu Xie, Qi Li, Yan Zhang, Zuozhu Liu, and Hongwei Wang. "Self-Improving for Zero-Shot Named Entity Recognition with Large Language Models". In: Mexico City, Mexico: Association for Computational Linguistics, June 2024, pp. 583–593. DOI: 10.18653/v1/2024.naacl-short.49. URL: https://aclanthology.org/2024.naacl-short.49/.

[393] Amy Xin et al. "AtomR: Atomic Operator-Empowered Large Language Models for Heterogeneous Knowledge Reasoning". In: KDD '25. Toronto ON, Canada: Association for Computing Machinery, 2025, pp. 3344–3355. ISBN: 9798400714542. DOI: 10.1145/3711896.3736849. URL: https://doi.org/10.1145/3711896.3736849.

[394] Derong Xu et al. "Large language models for generative information extraction: A survey". In: *Frontiers of Computer Science* 18.6 (2024), p. 186357. DOI: 10.1007/s11704-024-40555-y.

[395] Ran Xu et al. "A Survey on Unifying Large Language Models and Knowledge Graphs for Biomedicine and Healthcare". In: *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*. KDD '25. Toronto ON, Canada: Association for Computing Machinery, 2025, pp. 6195–6205. ISBN: 9798400714542. DOI: 10.1145/3711896.3736556. URL: https://doi.org/10.1145/3711896.3736556.

[396] Zhentao Xu et al. "Retrieval-Augmented Generation with Knowledge Graphs for Customer Service Question Answering". In: SIGIR 2024. ACM, July 2024, pp. 2905–2909. DOI: 10.1145/3626772.3661370. URL: http://dx.doi.org/10.1145/3626772.3661370.

[397] Ikuya Yamada, Hiroyuki Shindo, Hideaki Takeda, and Yoshiyasu Takefuji. "Learning Distributed Representations of Texts and Entities from Knowledge Base". In: *Transactions of the Association for Computational Linguistics* 5 (2017), pp. 397–411. DOI: 10.1162/tacl_a_00069. URL: https://aclanthology.org/Q17-1028/.

[398] Ikuya Yamada, Koki Washio, Hiroyuki Shindo, and Yuji Matsumoto. "Global Entity Disambiguation with BERT". In: *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Seattle, United States: Association for Computational Linguistics, July 2022, pp. 3264–3271. DOI: 10.18653/v1/2022.naacl-main.238. URL: https://aclanthology.org/2022.naacl-main.238/.

[399] Tao Yang, Dong Du, and F. Zhang. "The TAI System for Trilingual Entity Discovery and Linking Track in TAC KBP 2017". In: *Proceedings of the 10th Text Analysis Conference (TAC 2017)*. NIST, 2017. URL: https://tac.nist.gov/publications/2017/participant.papers/TAC2017.TAI.proceedings.pdf.

[400] Tao Yang, Dong Du, and Feng Zhang. "The TAI System for Trilingual Entity Discovery and Linking Track in TAC KBP 2017". In: *Proceedings of the 10th Text Analysis Conference (TAC 2017)*. NIST, 2017. URL: https://tac.nist.gov/publications/2017/participant.papers/TAC2017.TAI.proceedings.pdf.

[401] Shunyu Yao et al. *ReAct: Synergizing Reasoning and Acting in Language Models*. arXiv preprint arXiv:2210.03629. 2023. DOI: 10.48550/arXiv.2210.03629. arXiv: 2210.03629 [cs.CL]. URL: https://arxiv.org/abs/2210.03629.

[402] Yunzhi Yao et al. "Editing Large Language Models: Problems, Methods, and Opportunities". In: Singapore: Association for Computational Linguistics, Dec. 2023, pp. 10222–10240. DOI: 10.18653/v1/2023.emnlp-main.632. URL: https://aclanthology.org/2023.emnlp-main.632/.

[403] Susumu Yata. *MARISA: Matching Algorithm with Recursively Implemented StorAge*. 2025. URL: https://github.com/s-yata/marisa-trie (visited on 10/06/2025).

[404] Wen-tau Yih, Matthew Richardson, Chris Meek, Ming-Wei Chang, and Jina Suh. "The Value of Semantic Parse Labeling for Knowledge Base Question Answering". In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics, 2016, pp. 201–206. DOI: 10.18653/v1/P16-2033.

[405] Mohamed Amir Yosef, Sandro Bauer, Johannes Hoffart, Marc Spaniol, and Gerhard Weikum. "HYENA: Hierarchical Type Classification for Entity Names". In: *Proceedings of COLING 2012: Posters*. Mumbai, India: The COLING 2012 Organizing Committee, Dec. 2012, pp. 1361–1370. URL: https://aclanthology.org/C12-2133/.

[406] Mohamed Amir Yosef, Johannes Hoffart, Ilaria Bordino, Marc Spaniol, and Gerhard Weikum. "AIDA: an online tool for accurate disambiguation of named entities in text and tables". In: *Proc. VLDB Endow.* 4.12 (Aug. 2011), pp. 1450–1453. ISSN: 2150-8097. DOI: 10.14778/3402755.3402793. URL: https://doi.org/10.14778/3402755.3402793.

[407] Munazza Zaib, Wei Emma Zhang, Quan Z Sheng, Adnan Mahmood, and Yang Zhang. "Conversational question answering: A survey". In: *Knowledge and Information Systems* 64.12 (2022), pp. 3151–3195. DOI: doi.org/10.1007/s10115-022-01744-y. URL: https://link.springer.com/content/pdf/10.1007/s10115-022-01744-y.pdf.

[408] Klim Zaporojets, Lucie-Aimée Kaffee, Johannes Deleu, Thomas Demeester, Chris Develder, and Isabelle Augenstein. "TempEL: Linking Dynamically Evolving and Newly Emerging Entities". In: *Advances in Neural Information Processing Systems*. Vol. 35. Curran Associates, Inc., 2022, pp. 1850–1866. URL: https://proceedings.neurips.cc/paper_files/paper/2022/file/0c3464f16c854d395b880cf9e7bcaf2f-Paper-Datasets_and_Benchmarks.pdf.

[409] Urchade Zaratiana, Nadi Tomeh, Pierre Holat, and Thierry Charnois. "GLiNER: Generalist Model for Named Entity Recognition using Bidirectional Transformer". In: Mexico City, Mexico: Association for Computational Linguistics, June 2024, pp. 5364–5376. DOI: 10.18653/v1/2024.naacl-long.300. URL: https://aclanthology.org/2024.naacl-long.300/.

[410] Biao Zhang et al. *Encoder-Decoder Gemma: Improving the Quality-Efficiency Trade-Off via Adaptation*. 2025. arXiv: 2504.06225 [cs.CL]. URL: https://arxiv.org/abs/2504.06225.

[411] Heidi Zhang, Sina Semnani, Farhad Ghassemi, Jialiang Xu, Shicheng Liu, and Monica Lam. "SPAGHETTI: Open-Domain Question Answering from Heterogeneous Data Sources with Retrieval and Semantic Parsing". In: Bangkok, Thailand: Association for Computational Linguistics, Aug. 2024, pp. 1663–1678. DOI: 10.18653/v1/2024.findings-acl.96. URL: https://aclanthology.org/2024.findings-acl.96/.

[412] Jing Zhang et al. "Subgraph Retrieval Enhanced Model for Multi-hop Knowledge Base Question Answering". In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 5773–5784. DOI: 10.18653/v1/2022.acl-long.396. URL: https://aclanthology.org/2022.acl-long.396/.

[413] Qixuan Zhang, Xinyi Weng, Guangyou Zhou, Yi Zhang, and Jimmy Xiangji Huang. "ARL: An adaptive reinforcement learning framework for complex question answering over knowledge base". In: *Information Processing & Management* 59.3 (2022), p. 102933. ISSN: 0306-4573. DOI: https://doi.org/10.1016/j.ipm.2022.102933. URL: https://www.sciencedirect.com/science/article/pii/S0306457322000565.

[414] Zhongfei (Mark) Zhang, John J. Salerno, and Philip S. Yu. "Applying data mining in investigating money laundering crimes". In: *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '03. Washington, D.C.: Association for Computing Machinery, 2003, pp. 747–752. ISBN: 1581137370. DOI: 10.1145/956750.956851. URL: https://doi.org/10.1145/956750.956851.

[415] Tianyi Zhang*, Varsha Kishore*, Felix Wu*, Kilian Q. Weinberger, and Yoav Artzi. "BERTScore: Evaluating Text Generation with BERT". In: *International Conference on Learning Representations*. 2020. URL: https://openreview.net/forum?id=SkeHuCVFDr.

[416] Danna Zheng, Mirella Lapata, and Jeff Z. Pan. *How Reliable are LLMs as Knowledge Bases? Re-thinking Facutality and Consistency*. 2024. arXiv: 2407.13578 [cs.CL]. URL: https://arxiv.org/abs/2407.13578.

[417] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, et al. "Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena". In: *Advances in Neural Information Processing Systems*. Vol. 36. Curran Associates, Inc., 2023, pp. 46595–46623. URL: https://proceedings.neurips.cc/paper_files/paper/2023/file/91f18a1287b398d378ef22505bf41832-Paper-Datasets_and_Benchmarks.pdf.

[418] Ce Zhou et al. "A comprehensive survey on pretrained foundation models: A history from bert to chatgpt". In: *International Journal of Machine Learning and Cybernetics* (Nov. 2024). ISSN: 1868-808X. DOI: https://doi.org/10.1007/s13042-024-02443-6.

[419] Chunting Zhou et al. "LIMA: Less Is More for Alignment". In: *Thirty-seventh Conference on Neural Information Processing Systems*. 2023. URL: https://openreview.net/forum?id=KBMOKmX2he.

[420] Jiawei Zhou and Lei Chen. "OpenRAG: Optimizing RAG End-to-End via In-Context Retrieval Learning". In: *Scaling Self-Improving Foundation Models without Human Supervision*. 2025. URL: https://openreview.net/forum?id=WX0Y0rBsqo.

[421] Kang Zhou, Yuepei Li, Qing Wang, Qiao Qiao, and Qi Li. "GenDecider: Integrating "None of the Candidates" Judgments in Zero-Shot Entity Linking Re-ranking". In: *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*. Mexico City, Mexico: Association for Computational Linguistics, June 2024, pp. 239–245. DOI: 10.18653/v1/2024.naacl-short.22. URL: https://aclanthology.org/2024.naacl-short.22/.

[422] Fangwei Zhu, Jifan Yu, Hailong Jin, Lei Hou, Juanzi Li, and Zhifang Sui. "Learn to Not Link: Exploring NIL Prediction in Entity Linking". In: *Findings of the Association for Computational Linguistics: ACL 2023*. Toronto, Canada: Association for Computational Linguistics, July 2023, pp. 10846–10860. DOI: 10.18653/v1/2023.findings-acl.690. URL: https://aclanthology.org/2023.findings-acl.690/.

[423] Fengbin Zhu, Wenqiang Lei, Chao Wang, Jianming Zheng, Soujanya Poria, and Tat-Seng Chua. *Retrieving and Reading: A Comprehensive Survey on Open-domain Question Answering*. 2021. arXiv: 2101.00774 [cs.AI]. URL: https://arxiv.org/abs/2101.00774.

[424] Ayah Zirikly, Mona T. Diab, and Yassine Benajiba. "GWU English TAC-KBP EL Diagnostic Task with Name Mention". In: *Proceedings of the 10th Text Analysis Conference (TAC 2017)*. NIST, 2015. URL: https://tac.nist.gov/publications/2015/participant.papers/TAC2015.GWU.proceedings.pdf.

[425] Nazatul Nurlisa Zolkifli, Amir Ngah, and Aziz Deraman. "Version Control System: A Review". In: *Procedia Computer Science* 135 (2018). The 3rd International Conference on Computer Science and Computational Intelligence (ICCSCI 2018) : Empowering Smart Technology in Digital Era for a Better Life, pp. 408–415. ISSN: 1877-0509. DOI: https://doi.org/10.1016/j.procs.2018.08.191. URL: https://www.sciencedirect.com/science/article/pii/S1877050918314819.