UNICA

UNIVERSITÀ
DEGLI STUDI
DI CAGLIARI

Ph.D. DEGREE IN
Computer Science

Cycle XXXVIII

TITLE OF THE Ph.D. THESIS

Methods for Knowledge Graph Construction from Text Collections: Development and Applications

Scientific Disciplinary Sector(s)

INF/01

Ph.D. Student:	Vanni Zavarella
Supervisor	Prof. Gianni Fenu
Co-Supervisor	Prof. Diego Reforgiato Recupero

Final exam. Academic Year 2024/2025
Thesis defense session: February 2026

Statement of Authorship

I declare that this thesis entitled "Methods for Knowledge Graph Construction from Text Collections: Development and Applications" and the work presented in it are my own. I confirm that:

- this work was done while in candidature for this PhD degree;
- when I consulted the work published by others, this is always clearly attributed;
- when I quoted the work of others, the source is always given;
- I have acknowledged all main sources of help;
- with the exception of the above references, this thesis is entirely my own work;
- appropriate ethics guidelines were followed to conduct this research;
- for work done jointly with others, my contribution is clearly specified.

Abstract

Virtually every sector of society is experiencing a dramatic growth in the volume of unstructured textual data that is generated and published, from news and social media online interactions, through open access scholarly communications and observational data in the form of digital health records and online drug reviews. The volume and variety of data across all this range of domains has created both unprecedented opportunities and pressing challenges for extracting actionable knowledge for several application scenarios. However, the extraction of rich semantic knowledge demands the deployment of scalable and flexible automatic methods adaptable across text genres and schema specifications. Moreover, the full potential of these data can only be unlocked by coupling information extraction methods with Semantic Web techniques for the construction of full-fledged Knowledge Graphs, that are semantically transparent, explainable by design and interoperable.

In this thesis, we experiment with the application of Natural Language Processing, Machine Learning and Generative AI methods, powered by Semantic Web best practices, to the automatic construction of Knowledge Graphs from large text corpora, in three use case applications: the analysis of the Digital Transformation discourse in the global news and social media platforms; the mapping and trend analysis of recent research in the Architecture, Engineering, Construction and Operations domain from a large corpus of publications; the generation of causal relation graphs of biomedical entities from electronic health records and patient-authored drug reviews.

The contributions of this thesis to the research community are in terms of benchmark evaluation results, the design of customized algorithms and the creation of data resources in the form of Knowledge Graphs, together with data analysis results built on top of them. Most of the material presented in this thesis originates from research publications in international journals or conference proceedings.

Biography

Vanni Zavarella was born on June 23, 1978 in Sulmona (Italy). He is a PhD Candidate in Computer Science at the Department of Mathematics and Computer Science, University of Cagliari (Italy), under the supervision of Prof. Gianni Fenu and co-supervision of Prof. Diego Reforgiato Recupero. He received a MSc Degree in "Computer Science, Cognitive Science and Applications" from the University of Lorraine, France (formerly University of Nancy 2), with a specialization in Natural Language Processing.

Currently working as a freelance data scientist and NLP developer, he has served for more than 14 years as a Scientific Officer and consultant at the European Commission's Joint Research Centre (JRC) in the implementation and management of Natural Language Processing projects, being responsible for supporting JRC text analysis and media monitoring services in domains such as open source intelligence (OSINT), Global Health Surveillance and social media mining for Disaster Management. He is a former core developer of the popular news monitoring platform Europe Media Monitor (EMM).

He has co-authored over 40 research publications in international conferences and journals and has given talks and poster presentations at several conferences and workshops, including FSMNLP 2008, EACL 2014, ESWC 2014, ISCRAM 2017, LREC 2020, ECIR 2020, TEXT2KG 2024, UMAP 2024, LOD 2024 and 2025, etc. He has been program and organizing committee member of the CASE (Challenges and Applications of Automated Extraction of Socio-political Events from Text) workshops series.

Dissemination

The topics, techniques and resources presented in this Ph.D. thesis are the product of research efforts the resulted in scientific publications in international journals, conference proceedings and workshop proceedings. I express sincere gratitude to my co-authors for their invaluable contributions, which I acknowledge through the inclusive use of the scientific 'we' throughout this thesis. Moreover, during my nine month long research stay abroad at the Institute of Data Science and Artificial Intelligence of the Universidad of Navarra (DATAI), I had the privilege of collaborating with PhD Juan Carlos Gamero Salinas. The work described in Chapter 4 is the result of this collaboration.

I conceived the research concepts outlined in this thesis and undertook the majority of the research, implementation, testing and evaluation work. I conceptualized the methodologies, determined the research trajectories, and collected and analyzed the necessary datasets. The responsibility for script implementation also fell within my purview. Furthermore, I undertook the authorship of the papers, expertly navigating the peer-review process and iteratively refining them. My interactions with the co-authors were characterized by close collaboration and consultation. Their input encompassed offering insights into methodologies, providing technical assistance, engaging in the exploration of techniques, and contributing to the refinement of submitted work. Additionally, I assumed the role of presenter for 3 of the papers at conferences and workshops.

The detailed references to the produced papers are provided below.

Peer-reviewed Publications in International Journals:

i. **Vanni Zavarella** & Sergio Consoli, Diego Reforgiato Recupero, Gianni Fenu, Simone Angioni, Davide Buscaldi, Danilo Dessì, Francesco Osborne (2024). *Triplé-toile: Extraction of knowledge from microblogging text.* Heliyon, Volume 10, Issue 12, e32479 DOI: 10.1016/j.heliyon.2024.e324 (ISI/Scimago Q1)

ii. **Vanni Zavarella** & Juan Carlos Gamero-Salinas, Danilo Dessì, Sergio Consoli, Gianni Fenu, Diego Reforgiato Recupero *Mapping the AECO Research Landscape using Topic Modeling, Bibliometrics and Information Extraction methods*. under review by IEEE ACCESS

iii. **Vanni Zavarella** & Lorenzo Bertolini, Sergio Consoli, Gianni Fenu, Diego Reforgiato Recupero, Alessandro Zani. *Leveraging Large Language Models for Causal*

Relation Extraction in Biomedical Texts under review by Information Processing and Management, Special Issue on Causal Reasoning in Language Models.

Peer-reviewed Publications in International Conference and Workshop Proceedings:

i. **Vanni Zavarella** & Sergio Consoli, Diego Reforgiato Recupero and Gianni Fenu (2024) *Exploring Digital Health Trends in the Headlines via Knowledge Graph Analysis* proceedings of the 10th International Conference on Machine Learning, Optimization, and Data Science (LOD2024) by Springer Nature - Lecture Notes in Computer Science `https://www.springer.com/gp/computer-science/lncs` (Presenter)

ii. **Vanni Zavarella** & Diego Reforgiato, Sergio Consoli, Gianni Fenu (2024). *Charting the Landscape of Digital Health: Towards A Knowledge Graph Approach to News Media Analysis* Adjunct Proceedings of the 32nd ACM Conference on User Modeling, Adaptation and Personalization https://dl.acm.org/doi/abs/10.1145/3631700.3665237) (Rank B)

iii. **Vanni Zavarella** & Diego Reforgiato Recupero, Sergio Consoli, Gianni Fenu, Simone Angioni, Davide Buscaldi, Danilo Dessí, Francesco Osborne (2024). *Knowledge Graphs for Digital Transformation Monitoring in Social Media* CEUR Workshop Proceedings, 3rd International Workshop on Knowledge Graph Generation from Text (TEXT2KG), co-located with ESWC 2024 (`https://ceur-ws.org/Vol-3747/text2kg_paper8.pdf`)

iv. **Vanni Zavarella** & Juan Carlos Gamero-Salinas, Sergio Consoli (2024) *A Few-Shot Approach for Relation Extraction Domain Adaptation using Large Language Models* Proceedings of the Workshop on Deep Learning and Large Language Models for Knowledge Graphs (DL4KG@KDD2024) `https://ceur-ws.org/Vol-3894/` (Presenter)

v. **Vanni Zavarella** & Lorenzo Bertolini, Sergio Consoli, Gianni Fenu, Diego Reforgiato Recupero, Alessandro Zani. (2025) *LLM-Powered Knowledge Graph of Causal Relations in Drug Reviews* CEUR WORKSHOP PROCEEDINGS, 4th International Workshop on LLM-Integrated Knowledge Graph Generation from Text (Text2KG) `https://ceur-ws.org/Vol-4020/Paper_ID_9.pdf`

vi. **Vanni Zavarella** & Lorenzo Bertolini, Sergio Consoli, Gianni Fenu, Diego Reforgiato Recupero, Alessandro Zani (2025). *An Interactive Dashboard for Exploring Patient-Reported Drug-Condition Relations* under publication as conference post-proceedings of the 11th International Conference on Machine Learning, Optimization, and Data Science (LOD2025) by Springer Nature - Lecture Notes in Computer Science (Presenter)

Acknowledgments

I would like to express my gratitude to my supervisor, Prof. Gianni Fenu, and to my co-supervisor Prof. Diego Reforgiato Recupero for their continuous support and encouragement and for their invaluable management and strategic guidance throughout this research. Their insight and expertise have been instrumental in shaping this thesis and achieving the scientific results we have reached.

This work has leveraged the Collaboration Agreement (CA) #36805 between the Joint Research Centre of the European Commission and the Department of Mathematics and Computer Science of University of Cagliari aiming to develop Data Science applications for Healthcare, exploiting the value of health data by leveraging on novel cutting-edge technologies like those from Data Science and (Deep) Machine Learning, ensuring that the results obtained are used to support policy-making.

In this respect, I would like to thank the colleagues of the Digital Health Unit (JRC.F7) at the Joint Research Centre for the helpful guidance and support during the development of this research work. In particular, I would like to express my personal gratitude to Dr Sergio Consoli from JRC for his tireless support on improving the scientific rigor of this work. Nonetheless, all the views expressed in this thesis are purely mine and may not in any circumstance be regarded as stating an official position of the European Commission.

I would like to warmly thank the Institute of Data Science and Artificial Intelligence of the Universidad of Navarra and his Director, Prof. Jesús López Fidalgo, for making my research stay with them so enriching and in particular Juan Carlos Gamero Salinas, for helping making our scientific collaboration so fruitful and personally enjoyable at the same time. Moreover, I would like to collectively thank the various faculty members and researchers of the School of Architecture of the University of Navarra for their volunteer validation work on the SKG-AECO pipeline described in Chapter 4.

Above all, I owe my deepest gratitude to my partner Ana, for her unconditional love and faith in me throughout this journey. Her encouragement has been my constant source of motivation.

Finally, I dedicate this thesis to my parents Laura and Vittorio, their work ethic and high respect for the culture and science has been and will always be a guiding light in my professional and personal life.

x

Nomenclature

Abbreviations

NLP	Natural Language Processing
AI	Artificial Intelligence
KG	Knowledge Graph
SW	Semantic Web
RDF	Resource Description Framework
IRI	Internationalised Resource Identifier
OWL	Ontology Web Language
KB	Knowledge Base
ANN	Artificial Neural Networks
DL	Deep Learning
RNN	Recurrent Neural Network
CNN	Convolutional Neural Network
PEFT	Parameter-Efficient Fine-Tuning
GNN	Graph Neural Network
ML	Machine Learning
IE	Information Extraction
NER	Named Entity Recognition
RE	Relation Extraction
EL	Entity Linking
CRE	Causal Relation Extraction
ADE	Adverse Drug Event
LLM	Large Language Model
DT	Digital Transformation
AECO	Architecture, Engineering, Construction and Operations

Numerical Expressions

{n}k	{n} thousands
{n}M	{n} millions
{n}B	{n} billions
{n}T	{n} trillions

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Virtually every sector in society is experiencing a dramatic growth in the volume of unstructured textual data that is generated, stored and exchanged.

For example, news and social media (SM) online interactions produce massive streams of multi-domain text content timely reflecting both viral events or long term societal trends on a local to global level.

The globalization of scientific communities and the establishment of open access standards and dissemination platforms has brought about an average 4–6% yearly growth rate of scholarly communications [BM15], which in some areas are estimated to double their volume roughly every 10 years. Despite the advancement of indexing and semantic retrieval services, the volume of scientific publications are far beyond the manageable scope of human analysis or surveying. At the same time, the knowledge encoded in the text content of such documents remains still inaccessible to machine services.

In the medical domain, the digitization of health records and the increasing volume of patient-reported experiences with drugs and therapies in online forums, specialized websites and social media channels have opened up completely new scenarios for the passive collection of observational data [FCea24].

The volume and variety of data across all this wide range of domains has created both unprecedented opportunities and pressing challenges for extracting actionable knowledge for several application scenarios.

First, the monitoring and analysis of large and pervasive societal processes such as Digital Transformation (DT) can be effectively carried out via the detection and tracking of its key players' interactions from fast responsive SM platforms. Secondly, the dynamics of the scientific research and innovation in a determined domain can be studied by tracking the research entities and their semantic relations, such as the *tasks*, *methods*, *algorithms* and the *metrics* they have been evaluated against, hidden within the text of scholarly publications. Finally, the knowledge extracted from clinical notes and crowdsourced patient reviews can be used to extend and update current authoritative

Knowledge Bases (KB) about medical entities like drugs, therapies, conditions and their complex interactions.

However, the extraction of rich semantic knowledge from the sheer volume and variety of these data demands the deployment of scalable and flexible automatic methods leveraging compact and easy-to-process representations, capable to be adapted to different input text genres and characteristics and to the schema specifications of the given use case application.

Within the Natural Language Processing (NLP) field, several frameworks and techniques have been developed that generalize statistical patterns over linguistics features in the text data to extract abstract concepts such as named entities, topics, relations or events. One step further, Deep Learning (DL) architectures are able to discover complex patterns from large labeled data with no or minimal feature annotation, leveraging highly contextualized representations of the tokens in a text. More recently, the explosion of decoder-only pre-trained Large Language Model (LLM) architectures has enabled to solve high-level language understanding tasks with zero-shot or few-shot inference methods, sparing the need of collecting costly training datasets.

The data potential, though, can only be fully unlocked by coupling the above mentioned methods with Semantic Web (SW) techniques for the construction of full-fledged KGs out of domain data collections. KG representations are semantically transparent, explainable by design and machine-readable. Consequently, the knowledge extracted from a given text collection and consolidated using SW techniques can be retrieved and aggregated using semantically explicit predicates. Moreover, it is interoperable with and can update or expand existing knowledge repositories modeling the same domain, via linking of uniquely identified entities and predicates.

In this thesis, we experiment with the application of NLP, ML and Generative AI methods, powered by SW techniques and best practices, in three application domains and categories of text collections:

1. the analysis of DT discourse in the global news and SM platforms;

2. the mapping and trend analysis of the research in the Architecture, Engineering, Construction and Operations (AECO) domain from a large corpus of recent publications;

3. the construction of causal relation graphs from health records and patient-authored drug reviews.

1.2 Challenges

Unlocking the latent value of unstructured data for each of these use cases requires addressing several challenges, pertaining to the characteristics of the data itself and of the target domain.

Social media messages are typically short texts with little explicit context, using colloquial and often noisy language with context-dependent platform-specific expressions like hashtags. Scholarly publications contain technical discourse conventions, abbreviations, acronyms that are specific to a single domain and often ambiguous across domains. Clinical notes contain highly specialized jargon, frequent use of abbreviations for a large technical terminology of drugs, diseases, symptoms, dosages, etc. These text characteristics are challenging for standard NLP methods that need to be customized in order to extract entities and relations with acceptable recall.

Moreover, while standard named entities of type *Person*, *Location* or *Drug* have relatively lower variability in text, more abstract entities such as *Energy Efficiency*, categorized as instance of a general type *Task*, are far more ambiguous and hard to be merged to a canonical form and ultimately to link to entries in reference KBs.

Finally, as it will be shown, some application scenarios (like DT monitoring of Chapter 3) are better fit for a data-driven, unsupervised generalization approach, rather than to a prior specification of the target entity-predicate schema.

The main research questions addressed in this thesis, distributed across the three use cases, are:

Q1. How NLP and Semantic Web technologies can be combined to extract knowledge from noisy user-generated text collections and represent it in interoperable formats?

Q2. Which NLP and ML techniques better fit different application scenarios?

Q3. How can KG representations enhance the analysis of trends within a specific domain?

Q4. Can Generative AI techniques support the construction of scientific knowledge graphs of very abstract research concepts in technical domains, with only limited customization?

Q5. Which Generative AI techniques and LLM architecture and training methods are best suited to the generation of causal graphs from medical texts?

1.3 Contributions

This thesis makes contributions to the advancement of research on NLP and KG generation methods in terms of experimental results and of generated models, data resources and data analytics. The main contributions are:

- The design and evaluation of an open IE pipeline for KG extraction from microblogging and news articles text collections
- the release of KG data and visualization dashboards for the analysis of the DT discourse in news and social media
- the enhancement of an existing methodology for building research knowledge graphs from scholarly data and its customization to the AECO domain

- the release of a prototype data visualization dashboard for the analysis of AECO research trends
- A systematic benchmark evaluation of LLMs architectures, inference and learning techniques for a Causal Relation Extraction (CRE) task in the medical domain
- The release of a fine-tuned model for CRE and a demonstration of its applicability for building causal graphs of Drug-Adverse Drug Event (ADE) interactions

1.4 Outline

The thesis is organized as follows:

- In Chapter 2 we introduce the technical definitions of KGs, KG properties and SW techniques that we will be using throughout this thesis. Afterwords, we will briefly define the NLP, deep learning (DL) and enerative AI methods that are backing the whole KG extraction process. The aim of this chapter is not to provide a comprehensive survey of all techniques from the research literature but rather to offer a focused overview of the methodological options suitable for our particular use cases, thereby providing a rationale for the techniques presented in the subsequent chapters;
- Chapter 3 presents the use case application of NLP-based KG generation techniques to large collections of news and social media crowdsourced data for the monitoring of DT discourse;
- In Chapter 4 we describe how the integration of topic modeling, LLMs and SW techniques can be used to generate a large scale scientific KG of the AECO research landscape and thus support research trend analysis in this domain;
- Chapter 5 discusses the role of causality graph data in medical knowledge bases and pharmacovigilance and it benchmarks a large range of LLM architectures and methods for the generation of causal graphs of biomedical entities from clinical notes and drug reviews;
- Finally, Chapter 6 presents a discussion of the limitations of the proposed methods within the studied application domains along with an overview of the directions of our ongoing research on these topics.

Chapter 2

Knowledge Graphs Construction Methods

2.1 Knowledge Graphs for domain knowledge representation

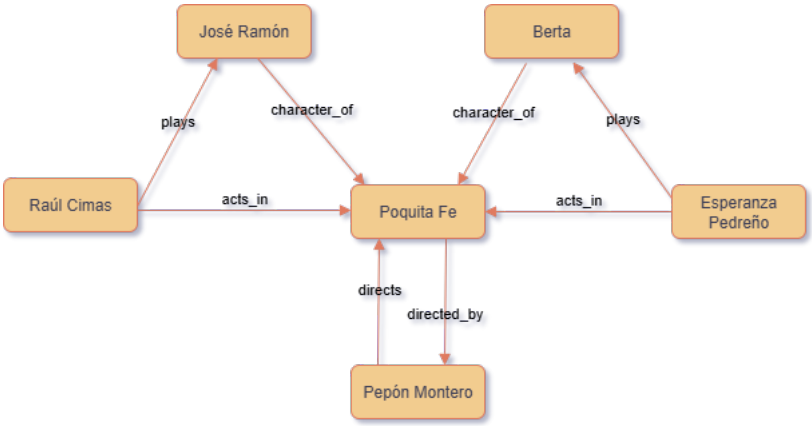
KGs are a very flexible formalism for representing data in a domain of interest as directed edge-labeled (DEL) graphs, where nodes represent domain objects and edges represent binary relationships between these objects. Formally, they are described by a tuple $G = (V, E, L)$ where V is a finite set of nodes, L a finite set of labels and $E \subseteq V \times L \times V$ is a set of edges. An edge is often conventionally represented as a triple (h, r, t), where $r \in L$ is a relation predicate connecting a "head" node h to a "tail" node t.

The formalism is expressive enough to enable encoding n-ary relations with $n > 2$ without the need to change the whole data schema, like in a relational database. For example, in the fragment of a sample KG of Spanish TV series in Figure 2.1, one can add an intermediate "character" node *Jose Ramón* to further qualify the binary *acts_in* relation between actor *Raúl Cimas* and the TV series *Poquita Fe*. Notice also that the basic formalism does not impose constraints on the graph topology, for example inverse relations connecting pairs of nodes (like *directs* and *directed_by* in Figure) can introduce cycles.

No matter how flexible and expressive, interoperability requires KGs to explicitly define the semantics of the nodes and relation labels. A minimal formalization of DEL graph semantics is achieved by using RDF/RDF Schema.

RDF RDF is a standardized data model that enforces restrictions on node/edge identifiers. Namely, nodes can be:

- Internationalised Resource Identifiers (IRIs), i.e. global persistent identifiers that can be looked up by web-servers to return RDF descriptions of the entity;
- literals, for representing strings and other XML Schema datatypes



a.

```
@prefix sdbo: <http://series-database/ontology#> .
@prefix sdb: <http://series-database/resource> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix dbr: <http://dbpedia.org/resource/> .
@prefix dbo: <http://dbpedia.org/ontology/> .
sdb:raúl_cimas rdf:type owl:NamedIndividual ;
     rdf:type sdbo:Actor ;
     sdbo:hasName "Raúl Cimas".
sdb:poquita_fe rdf:type owl:NamedIndividual ;
     owl:sameAs dbr:poquita_fe .
sdbo:Actor rdf:type owl:Class ;
     rdfs:subClassOf sdbo:Performer .
sdbo:acts_in rdf:type owl:ObjectProperty ;
     rdfs:subPropertyOf sdbo:performs_in ;
     rdfs:domain sdbo:Actor .
```

b.

Figure 2.1: Fragment of a TV series knowledge graph (a.), with corresponding RDF serialization in Turtle format (b.).

- blank nodes, anonymous nodes used for representing more complex data structures like lists, etc.

For example, in the RDF serialization of Figure 2.1.b, the node labeled as *Raúl Cimas* is identified by an IRI like *http://series-database/resource/raúl_cimas*, assuming *http://series-database/resource* is the namespace of our TV series KG, and it is connected to the literal "Raúl Cimas" via a data property (that is, a string valued predicate) *hasName*. The unique namespace *http://series-database/resource* can be prefixed (like in *sdb:raúl_cimas*) and prevents name clashing with resources in other KGs modeling the same domain entities.

RDFS RDF Schema is a metalanguage making available predefined predicates that allow a partial definition of the semantics of the terms in a RDF KG (what is often

called the KG's ontology). For example, one might further specify the information in the sample KG of Figure 2.1 by making explicit that the resource *sdb:raúl_cimas* is an instance of class *sdbo:Actor* defined by the KG ontology, which on its turn is a subclass (*rdfs:subClassOf*) of the *sdbo:Performer* class. Moreover, *sdbo:acts_in* is an *owl:ObjectProperty* and a subproperty of *sdbo:performs_in*, etc. Additional constraints can be enforced by using the full range of RDFS predicates or defining a set of rules using the more expressive OWL ontology language.

Bridging A major mechanism for knowledge integration in the Semantic Web is bridging. It consists of referencing external resources (so called named graphs) in the namespace declarations of a KG and then using the OWL predicate *owl:sameAs* to state equality of individuals, like in Figure 2.1 where the TV series *sdb:poquita_fe* is stated to be equivalent to the DBpedia entity *dbr:poquita_fe*.

In this way, one can access heterogeneous information about the same individuals as encoded in different KGs. This opens up a vast analytical potential when doing knowledge retrieval from KGs, as SPARQL queries can get evaluated against the union of a default RDF and the additional set of named graphs.

DBpedia With many KGs linked to it and by bridging to several external open resources, DBpedia is one of the most central knowledge hubs in the Linked Open Data ecosystem.

It is a large scale, open domain and multilingual graph that has been collectively developed under the open data philosphy [LIJ+15]. It is constantly expanded using an information extraction framework that parses Wikipedia infoboxes to identify property-value pairs and align them with a community-maintained ontology. This guarantees that DBpedia remains up-to-date to the evolving structure of Wikipedia. As of release 3.8 (2015) DBpedia features 3.77M entities just for English, but recent snapshots counted over 850M triples and a total of nearly 1.9B RDF statements.

We perform bridging to DBpedia for the KGs described in Chapters 3 and 4.

Density Measures KGs can be characterized by diverse density measures (or equivalently, inverse sparsity measures), which have an impact on the effectiveness of the graph generation techniques discussed in the subsequent sections [KDTZ26].

Graph-theoretic density measures how many edges are present in a graph $G = (V, E, L)$ relative to the maximum possible:

$$Density(G) = \frac{|E|}{|V| \cdot (|V| - 1)} \tag{2.1}$$

It is a raw density measure as KGs are typically extremely sparse in this respect, because not every pair of entities should be linked, depending on the relation label.

Relation Type Instantiation defines how much of the potential space of valid triples for a relation type is instantiated in the graph. That is, if $Dom(r)$ and $Range(r)$ are

respectively the domain and range of a relation label r in G,

$$Density(r) = \frac{|\{(h,r,t) \in G\}|}{|Dom(r)| \cdot |Range(r)|} \tag{2.2}$$

Density per Entity measures the average degree (number of edges) per entity V (or per entity type) in a graph:

$$AvgDeg(V) = \frac{|2E|}{|V|} \tag{2.3}$$

Finally, *Schema Coverage* (also called ontology usage density), measures the fraction of schema-defined relations that appear at least once in the KG. This is useful to quantity coverage imbalance, that is how much unequally the KG instantiates the relations in the ontology, and semantic density, that is how fully the ontology's expressive capacity is realized in the KG facts.

2.2 Generating Knowledge Graphs

While KG are a semantically transparent representation of structured data and support effective methods for retrieving and inferring knowledge from them, their symbolic nature makes them hard to be manipulatet at scale.

Therefore, recent years have seen the emergence of inductive knowledge discovery methods for KGs that are based on compact representations of the graphs on low-dimensional vector spaces, known as graph embeddings, that preserve the latent structure of the graph and at the same time can be used my standard ML algorithms for inducing new knowledge.

The general idea of graph embeddings is to apply a self-supervised approach where, after initializing a random embedding vector representation of each entity $e \in \mathbb{E}$ and relation $r \in \mathbb{R}$, a representation for each triple (h,r,t) in the graph (e.g. $(AlfredHitchcock,\ DirectorOf,\ Psycho)$) is learned by optimizing on a scoring function that maximizes the plausibility of the set $\mathbb{D}^+ = \{(h,r,t)\}$ of all the facts in the graph (positive triples) and minimizes the plausibility of (synthetically generated) negative triples.

In its basic version (translational models such as TransE [LLS+15]), graph embedding methods represent both entities and relations in the same d-dimensional vector space $\mathbb{R}^d$ and interpret relations as translation vectors **r** connecting the vectors **h** and **t**, such that if (h,r,t) holds, $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$ [WMWG17]. Here, the underlying intuition, taken from the paradigm of distributional semantics [MCCD13], is that a vector representation, say for the relation $DirectorOf$, must guarantee that both $JamesCameron + DirectorOf \approx Avatar$ and $AlfredHitchcock + DirectorOf \approx Psycho$ hold. Therefore, the scoring function is defined as the negative distance between $\mathbf{h} + \mathbf{r}$ and t, that is:

$$f_r(h,t) = -\|\mathbf{h} + \mathbf{r} - \mathbf{t}\|_{1/2} \tag{2.4}$$

Other variants of graph embeddings have been proposed to solve some of the shortcomings of translational models, such as for example semantic matching models that use scoring functions matching the latent semantic components of entity and relation vectors (tensor decomposition models [RSG17]), or RDF2Vec [RP16], that linearize a graph as a set of sentences by randomly traversing it and collecting the visited paths and finally feed the collected sentences to a standard language embedding learning algorithm such as word2vec [MCCD13].

Instead of creating dense vector representations of graphs to be fed standard ML algorithms, Graph Neural Networks (GNN) are ANN architectures that mirror the topology of the graph data [WJL+22], with network nodes representing graph nodes and node connections representing graph edges.

The general idea here is that each node and edge in the graph is associated with a feature vector (one-hot representation or based on node attributes), while an embedding vector h_v (or *state vector*) for a node v is learned by iteratively applying for several layers k a "message passing" function:

$$h_v^{(k+1)} = f(W \cdot AGG\{h_u^{(k)} : u \in N(v) \cup \{v\}\}) \tag{2.5}$$

where $N(v)$ is the set of neighbors of node v, AGG is an aggregation function and W is a learnable weight matrix. In practice, a GNN updates node embeddings for all its nodes by aggregating information from their neighbors in the graph. Then another function g is used to compute an output value for a node v based on its embedding vector h_v, its feature vector n_v and a weight matrix W', that is:

$$o_v^k = g(W', h_v^k, n_v) \tag{2.6}$$

The idea here is that feature vectors remain fixed across the learning process, while the desired output values are given only for a subset of supervised nodes of the graph, which represent the training set of the process. The framework is able to learn the parameters W and W' of the two functions that generate the expected output, and the output function g is then applied to other nodes of the graph.

What all the presented frameworks share is the capability to leverage the latent properties of an initial KG nucleus for downstream tasks of KG completion. Typical graph completion tasks involve link prediction, where a missing edge between two existing nodes is predicted based on characteristics of the involved nodes and their connectivity patterns with other nodes in the graph [CP17].

For example, for a task of predicting which entity head h is connected via a relation r to an entity tail t (i.e. $(?, r, t)$) one can take every candidate entity h' in the graph, compute the corresponding score $f_r(h', t) = -\|\mathbf{h'} + \mathbf{r} - \mathbf{t}\|_{1/2}$ based on the learned embeddings and scoring function (in this case, using a translational model), ranking the scores in descending order and finally add predicted triples based on some heuristic thresholds on these ranked scored.

However, overall these techniques best fit use case scenarios where:

- an existing nucleus of graph-encoded knowledge is already available and at the same time there is scarcity of other sources of information from where additional knowledge could be derived;
- a set of relation labels is already defined in a KG ontology;
- the target graph that is to be built has a significant level of expected relation instantiation and entity density, as defined in Section 2.1;
- there are no strong requirement for explainability or source traceability of the automatically-induced knowledge [Mol25]

However, the use cases we discuss in this study exhibit traits that, to varying extents, fail to comply with one or more of the conditions listed above.

In all three use case scenarios, a rich and up-to-date source of information is available in the form of a large collection of unstructured text from which the target KG is to be generated from scratch. The nucleus of pre-existing knowledge about the nodes of the target graph is not necessarily null: for example, in the DT monitoring use case in Chapter 3 the entities we target are described in a number of reference KGs we eventually link to. However, the set of relations we aim to derive are meant to be an extension to the static body of relations already encoded in those reference KGs, so that few evidence on these new relation types is initially available. Moreover, in some use cases (such as the DT monitoring) the set of relation labels is simply not given *a priori* and instead is being generalized from the relation instances extracted from the text collection. Overall, most of the target relations we aim to induce are very sparse, therefore graph embeddings would struggle due to lack of training signal or overfit to the few represented relation types.

Finally, for sensitive domain use cases like the causal relation graph of drug and ADE entities introduced in Section 5, the acquired relation instances might need to be explainable or at least traced back to the source document which generated them, while an holistic induction from global patterns of the graph without human-interpretable justification might not be acceptable.

For these reasons, in this thesis we make use of and describe exclusively extractive KG construction techniques that generate graph nodes and edges via the application of NLP/DL methods to large collections of unstructured data, i.e. text corpora.

2.3 Knowledge Graphs Construction Process

The overall task of extracting a KG from a collection of unstructured text documents can be factorized into two main sub-tasks [HMRHLA20, RvEV+16], namely:

1. Entity Extraction and Linking (EEL), which consists of detecting in the input text occurrences (mentions) of named entities and associating them with existing unique nodes in an external Knowledge Base (e.g. via DBpedia IRI identifiers), or

creating novel candidate identifiers in the KB (also called emerging entities). The detected entities represent the nodes of the generated graph and can be assigned a type (typically via *rdf:type* property) contextually with the extraction process or by inheriting from the linked identifier in an external KB.

2. Relation Extraction and Linking (REL), which consists of extracting n-ary relations (with $n \geq 2$[1]) between detected entities in the input text, linking the relation predicates to (object or data) properties in a reference ontology or to a newly defined relation predicate[2]. When both the relation predicate and entities get linked to a KB or when identifiers are created for them, the result of the EEL and REL process can be used to populate the KB with additional facts.

The example in Figure 2.2 illustrates the input and (partial) output of the combination of the two tasks, where four entities (*Raúl Cimas*, *Jose Ramón*, *Poquita Fe* and *Spain*) are detected and mapped to their corresponding DBpedia resources (namespace prefix *dbr*), and five relation triples are extracted (e.g. $(Raul\_Cimas, Occupation, Actor)$) and their predicates mapped to DBpedia object properties (e.g. *dbo:occupation*).

```
Input:
Raúl Cimas is a Spanish actor, who is known for interpreting Jose
Ramón in the comedy series Poquita Fe.
```

```
Output:
dbr:Raúl_Cimas dbo:occupation dbr:Actor ;
     dbo:birthPlace dbr:Spain .
dbr:Jose_Ramón(Poquita_Fe) dbo:portrayer dbr:Raúl_Cimas ;
     dbo:series dbr:Poquita_Fe .
dbr:Poquita_Fe dbo:genre dbr:Comedy .
```

Figure 2.2: Example of applying Entity Extraction and Linking and Relation Extraction to an input text, using DBPedia as a reference KB.

Entity Extraction and Linking Although the EEL task could be directly accomplished by creating a dictionary of entity labels out of the set of nodes of the reference KB and then applying some form of lexical lookup across the input text collection using this dictionary, this simplistic approach typically suffers from low recall, not to mention that it does not support by definition the extraction of emerging entities that are currently missing in the KB.

[1] In all the use cases described in this thesis, we will only deal with binary relations.

[2] As we will describe later on, this latter option characterizes the Open Information Extraction paradigm.

Therefore, a standard approach is to split the EEL task into a Named Entity Recognition (NER) step that locates mentions of (generic or typed) entities based on contextual information from the input text where they occur, and an Entity Linking step that leverages the reference KB graph information to map entities to it [AMGOLOV20]. Methods here vary significantly with respect to the type of information from entity mentions and KB node candidates they use. Typically, each mention-KB node candidate pair is scored based on string similarity between mention and candidate labels, or keyword-based similarity between the context of the mentions and a context from the reference corpus mapped to the KB (e.g. Wikipedia for DBpedia), or keyword-based similarity with the context of the nodes connected to the candidate node in the KB graph, etc. [DJHM13].

Relation Extraction When a reference KB covering the target entities and relations is available, and once the EEL task is solved, a popular REL technique is "distant supervision" [HZL$^+$11]. This is based on the hypothesis that if a given triple (h, r, t) (say for example: *(dbr:Poquita_Fe,dbo:genre,dbo:Comedy*) is found in the KB, the text of a sentence where the triple's entities are mentioned would also contain a mention of the target relation (e.g. "one of the most innovative comedy series was Poquita Fé"). In this way one can collect a training dataset by heuristically matching the entities of the KB triple set, generalize some linguistic patterns for each target relation predicate, and then apply the pattern to new matched entity pairs, expanding the KB with new triples.

However, for those application scenarios, including the use cases dealt with in this thesis, where context knowledge provided by a pre-existing KBs is null or extremely sparse, the KG construction process is entirely carried out by standard NLP or DL enabling technologies processing solely the input data stream, prior and independently of the KB linking phase. In the remaining sections of this Chapter we outline the main families of such methods and architectures, along with their advantages and disadvantages.

2.4 Natural Language Processing Methods

By NLP methods, we refer here to approaches that solve the Named Entity Recognition and Relation Extraction tasks by using, as part of their input, language structures generated by NLP processing of the raw text. This encompasses both rule-based methods, that build heuristic rules over NLP-generated structures, or learning-based approaches that apply some standard ML techniques over features generated by NLP processing.

2.4.1 Dependency Parsing

One of the central building blocks of NLP-based NER/RE pipelines is dependency parsing. Dependency parsing represents the syntactic structure of a sentence solely in terms of its words (lemmas) and a set of binary grammatical relations connecting pairs of words.

Grammatical relations are directed, typed edges from a *head* to its *dependent*. For example, in the parse tree in Figure 2.3 the main verb lemma "interpreted" is the head of a *dobj* (direct object) relation to the lemma "Ramón" (black edge). An inventory of labels for these dependency relations that are valid cross-linguistically is provided by the Universal Dependency set [dMDS+14].

Therefore, we can formally define a dependency tree as a directed, labeled multi-relational graph $G = (\varepsilon, \mathcal{R}, \mathcal{T})$ with ε the set of lemmas in the vocabulary plus a *root* node marking the head of the sentence dependency tree, $\mathcal{R}$ the set of Universal Dependency labels, and where the additional constraints hold that: 1) there is a single designated *root* node that has no incoming edge; 2) with the exception of the *root* node, each node has exactly one incoming edge and 3) there is a unique path from the *root* node to each node in ε.

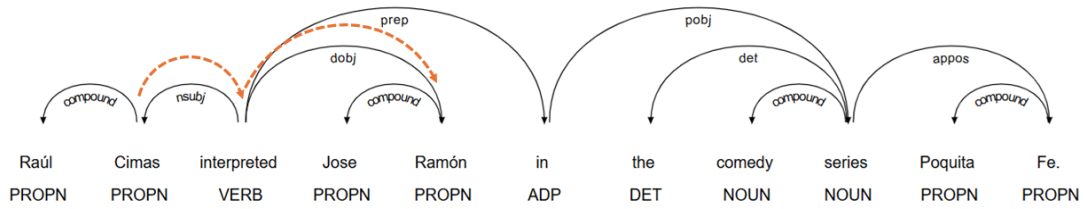


Figure 2.3: Dependency tree of an example sentence (root node omitted).

In Figure 2.3 we also mark in orange the Shortest Dependency Path (SDP, [BM05]) between the two lemmas "Cimas" and "Ramón". We can describe those shortest paths by collecting the edge labels during path transversal. For example, in this case the path connecting the two target lemmas is of type $(nsubj, dobj)$. In Section 3.5.2 we will apply constraints on the labels of the paths connecting entities and containing verb lemmas, in order to extract candidate relations.

2.4.2 Open Information Extraction

As dependency parsing algorithms are relatively efficient, dependency trees are the standard input representation for a family of RE methods known as Open Information Extraction.

Open IE is an highly scalable framework that has been succesfully deployed to extract millions of triples from large web corpora. It best fits use cases when a curated relation schema definition (and corresponding manually annotated data) is not provided and one aims to "discover" the most significant relation types emerging within a target text collection (similarly to the use case of Chapter 3). The selection of the extracted triples is purely syntactical, and its soundness is guaranteed by imposing a set of constraints on the dependency trees connecting entity tokens. These constraints can be in the form of handcrafted rules like in the earlier systems, or they can be patterns learned by distant supervision.

For example, [AJPM15] uses a two-stage process that: 1.transforms longer sentences into self-contained clauses using a multinomial logistic regression classifier; 2. applies logical inferences to reduce clauses to maximally compact but semantically equivalent sentences, parsing them into simple subject-verb-object triples. A triple extraction example is illustrated in Figure 2.4.

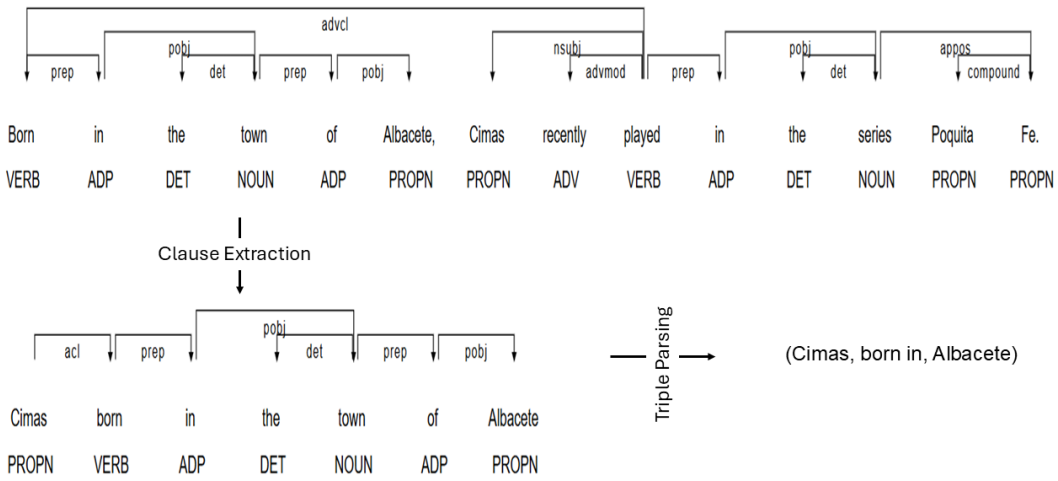


Figure 2.4: An example of sentence to triple transformation using Open IE.

One of the advantages of Open IE approaches is that they work reasonably well across domains. However, they have the limitation of generating surface-level triples that are not *per se* unified into more abstract predicates that can be later used for triple retrieval. If a reference KB is given, one can try and map relations onto the KB predicates. Otherwise, an unsupervised solution consists of clustering the extracted relation expressions.

2.4.3 Relation Clustering

Given a collection of relation instances $R = (r_1, ..., r_n)$ the goal of relation clustering is to segment the collection into partitions $C = (C_1, ..., C_m)$, with $m < n$ such that relations within the same partition are more similar than instances across partitions, for some measure of semantic similarity. In cases when the number of resulting partitions cannot be anticipated (like for relation clustering), it is common to resort to *hierarchical clustering* algorithms. One method that is often applied in NLP is HDBSCAN.

HDBSCAN Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN, [MB20]) is the hierarchical version of the popular density-based DBSCAN,

which is characterized by grouping together points in highly dense regions of the representation space and taking as outliers (therefore leaving unclustered) the data points lying in low-density regions.

However, HDBSCAN additionally builds a hierarchy of clusters (a dendrogram) over varying density thresholds, by recursively merging smaller clusters of points that are adjacent to each other. The resulting, flat clustering is the most stable one across varying scales.

A standard procedure to make words processable by clustering algorithms is to represent them via pre-trained *word embeddings*, which are distributed representations as dense multidimensional vectors, learned using self-supervised word prediction tasks from large unannotated text corpora, and that capture some features of word meaning. Word embeddings vary as for the language objects they model (from sub-tokens to n-grams), the ANN architecture and the cost function they use for training, and finally the size of the generated vectors representations. Here we will only mention two popular embedding models that we will be using in the following chapters.

GloVe Global Vectors for Word Representation (GloVe, [PSM14]) is an unsupervised algorithm that learns 300-dimensional static embeddings aggregating global statistics for word types (that is for predicate lemmas like "interprets") across all its occurrence contexts in the training corpus (e.g. Wikipedia). Namely, for each lemma it learns a weight matrix that maximizes its co-occurrence probability with lemmas that appear in a text window of given size to the left or right of the lemma, in the reference corpus. The embeddings are called static as, once a lemma is assigned a unique embedding vector at learning time, then at inference time every occurrence of that lemma uses the same embedding, regardless of the context where it appears.

BERT The Bidirectional Encoder Representations from Transformers ([DCLT19]) is the groundbreaking DL architecture that introduced the *self-attention* mechanism [VSP+17], by which each word's representation is updated by attending to other words in the sequence. For a word w_i in a context (e.g. sentence), BERT embeddings are computed as the sum of embeddings of all other words w_j in the context, weighted by the attention weight matrix controlling how much w_j influences w_i. The attention weights are learned by optimizing on self-supervised Masked Language Modeling tasks over large corpora. The resulting embeddings are contextual, as each word's vector representation is dynamically computed from its surrounding words, rather than being a fixed lookup vector.

Dimensionality Reduction It is widely recognized as the so called "curse of dimensionality" problem affects the analysis of high-dimensional data. Specifically, in our case high-dimensional word embedding relation representations require more observed samples to produce a suitable level of density for HDBSCAN to work properly. However, applying UMAP to perform non-linear, manifold-aware dimension reduction [MHM20] has been

proven to transform the datasets down to a dimension small enough for HDBSCAN to cluster a large majority of instances.

The UMAP-HDBSCAN combination is controlled by a number of hyper-parameters[3] [4], the main ones are described in Table 1 in the Appendix A, together with some common sample values. In Sections 3.7 and 4.4 we will perform hyperparameter grid search for optimizing the UMAP-HDBSCAN combination for relation clustering and topic modeling, respectively.

2.5 Deep Learning Methods

Instead of learning functions over an engineered set of features extracted from the input sentences, DL models start with some (learnable or pre-trained and fixed) vector representation of the sentence tokens and incrementally learn the weight parameter matrices of complex ANN architectures that allow mapping token sequences onto output relational triples.

These methods largely vary as regards the input text representations they use and the network architectures deployed to encode the dependencies within sentence components. Moreover, the flexibility of ANN architectures enables the creation of both pipeline frameworks - which sequentially perform entity pair detection followed by relation classification using the detected pairs - and joint RE frameworks, where the NER and RE tasks are simultaneously solved by the same learning architecture [ZDY+24].

A survey of the vast variety of DL models for NER/RE is out of the scope of this chapter. We will rather outline here a few classical architectures, some of which will be referenced in the next chapters.

2.5.1 BiLSTM for NER

Named entities are sequences of tokens and the decision whether a token belongs to an entity depends jointly on the features of the token itself and the surrounding ones. Consequently entity detection can be formalized as the task of assigning a label from a tagging schema to each token in an input sentence. In the example below we show an input sentence with on top the expected entity tagging labels, using the IOBES schema[5] [RR09], stating for instance that the token "Raúl" begins a PERSON entity sequence, "Cimas" is inside a PERSON entity, "plays" is outside any entity sequence, etc.

	B-PER	I-PER	O	B-CHAR	I-CHAR	O	O	O	O	B-MOVIE	I-MOVIE	
(1)	*Raúl*	*Cimas*	*plays*	*Jose*	*Ramón*	*in*	*the*	*TV*	*series*	*Poquita*	*Fe*	.

Intuitively, sequential ANN models like Bidirectional Long Short-Term Memory

[3] https://umap-learn.readthedocs.io/en/latest/parameters.html

[4] https://hdbscan.readthedocs.io/en/latest/parameter_selection.html

[5] IOBES stands for Insides, Outsides, Beginnings, Ends and Single-tokens.

(BiLSTM) are best suited to carry out this task. BiLSTM networks are extensions of Recurrent Neural Networks featuring: a layer of hidden units h_t that get activated by input element x_t and by input from unit h_{t-1} preceding h_t (recurrently passing left-context to each activation unit); an equal layer running in opposite direction, recurrently passing right context; gating mechanisms that allow to control how far elements in the sequence affect the activation of each hidden unit, enabling to model long-range linguistic dependencies in a sentence.

Leaving aside details, Figure 2.5 outlines a popular architecture stacking a word embedding vector representation with a BiLSTM layer (in orange) and a sequential conditional random field (CRF, [SM12]) for generating a sequence of IOBES tags from the input sentence (1) above [LBS+16]. The BiLSTM is fed a sequence of vectors x_i for each token and builds a representation c_i by concatenating the learned left- and right-context representations $\overrightarrow{h}_i$ and $\overleftarrow{h}_i$. The matrix P of size $n \times k$ (where k is the size of the IOBES tag vocabulary) of scores output by BiLSTM for the sequence of size n is finally modeled jointly with a matrix of tag transition scores via CRF and the log-probability of the correct tag sequence is maximized during training, generating the tag sequence shown in the top layer in Figure.

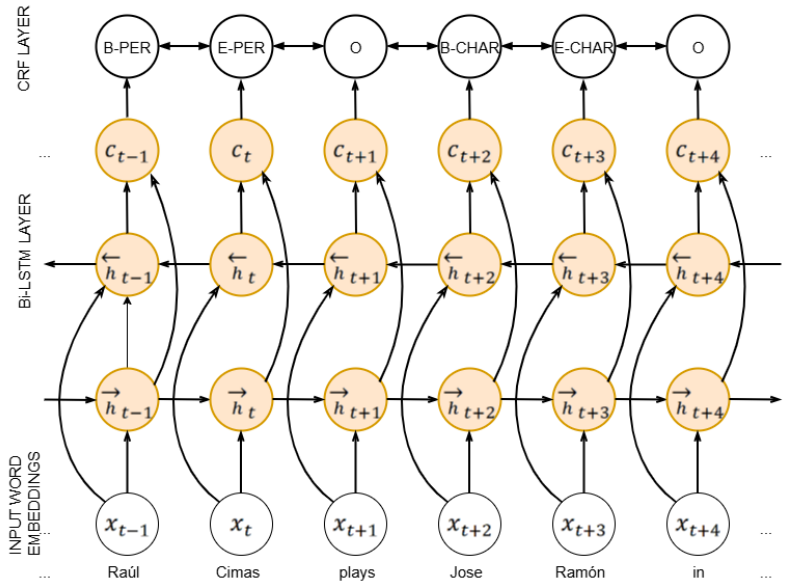


Figure 2.5: A sample BiLSTM-CRF architecture for NER. Adapted from [LBS+16].

2.5.2 CNNs for Relation Classification

In a pipeline approach, once the pair of target entities have been already detected, the RE task boils down to a n-class classification problem, with n being the number of possible relation labels.

Instead of relying on a set of lexical and syntactical features extracted by NLP pre-processing modules (typically suffering from error propagation from the deployed tools), [ZLL+14] propose to directly feed pre-trained word embedding vectors of the input sentence tokens to a Convolutional Neural Network (CNN) that learns a deep representation of sentence level features of the context of the input entities. As shown in Figure 2.6, the lexical feature vector is constructed by simply concatenating the embedding vectors of the entity-marked tokens (shown in green and blue at the bottom of the Figure) and of their left- and right-context tokens[6]. Instead, sentence level features are learned by a convolution and max pooling layers, followed by a non-linear layer with $tanh$ activation. The input to this sub-network is computed by a Window Processing module that, for each of the n tokens in the sentence, combines a window of the w embedding vectors around the token with a position feature encoding information on token distance from the two entity tokens. The output of the CNN module is then passed through a non-linear tanh layer and forms a compact representation of the most significant dependencies among the tokens in the sentence.

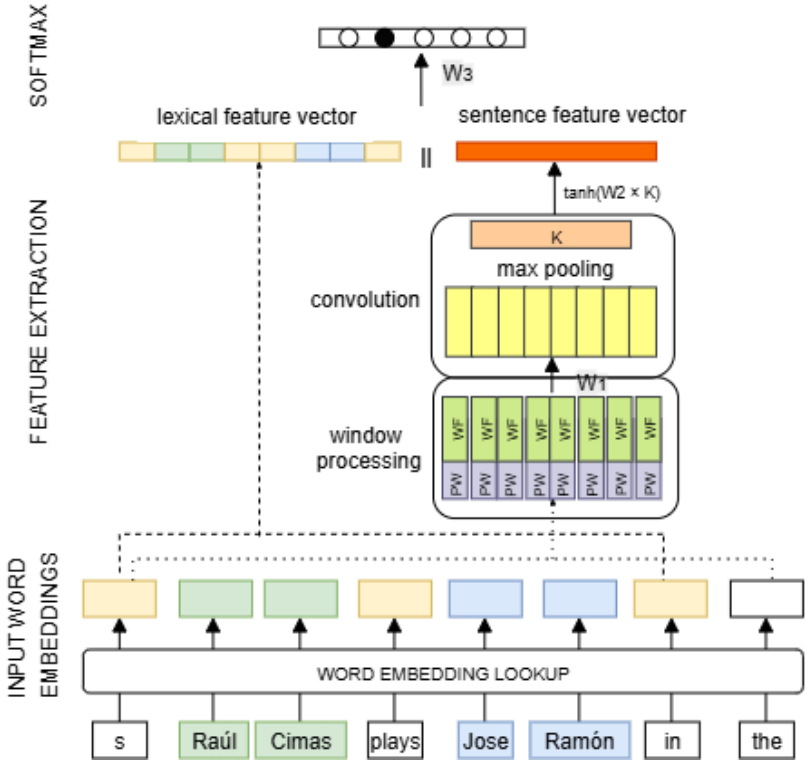


Figure 2.6: A CNN architecture for relation classification of entity pairs in a sentence. Adapted from [ZLL+14].

The lexical and sentence feature vectors are then concatenated into a final feature vector which is finally passed to a softmax layer for predicting the most likely relation

[6]Here a size one context window is illustrated for simplicity.

label.

2.5.3 Joint NER-RE models

Entity detection and relation classification may benefit from exploiting interrelated signals, for example in the sentence in Figure 2.6 the fact that "Cimas" is of type *actor* is relevant to the decision whether the predicate "plays" is an instance of an *acts_in* relation type, and vice versa. Therefore, recent DL architectures have proven successful at solving the two tasks jointly.

As a notable example, SpERT(Span-based Entity and Relation Transformer) [EU20], is a model that reached SOTA performances for some of the RE tasks we will deal with in this thesis. It assumes a span-based approach that exhaustively searches every token subsequence of an input text for candidate entities, thus allowing overlapping or nested entities (differently from the IOBES tagging paradigm).

The relatively simple SpERT's architecture is summarized Figure 2.7.

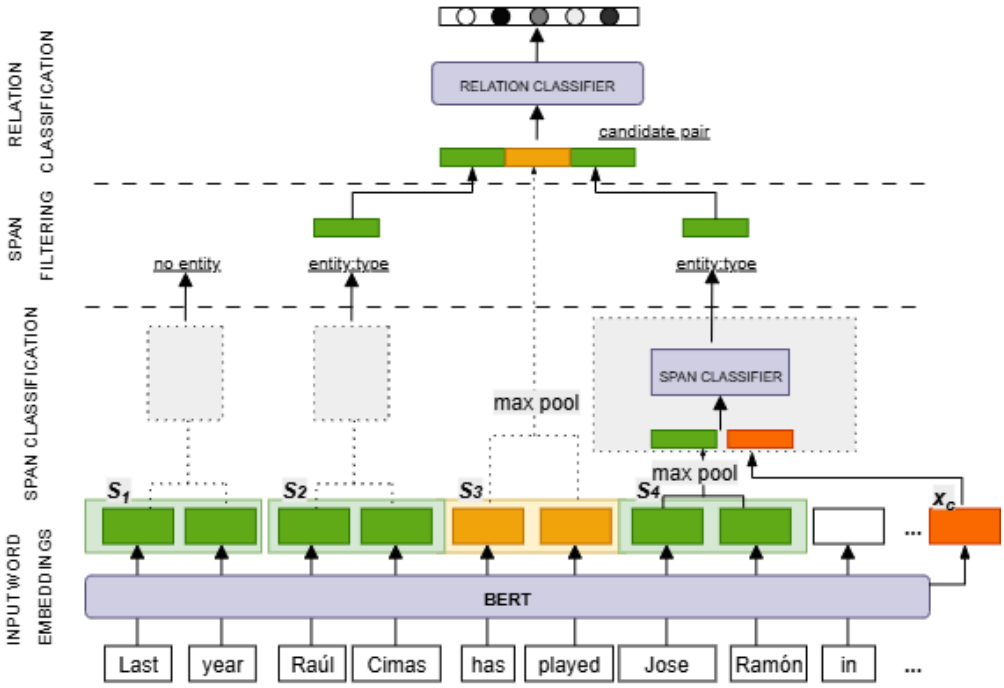


Figure 2.7: A span-based architecture for joint NER-RE. Adapted from [EU20].

It leverages pre-trained BERT embeddings for representing tokens in the input sentence (plus an additional token c encoding global information about the sentence)[7]. For each span s_i, it fuses its token vectors with max pooling, concatenates them with a

[7]The BERT embeddings are tuned during the training of the network.

context vector x_c capturing global sentence information[8] and classifies its type with a softmax layer, filtering non-entities (i.e. entities of class *None*). Finally, it classifies the relation holding between all pairs of remaining entities by using a concatenation of the entity-fused BERT embeddings (s_2 and s_4 in Figure) plus the max-pooled fused embeddings of the token span between the entities (marked in orange in Figure) and applying a single layer sigmoid classifier, such that every relation with an activation score above a confidence score is returned, otherwise no relation is output for the entity pair.

One major limitation of DL methods is that their performance level depends on the availability of large annotated training sets. Such resources have been built over the years for a number of research competitions, for example SemEval-2010 Task 8 [HKK+10], TACRED [ZZC+17] and DOCRed [YYL+19]. However, they tend to be general or encyclopedic corpora focusing on a limited range of general entity and relation types (e.g. *Person*, *Location*, *Organization*, etc.), languages, and language styles (e.g. English news), which might not fit the specification of one's own application[9].

The recent scaling of the Transformer architecture has led to the explosion of Large Language Models (LLMs), which encode extensive general linguistic and world knowledge via pre-training over web scale unannotated text collections, making them robust to unseen domains. Therefore, LLMs represent a powerful solution to the KG construction task, particularly in the low-resource settings we discuss in this thesis.

2.6 LLM-based Methods

We briefly discussed *self-attention* in Section 2.5. Combining a multi-head self-attention with a feed-forward layer makes what is usually referred to as a *transformer block*, which forms the backbone of LLM architectures. However, it is necessary to distinguish between *encoder* blocks, which can process the entire input sequence in parallel using *bidirectional self-attention* and are used to generate contextual representations of text, and *decoder* blocks, generating text token by token, using masked self-attention over previously generated tokens only.

Various architectures have been proposed that build upon these processing blocks, all of which strictly speaking fall under the definition of LLMs. For example, the encoder-only model BERT, or the encoder-decoder model T5 [CHL+24].

However, in the rest of this thesis, we will use the term LLM to refer to *generative*, decoder-only LLMs (originally named Generative Pre-trained Transformer, or GPT [RN18]), i.e. models that omit encoder blocks and consist solely of a stack of decoder layers incorporating masked self-attention and feed-forward sub-layers. Rather than processing the input holistically like encoder-decoder models, these models take a token sequence as input and generate a maximum likelihood output token sequence auto-regressively, that is one token at the time and conditioning also on previously predicted

[8] Plus a learned embedding encoding span width, not shown in Figure for simplicity.

[9] Distant supervision has been an explored solution to this issue [MBSJ09].

tokens[10].

While sacrificing the rich bidirectional understanding of inputs provided by encoder-decoder models, this design choice reduces the number of learnable parameters making generative models extremely scalable for extensive training on sequence generation tasks.

LLMs usually follow a two-step training paradigm:

- during initial *language modeling* phase, they undergo a pre-training via self-supervised language masking or word prediction tasks on massive, trillion-token scale text corpora, generating general-purpose *foundational* models, such as Mistral [JSM+23a], LLaMa 3 [TMS+23a, HMQ+24], Gemma [TMH+24], and GPT 4.0 [Ope23];
- these base models can be further specialized to downstream tasks either by prompting techniques or via supervised fine-tuning over (typically much smaller) training sets. For example, they can be trained to follow instructions by being presented with pairs of question-response data[11]. In this latter case, the model's weight parameter matrix is (partially) updated.

Both foundational and instruction-tuned LLMs have shown strong capacity to carry out standard NLP tasks with near-SOTA performance levels via *in-context learning*, that is by being exposed only to natural language instructions for the target task and optionally a few task solution examples [BMR+20]. The instructions used to query an LLM are commonly called *prompts* and prompt design is known to significantly affect LLM performance, depending on its ability to elicit the LLM's vast pre-training knowledge for the new task [LYF+23a]. In the remaining sections we will briefly introduce some popular prompt engineering techniques, which will be applied in the benchmark evaluation of Chapter 5.

Aside from prompt design, LLM output is controlled by a few inference parameters which determine how the model's next token prediction probabilities are applied. They are listed in Table 4 in Appendix D.

2.6.1 Instruction Prompting

Regardless of the target task, a prompt features five basic components:

1. **Role** Prompts of modern LLMs are encoded as sequences of messages, where each message has an assigned role. Role labels are system dependent and reflect the dataset formatting of the specific LLM's instruction tuning, but common role labels are: *system*, for setting background instructions; *user*, encoding the human request and task instruction; *assistant*, for describing the model's prior or expected output. This is where usually few-shot examples are included.

[10] Maximun probability is in fact only one of various decoding strategies for picking up a token at each generation step.

[11] Note that a third standard training phase, called preference tuning, is out of the scope of this thesis.

2. **Instructions** This contains the task description;
3. **Data** or input context, that is the actual content on which the task should be applied;
4. **Output format**, enforcing constraints on how the model's answer should look like;
5. **Examples** (optional): in few-shot prompts, these are input–output sample pairs that illustrate the task.

Figure 2.8 shows a minimal instruction prompt for RE which directly asks LLMs to extract relation triples from text.

Instruction Prompt

```
SYSTEM:  You are an Information Extraction assistant.

USER:    Given the possible relations:
         ["acts in", "directs", "directed by", "featuring" ]
         What are the relations between the subject entity and the
         object entity expressed by the sentence?
         Provide the output as a triple (head, relation, tail)
         Sentence: Raúl Cimas is a Spanish actor, who played Jose
         Ramón in the comedy series Poquita Fe.
         Subject: Raúl Cimas
         Object: Poquita Fe
         Triple:
```

Figure 2.8: A baseline instruction prompt for RE.

In Section 5.5.2 we analyze empirically how these five and additional standard prompt design components and features impact LLM performance on a Causal RE task.

2.6.2 Few-Shot Learning

In few-shot learning [BMR+20], the LLM is provided at inference time with a prompt with the following components:

- *Task Instruction*: A description of the task to be solved.
- K *Examples of Context-Completion Pairs*, with K typically ranging from 1 to a dozen, depending on the model token size limit.
- *Input Context*: The specific context for which the model is expected to generate a completion.

The context-completion pairs act as a form of conditioning, enabling the pre-trained model to leverage its knowledge for a new task without updating any model parameters [RWC+19].

2.6.3 Prompt Chaining

Prompt chaining is a technique that breaks down the instruction prompt of a complex task into a recursive sequence of simpler sub-prompts. Each sub-prompt in the sequence takes as input the outputs of previous sub-prompts in the chain [SYC+24]. This approach is motivated by the well-documented observation that single instruction prompts often perform poorly for RE tasks. This is because they require LLMs to handle three non-trivial reasoning processes in a single step: i) Extracting the semantic relationship between the subject and object entities in the text, ii) understanding the semantics of the relation labels, and iii) matching the extracted relationship semantics to the appropriate relation labels [JGMW+22, LWK23].

2.6.4 Chain-of-Thought

Similar to prompt chaining, Chain-of-Thought (CoT) [KGR+23] is a prompt engineering technique that has proven effective in eliciting structured reasoning from LLMs for solving complex problems. Unlike prompt chaining, which involves multiple generations and passing intermediate results, CoT guides the model through the reasoning steps (or "thoughts") in a single prompt. This can be achieved either by providing reasoning examples (few-shot CoT) or by explicitly instructing the model to reason step-by-step (zero-shot CoT).

2.6.5 Instruction Fine-Tuning

Pre-trained, foundational models typically excel at text completion tasks, but are not necessarily able to follow instructions for an user-defined task. A potential solution would be to apply standard backpropagation to optimize the entire weight matrix of the model to the desired task, training it on a (typically much smaller) labeled dataset of instruction-response pairs. For example, for the base RE instruction prompt of Figure 2.8, one such pair might be encoded in a template like the one shown in Figure 2.9.

However, this full fine-tuning approach is rarely applied in low-resource settings for its heavy computational cost and because it has a "catastrofic forgetting" side-effect, such that the model loses much of its previously acquired general knowledge from pretraining after being fine-tuned on a specific downstream task.

Instead, PEFT (Parameter-Efficient Fine-Tuning) methods adapt LLMs to downstream tasks or domains by modifying only a small subset of parameters (typically less than 1% of the total parameter matrix), called *adapters*, while keeping the majority of the model weights frozen. In particular, in Chapter 5 we will apply the Low-Rank

Instruction Data Template

```
USER:       Given the possible relations:
            ["acts in", "directs", "directed by", "featuring" ]
            What are the relations between the subject entity and the
            object entity expressed by the sentence?
            Provide the output as a triple (head, relation, tail)
            Sentence: Raúl Cimas is a Spanish actor, who played Jose
            Ramón in the comedy series Poquita Fe.
            Subject: Raúl Cimas
            Object: Poquita Fe
            Triple:

ASSISTANT:  (Raúl Cimas, acts in, Poquita Fe)
```

Figure 2.9: Schema of a training data template for instruction tuning on a RE task.

Adaptation (LoRA) PEFT technique [YYK+23]. For each transformer block weight matrix[12] $W \in \mathbb{R}^{d \times k}$, LORA approximates it with a linear combination $W + \Delta W$ and then decomposes the update matrix ΔW into two low-rank matrices:

$$\Delta W = BA \tag{2.7}$$

with $A \in \mathbb{R}^{r \times k}$, $B \in \mathbb{R}^{d \times r}$ and $r \ll min(d, k)$.

Now the forward pass computation within the transformer block is:

$$h = Wx + \alpha BAx \tag{2.8}$$

but W is kept frozen during training, while the factorized weight matrices A and B that are updated have a total size of $r(d + k)$, which is much lower than $d \times k$.

By training only lightweight low-rank updates to specific weight matrices (e.g., attention projections), LORA dramatically reduces training cost while keeping inference efficient, making it the most popular approach for domain adaptation of LLMs.

[12] Notice that each of these weight matrices is roughly on the order of 1000×1000 in size, depending on the model.

Chapter 3

Digital Transformation Monitoring

3.1 The Process of Digital Transformation

Digital Transformation (DT) is recognized as "an holistic reconfiguration of organizational strategies, processes, and culture enabled by digital technologies, with the aim of creating new forms of value and ensuring long-term adaptability" [Via19, BESPV13]. DT is made possible by "core enabling technologies", an evolving set of digital tools and infrastructures that act by reshaping processes, customer experiences, and business models. These include; Cloud Computing, Big Data and Advanced Analytics, allowing the collection and processing of massive and heterogeneous datasets as well as predictive modeling and data data-driven decision making; Artificial Intelligence (AI) and Machine Learning (ML); Internet of Things (IoT), networked sensors and devices generating real-time data; Robotic Process Automation (RPA); Blockchain and Distributed ledger Technologies, ensuring trust, traceability, and transparency in digital transactions. However, unlike mere digitization and digitalization, which focus on technological conversion and enhancement of existing processes via automation, DT implies fundamental organizational change and innovation in business models([WCB+11].

Consequently, monitoring the process of DT involves tracking an heterogeneous range of domain entities from both scholarly and industrial publications (scientific papers, patents) as well as in the fast-reactive news and social media, tracking concepts like computational methods, algorithms, infrastructures and platforms as well as key players as varied as researchers, innovators, academic institutions, industry and financial corporations. This vast set of domain entities is interconnected by an heterogeneous network of semantic relations, including software processes like method implementation, customization, model training and deployment, as well as managerial and financial activities like technology adoption, company acquisition and merging.

3.1.1 Monitoring Unconventional Sources

The European Commission's Competence Center on Composite Indicators and Scoreboards[1] at the Joint Research Centre (JRC)[2] is carrying out research activities aimed to track societal and economic activities in European countries using unconventional data [CCPB22]. In particular, they explore the application of data-driven and AI modeling to the creation of tools assisting investors in decision-making and policymakers in creating policy interventions, assessing their potential to boost economic growth and enhance societal well-being. Applying such technologies to social media and news has a great potential for forecasting and nowcasting methods, since they provide a larger set of information than standard, lower-frequency socio-economic indicators [BCM22, CBM22].

DT is an ideal target for this endeavor, both for its disruptive change potential in the EU socio-economic ecosystem and because the discourse on DT is pervasive through the news and crowdsourced content platforms such as social media. Therefore, we have designed, implemented and evaluated two prototype pipelines contributing to a underdevelopment DT monitoring system from alternative sources, namely:

- *Triplétoile*, a pipeline for the extraction of a knowledge graph of open-domain entities from micro-blogging posts on the social media platform X[3] (formerly Twitter);
- an enhanced architecture for the extraction of a knowledge graph of open-domain entities from news articles about digital health technology

3.1.2 Challenges

Examining, connecting, and understanding content sourced from microblogging platforms presents several challenges, particularly demanding due to the Internet's diverse array of social platforms, featuring natural language text in varying formats, structures, and lengths.

Social media analysts and various stakeholders commonly navigate them via aggregation tools such as Hootsuite[4], Brandwatch[5], Talkwalker[6], Sprout Social[7]. However, these platforms are constrained to basic queries and merely provide a list of pertinent documents that require manual analysis, while not supporting advanced queries regarding the entities mentioned in the posts.

In order to enable the detection and tracking of potential trends, gauge the influence of events or individuals, and understand their relationships, the research community has put forth numerous proposals aimed at generating organized, interconnected,

[1] `https://composite-indicators.jrc.ec.europa.eu/`.

[2] The Joint Research Centre (JRC) of the European Commission (EC): `https://ec.europa.eu/info/departments/joint-research-centre_en`.

[3] `https://x.com/`

[4] `https://www.hootsuite.com/`

[5] `https://www.brandwatch.com/`

[6] `https://www.talkwalker.com/`

[7] `https://sproutsocial.com/`

and machine-readable data frameworks of social analysis knowledge found within text from microblogging platforms, typically using KG technologies [RS16, DKC+22, HYS20]. Nonetheless, creating extensive and high-quality KGs from social media is a current open problem. Support tools that aid social media experts in structuring their knowledge ([BFVX19]) represent poorly scalable solutions, while information extraction (IE) approaches [DKC+22, AGP+17, MRLARA18] have the potential for scalability but often struggle to generate outputs of sufficient quality for practical applications [DORR+21].

Instead, crafting a large-scale, coherent, and semantically sound representation of social media texts drawn from millions of posts, involves addressing at least the following challenges:

- integrating the extracted information from various posts into a cohesive representation, merging operations of entities and relations via linking to external knowledge bases;
- defining a flexible ontological framework to formalize a range of statements originating from social media posts
- estimating the validity of the resultant triples and its correlation with triple support from text sources;

In order to address these issues, we designed two scalable and flexible architectures for triple extraction from social media and news text. The proposed pipelines, based on open IE paradigm, support the detection and merging of entity instances matched in text as well as the generalization of various relationships among these entities by using hierarchical clustering, word embeddings, and dimensionality reduction techniques. The manual evaluation we conduct on the triple sets generated by the pipelines reveal that they outperform alternative methods in terms of accuracy, while at the same time generating a relatively higher number of triples.

3.2 Data Collections

3.2.1 Social Media

For our experiments on DT monitoring from social media, we collected a topic-specific dataset of tweets by using the (now discontinued) Twitter public API v2 full-archive search endpoint[8]. Namely, we retrieved English language tweets from 2022 containing the hashtag #DigitalTransformation, removing all retweets. We store the resulting corpus of approximately 4M tweets in an Elastic Search index (shown in Figure 3.1), keeping tweet metadata and tweet ids, for linking back from the extracted triples.

[8] `https://developer.x.com/en/docs/x-api/early-access`

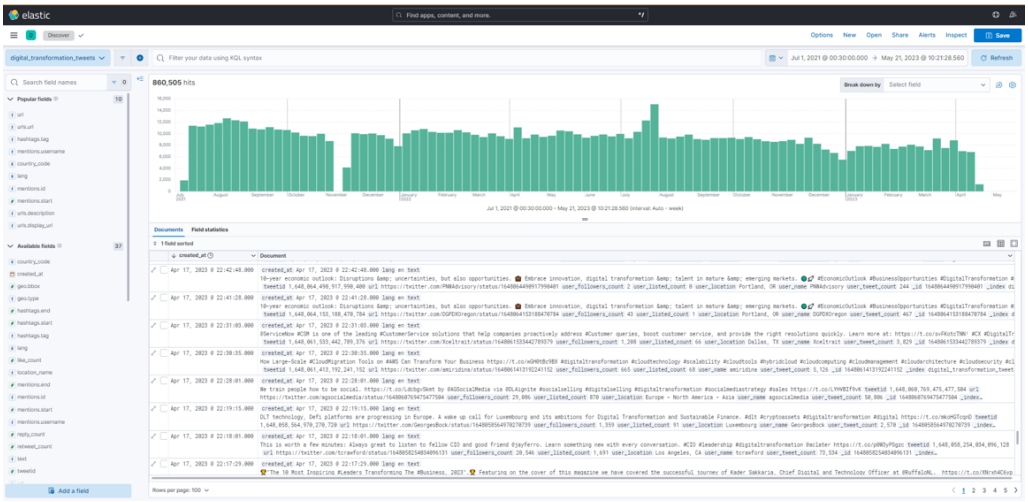


Figure 3.1: Snapshot of a Kibana dashboard visualization of the tweet collection Elastic Search index.

From the stored collection, we sampled a dataset of around 100k, after removal of duplicates and near-duplicates[9]. This is the input data to our social media graph generation pipeline.

3.2.2 News

The news analysis pipeline is applied to a topic-specific news dataset reporting updates on different aspects of the Digital Health domain.

The initial dataset comprises around 7.8 million English-language news articles gathered from the Dow Jones Data, News, and Analytics (DNA) platform[10], covering the time frame September 1987 through December 2023 and originating from diverse global English-language outlets, such as The Wall Street Journal, the New York Times, and The Guardian.

In addition to the basic article data such as title, full text, publication date, etc., DNA provides a range of curated content-based descriptors that are useful for filtering along specific dimensions. These include an 8-level taxonomy comprising approximately nine hundred Subject codes; a 7-level industry code taxonomy, and a set of Region codes encompassing all countries and regions mentioned in the news items.

We started by discarding spurious news items[11] and by filtering and merging region codes, ending up with a two-valued (*Europe/US*) macro-area attribute.

We then tested for various combinations of DNA metadata tags as a means to collect a representative sample of news articles about Digital Health technologies. How-

[9]Namely tweets over a 0.85 Levenshtein string similarity threshold, computed after applying the preprocessing described in Section 3.4.

[10]`https://professional.dowjones.com/developer-platform/`

[11]Articles with missing titles or with text body character length lower than 300.

ever, health-related Subject tags fall short of retrieving financial/market news updates involving health tech key players, while DNA's Industries classification schema is too coarse-grained to capture emerging technologies and products in this domain. Therefore, we opted for using a trained Deep Learning binary classifier to this purpose.

Topic Classifier We fine-tuned the BERT (Bidirectional Encoder Representations from Transformers)[12] language model using a near-balanced small set of 9097 news items sampled from DNA and several RSS feeds from specialized news outlets in health tech[13]. We will refer to positive instances as digital health-related documents, while negative instances will denote non-digital health-related documents.

Out of the 4602 negative instances, 3000 were concatenated title and full text of a sample of 500 items for each set of 'negative' topic codes[14] and 1602 were title and full text of articles scraped from 'negative' topic feeds of technology news outlets[15]. As for the 4495 positive instances, 4187 consisted of articles from the health tech news outlets mentioned above, while positive instances from DNA were sampled by filtering for health-related Subject codes and manually checking the results, ending up with a subset of 308 health tech items.

The textual data underwent preprocessing, which involved the removal of URLs, all-numeric tokens, and DNA and news outlet-specific tokens (e.g., "Reuters", "Reuters Limited", "techcrunch"). Additionally, all texts were truncated to 1000 characters to eliminate any correlation between the topic and text length features of the article sources.

We then performed fine-tuning using 10-fold stratified cross-validation with 80-20% data splits and Binary Cross Entropy as Loss function, training for 10 epochs with Early Stopping on 1 epoch of non-increasing Accuracy score. To mitigate over-fitting on the relatively small training set we kept the model size small (4.3M trainable parameters) and added a dropout regularization layer in the training phase (0.2 dropout rate).

The model reaches average cross-validation F1 score of 98.6%. Moreover, on an additional, hold-out test set comprising 100 negative and 100 positive instances, sourced from DNA using Subject codes filtering, the classifier achieves 98.8% F1 score, with the Recall falling short of the Precision (93% and 98.9%, respectively). The results indicate that, while missing a high number of positive instances, the model is able to sample a consistent subset of relevant health tech articles from the DNA multi-domain corpus. Therefore, we deploy it on the entire DNA dataset to achieve an overall set of 97k health tech articles (1.2% of the entire DNA input data) for our further analysis. The model, after training on the entire train set, has been made publicly available at the project

[12] `https://tfhub.dev/tensorflow/small_bert/bert_en_uncased_L-2_H-128_A-2/1`

[13] For example, `https://www.healthtechdigital.com/`, `https://techcrunch.com/tag/healthtech/feed/` `https://www.digitalhealth.net/news/`.

[14] Namely, *gcat* (Political/General News), *mcat* (Commodity/Financial Market News), *ccat*(Corporate/Industrial News), *ecat*(Economic News), *gent*(Arts/Entertainment), *gcrim*(Crime/Legal Action).

[15] For example, `https://techcrunch.com/tag/security/`.

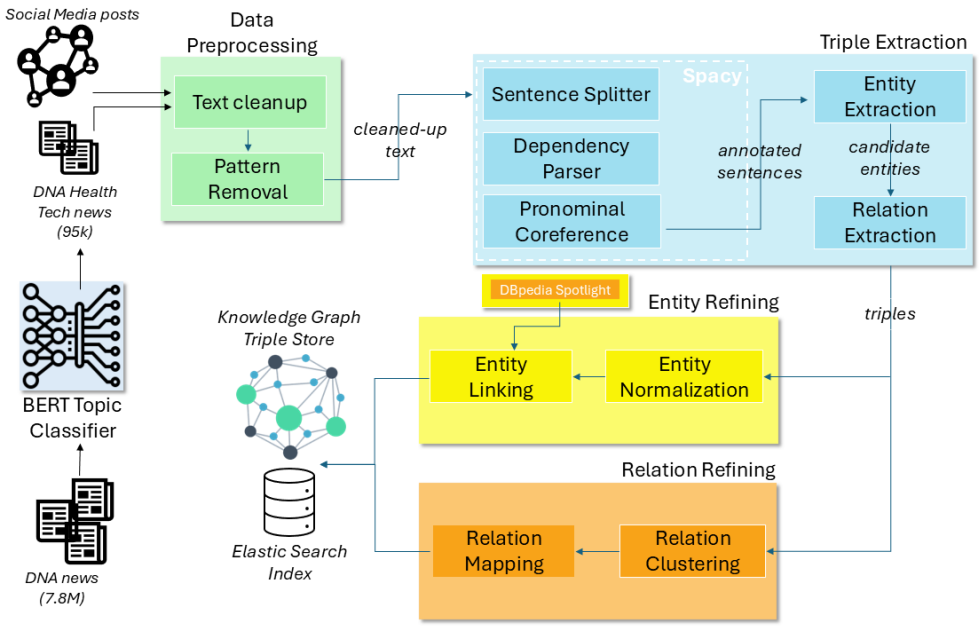


Figure 3.2: Merged flowchart of the pipelines for knowledge graph generation from micro-blogging and news wire text sources.

repository[16].

3.3 Architectures

Figure 3.2 shows a merged workflow of the two proposed architectures for KG generation. The information extraction architectures consist of customized NLP pipelines built using the spaCy libraries [HM17][17] coupled with a series of novel Entity and Relation processing modules.

The two pipelines share the same core modules, with the news-based pipeline adding a pre-filtering step based on the deep learning-based topic classifier described above.

The main blocks of the KG generation architecture include:

- *Data Preprocessing*, a step responsible for the normalization of the micro-blogging text in order to make it processable by the downstream text analysis modules;

- *Triple Extraction*, a block comprising core modules applying text processing libraries and models for the extraction of candidate entity-relation triples. It generates a set $E = e_0, ..., e_n$ of non-unified, candidate entity phrases, a set of verbal

[16] https://github.com/zavavan/dtm_kg/tree/master/data-collection/dna/bert_fine_tuned_healthTech

[17] https://github.com/explosion/spacy-models/releases/tag/en_core_web_lg-3.6.0.

relations $V = v_0, ..., v_k$ and a set of triples $S = s_0, ..., s_k$ in the form $< e_m, v_i, e_n >$ where $v_i \in V$ and $e_m, e_n \in E$.

- *Entity Refining*, a block responsible for the cleaning and generalization of entity mentions to canonical forms, in view of subsequent entity merging;

- *Relation Clustering*, in which relation instance verbal forms are mapped to canonical forms, computed as a representative element of the relation cluster they belong to;

The final objective of the pipeline is to enable the generalization from the surface form triples in set S to a smaller set $T = t_0, ..., t_h$ of triples in the form of $< \epsilon_m, r, \epsilon_n >$, where each $\epsilon_i \in E$ represents a unified entity and r is a label drawn from a generalized relation vocabulary R. In other terms, the final output is a knowledge graph of generalized triples annotated with references to the micro-blogging/news text items they were matched in.

The following subsections describe in more detail the individual components of the pipeline across the four main blocks and how they are applied. The code repositories for the two pipelines are referenced in the Appendix B.

3.4 Text Pre-processing

For news data we are able to perform only minimal normalization[18] since the models we subsequently apply for triple extraction are known to perform with high accuracy on benchmark corpora with comparable characteristics[19].

Twitter status updates (tweets) instead, feature an informal (often plainly ungrammatical) style and an abundance of platform-specific conventions that are known to be hard to process for standard NLP tools, thus requiring ad-hoc preprocessing.

We follow a two-fold approach to tweet normalization [SBL22] which can be readily extended to normalize social content from other platforms [TT15, CBF20].

On one hand, we remove tokens and token sequences encoding platform-specific metadata or denoting communicative conventions that (typically) do not carry any syntactic function in the tweet sentence, such as sentiment emoticons and smileys, reserved tokens (e.g., RT for 'retweet') and URLs.

On the other hand, we keep by default other platform-specific tokens that can carry syntactic functions depending on the context, like hashtags and @ entity mentions (e.g. *#digitaltransformation, @NASA*). Then, we identify token patterns that typically disrupt the syntactic parsing of the sentence, and remove them from the original tweet. For example, among the preprocessing heuristics:

[18] Basically removing URLs and a list of other news platform-specific token patterns.

[19] For example, the OntoNotes corpus (`https://catalog.ldc.upenn.edu/docs/LDC2013T19/OntoNotes-Release-5.0.pdf`)

1. we remove sequences of n entity mentions and retweet markers at the beginning of a sentence, for $n > 1$ or when the sequence is not followed by a verb. For example, we remove the leading sequence in *"@bansijpatel @RTatsat @kiranpatel1977 Thanks for updating the information with us."* but not in *"@AMDRyzen enabling #DataAnalytics in [...]"*.

2. for any sequence of size $n > 1$ hashtags/mentions/URL, we drop the sub-sequence with indexes $[1 : n]$ or drop the entire sequence if preceded by a sentence closing marker like ('!',':','?','.'). For example, in the text *"According to the @PayNews survey, 84 percent of #employees in the U.S. have instant access to #information about their pay and #benefits #Sapper #AI #hr #support #goals[...]"* we keep only the first element of the trailing hash tag sequence.

Entity mentions and hashtags, that are typically removed from tweet preprocessing pipelines for NLP tasks such as sentiment analysis, are highly relevant for knowledge graph generation as they can be nominal subjects, objects, or modifiers within dependency parse trees and therefore can be extracted as elements of candidate triples, like the tokens @mymdec and #SME in Figure 3.3a. Notice, though, that the trailing sequence of purely referential elements can often lead to noisy edges, for example in the figure the parser wrongly draws a $dobj$ dependency edge from the main verb "launches" onto the hashtag #digitaltransformation.

Figure 3.3b shows that the application of the second preprocessing heuristics above can enhance the parsing of the tweet, without losing too much information. The preprocessing step is carried out using the output of spaCy's English transformer pipeline *en_core_web_trf-3.6.1*[20].

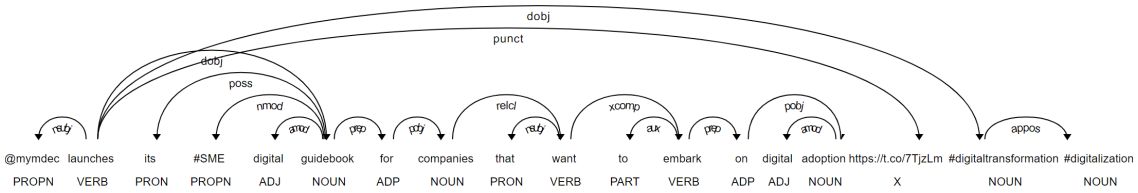
3.5 Triple Extraction

In the triple extraction block, preprocessed tweets are split into sentences and each sentence is fed to a spaCy pipeline. Building upon the works in [DOR+22a] and [DOR+22b], we define a set of procedures to extract candidate nominal entities and predicative triples connecting them, out of dependency parse trees computed by spaCy models.

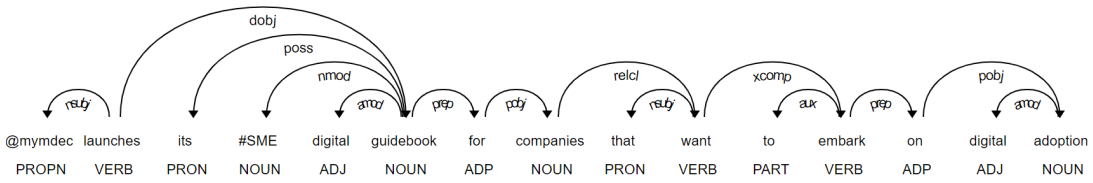
3.5.1 Entity Extraction

The entity extraction module detects local nominal phrases with a restricted range of syntactic modifications (e.g., compound nouns and adjectives). Then it connects and expands them with: a) a non-recursive set of attached prepositional phrases; b) spaCy quantity-type entities, such as *money*, *percent*, *quantity* and *cardinal*. We also use pronominal anaphora links, output by the spaCy module coreferee[21] replacing them with

[20] `https://github.com/explosion/spacy-models/releases/tag/en_core_web_trf-3.6.1`
[21] `https://github.com/richardpaulhudson/coreferee`



(a) Dependency parse of a tweet's original text.



(b) Dependency parse of a tweet after preprocessing

Figure 3.3: Example of tweet preprocessing.

the expanded entity spans of the tokens they reference. In Figure 3.4 we show a sample of extracted candidate entities.

For multi-token entity spans including quantifying modifiers (e.g. *'Less than 15% of the #banks'*) we maintain a structured representation separating the lexical head (*'#banks'*) from the quantifying modification of the noun phrase (*'Less than 15%'*), which then allows a more accurate entity normalization (see Section 3.6 below).

For the tweet collection, around 33.9% and 6.44% included hashtags and @ entity mentions, respectively; 3.34% were complex noun phrases with prepositional attachments while around 16.6% contained quantitative modifiers of any type (currency, percent, etc.). Out of all the generated triples, a 5.98% had either the subject or object entity consisting of a resolved pronominal anaphora.

Notice that at this stage the hashtag "#digitaltransformation" in the second sentence and the noun phrase "digital transformation" in the first are not mapped to the same general concept *digital transformation* yet, so that the triples in which they occur would still be considered as unrelated.

3.5.2 Relation Extraction

In the relation extraction module, for each sentence s_i all the shortest dependency paths (see Section 2.4) of the dependency tree between each pair of entities (e_m, e_n) containing a verb and matching any of the patterns listed in Table 3.1 below are selected.

The deployed path set has been selected through an expert validation process, carried out on an open-domain text corpus. This process consisted of the collection of the twenty most frequent paths, among all shortest paths connecting any pair of (automat-

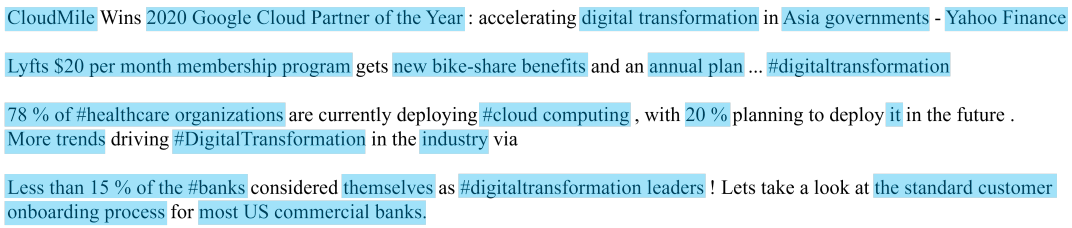


Figure 3.4: Visualization of candidate entities (highlighted in blue) extracted from a few sample tweets.

Target dependency paths
$[nsubj, dobj]$
$[acl, relcl, dobj]$
$[acl, dobj]$
$[nsubjpass, agent, pobj]$
$[nsubj, dobj, conj]$
$[nsubj, conj]$
Sample discarded paths
$[obj, pobj]$
$[obl, pobj]$
$[nsubj; pobj; nmod]$

Table 3.1: List of target and some of the discarded relation dependency paths.

ically detected) entities in the corpus. Then, three independent evaluators assessed the correctness of a random sample of 20 triples generated by each of these top frequency paths, majority vote was used to label each triple as correct/incorrect and only the subset of paths with a prevalence of correct triples (i.e., more than 10) were considered reliable and added to the list[22].

However, we found that in our tweet collection the extraction of triples via the dependency path $[acl, dobj]$ was a potential source of noise, in cases where the noun's clausal modifier was an infinitive verb, as exemplified in the following sentence:

"Salesforce really has the power to transform your business."

that incorrectly generates a triple such as $< power, transform, business >$. Consequently, we added a constraint to the dependency path $[acl, dobj]$ in order to filter out those paths where verb nodes had a relation *aux* with an infinitive particle node[23].

Analogously to what pointed out for the entities, note that the v in V are surface forms, that is individual inflected verbal phrases that do not enable as such to generalize triples over morphological or lexical variations. For example, the following triples:

$< BLEND360, acquires, EngagementFactory >$

$< BLEND360, acquired, EngagementFactory >$

$< BLEND360, bought, EngagementFactory >$

are considered distinct facts at this stage.

3.6 Entity Refining

The function of this module is to clean up and normalize the candidate entities into a normalized form that allows the merging across entity name variants[24].

Entities are first cleaned up by removing leading/trailing punctuation marks as well as stop-words. Afterwords, we distinguish the following cases for normalization:

- For hashtags and @ mentions, we remove hashtags and @ symbols, split the "camel case" forms (e.g., *#SmartCities*) and lowercase the resulting string.

- For all other entities, we lemmatize and lowercase all component tokens whose POS tag is neither Verb nor Proper Noun, otherwise we simply lowercase.

[22]The resource can be found at the URL `https://github.com/zavavan/dtm_kg/blob/master/resources/paths.txt`

[23]In the example above, *transform* has an *aux* relation to the particle *to* and, therefore, it is discarded. More precisely, the following expressions hold:

`SUBJ`=*power* $\xrightarrow{\text{acl}}$ `PRED`=*transform* $\xrightarrow{\text{dobj}}$ `OBJ`=*your business*

`PRED`=*transform* $\xrightarrow{\text{aux}}$ `to`.

[24]Splitting is another typical subtask of Entity Refining functions, for example by separating the individual entities in parsed coordinated noun phrases like in *'#testautomation and #datamanagement can accelerate your #digitaltransformation'*. However, we deal with these cases earlier on at the triple extraction phase by generating a triple for each coordinated entity.

- For nouns that have variants in American English, we finally map to the British English variant.

We make use of such normalized versions of the candidate entities for merging them, by using the spaCy-integrated DBpedia Spotlight Entity Linking library[25].

The DBpedia Spotlight model is trained to perform both entity detection and linking at once. In order to power this module with the entity normalization, performed by our pipeline, we run the module over modified article sentences where the original subjects and object entity spans are replaced with their normalized forms. Entities that are linked to the same DBpedia entries are merged. More precisely, we link the normalized versions of the entities to the corresponding DBpedia entries of the spaCy native entities whose text spans are both:

- included within the subject or object text spans of the corresponding normalized entities;
- overlapping with the syntactic heads of the corresponding normalized entities.

In other terms, we let spaCy's DBpedia Spotlight perform the merging of entities that were normalized to the same or similar forms, by linking to the same DBpedia entries. For example, the two candidate entities *"Gartner"* and *"@Gartner_inc"* are merged together by linking them to the DBpedia entry of the Gartner consulting firm `http://dbpedia.org/resource/Gartner`).

In case only the first condition is met, we assign a semantic "relatedness" link between the candidate entity and the DBpedia entry, indicating that the former is not an instance of, but rather is related to the latter[26]. For example, the span *'@gartner_survey'* is considered only "related' to the DBpedia entry for Gartner.

In Section 3.9 we describe how these relations are encoded in the resulting knowledge graph by inheriting from existing ontology relations.

3.7 Relation Refining

The extracted triples often contain numerous distinct relations that convey similar meanings. To minimize redundancy and support semantic retrieval of the triples in the generated graph, these extracted relations need to be consolidated into a smaller set of predefined relations. This block aims to find the best predicate label r for each relation verb v in a triple $< e_m, v, e_n >$ and to map v to r in the resulting triples.

The approach we followed consisted of deriving a word embedding representation of the verb predicates from a pre-trained model, computing an optimized clustering of the relation vectors, and finally using a representative instance of each cluster to map verb

[25] `https://spacy.io/universe/project/spacy-dbpedia-spotlight`

[26] We keep out the cases when only the second condition is met, as they typically arise from inaccuracies of the entity span detection.

predicates. A similar method, using however a "bag of words" vector representation, was proposed by [HSG04].

After experimenting with several (contextual and non-contextual) word embedding models and clustering algorithms, we converged to a setup using static word embeddings learned with GloVe (see Section 2.4.3) and applying HDBSCAN clustering to the vectors. We tested using verb phrase contextual embeddings from Huggingface's *bert-large-uncased*[27] and Sentence-BERT[28]. However, it turned out that the optimal cluster scores, in this case, were achieved for a number of clusters too close to the number of items in the dataset[29]. Table 2 in Appendix B reports the clustering scores and number of resulting clusters for some best performing configurations using the different embedding models, for the tweet collection.

Relation Embeddings For each single or multi-token relation predicate, we get the static, 300-dimensional word embedding vector made available for text span objects in the spaCy *en_core_web_lg-3.6.0* pipeline[30].

Dimensionality Reduction and Clustering We used the HDBSCAN clustering algorithm enhanced by previously applying the UMAP dimension reduction technique on the word embeddings vectors (see Section 2.4.3).

In order to optimize the combination of UMAP and HDBSCAN, we perform a grid search over the hyperparameters of both algorithms and evaluate the clustering using the score indicated in Equation 3.1:

$$S = silhouette_X \cdot clustered_X, \tag{3.1}$$

where the silhouette coefficient $silhouette_x$ of an instance $x \in X$ is defined in Equation 3.2:

$$(b\text{-}a)/max(a,b), \tag{3.2}$$

with a being the mean distance to the other instances in the same cluster and b being the mean distance to the instances of the next closest cluster.

In the S score formula, the $silhouette_X$ is the mean silhouette coefficient over all the instances of the dataset X that were clustered by HDBSCAN [BH21] while $clustered_X$ is the fraction of instances of X that were actually clustered by HDBSCAN.

In practice, we optimize for the classical measure of cluster cohesion and separation while penalizing the configurations with low coverage of the dataset. We finally chose a subset of best-scoring hyperparameter configurations and plotted their S score over

[27] https://huggingface.co/bert-large-uncased

[28] `https://sbert.net/`

[29] In other terms, these representations were not suitable for generalizing enough over relations, probably due to the context-specific information they are encoding.

[30] `https://github.com/explosion/spacy-models/releases/tag/en_core_web_lg-3.6.0`

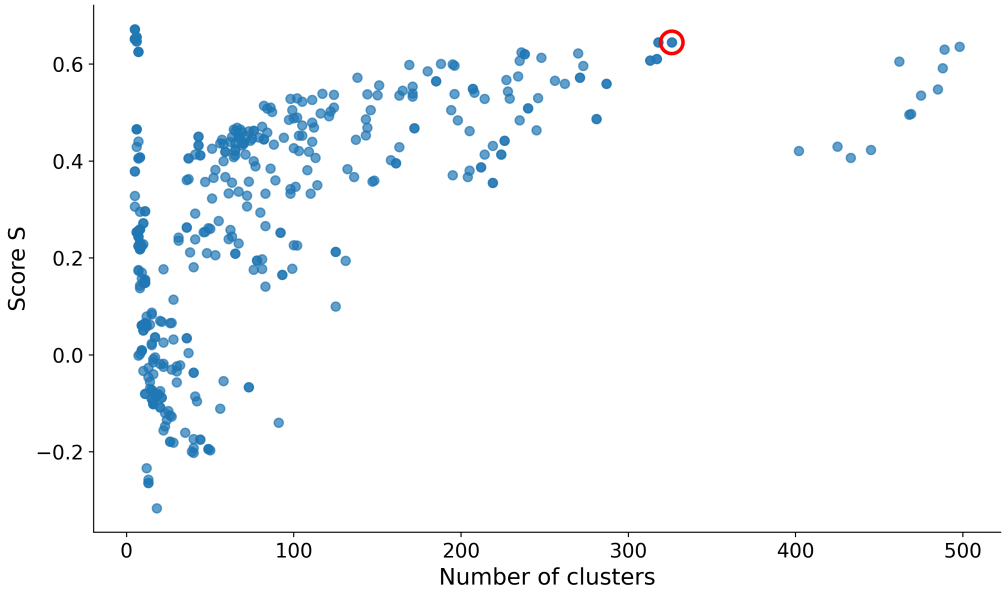


Figure 3.5: S score over number of generated clusters for a subset of best-scoring UMAP-HDSCAN hyperparameter configurations on GLOVE embeddings of relations, from the tweet collection, with the picked up sub-optimal value circled in red.

the number of output clusters they generate, so that we are able to pick a sub-optimal configuration that strikes a balance between generalization (fewer clusters) and accuracy (cluster number closer to the dataset size).

In our experiments with the *X/Twitter* data, we started with a set of 29,335 raw triples[31] from which we computed and standardized 2,539 unique 300-dimensional word embeddings from GloVe. Figure 3.5 reports the S score over the number of resulting clusters for a subset of best-scoring UMAP-HDBSCAN configurations in the grid search. Highest score values are initially reached for configurations sensitive to the global structure of the dataset (very few clusters, very coarse-grained generalization), while they subsequently tend to grow with the increasing number of clusters until they flatten again. We picked up a sub-optimal configuration with an overall score of around 0.65, silhouette score on clustered points 0.73 and data clustering percentage 0.89, returning 327 clusters, with an average cluster size of 7.6 elements[32].

The same procedure was followed for the news dataset, where the chosen configuration reached a S score of 0.62, with $silhouette_X = 0.65$, $clustered_X = 0.92$ and 2 UMAP components.

[31] These are surface-level candidate triples from the Triple Extractor, counted prior to entity and relation merging.

[32] This corresponds to an UMAP 2 dimension-reduced representation of the vector dataset, obtained using a $n\_neighbors = 5$ hyper-parameter and to a $min\_cluster\_size = 3$ and $min\_samples = 1$, for HDBSCAN.

Relation Verb	Relation Predicate	Example
fuel	FUEL	*'How the UR+ Ecosystem is Fueling Cobot Market Growth'*
driven by	FUEL	*'Digital transformation in Ho Chi Minh is being driven by remote working'*
accelerated by	FUEL	*'huge social trends being accelerated by the pandemic.'*
identify	IDENTIFY	*'Machine learning can identify signs of Alzheimers in patients '*
quantify	IDENTIFY	*'Research quantifies G's potential in roaming and manufacturing '*
predict	IDENTIFY	*'AI-supported test can predict eye disease that leads to blindness'*

Table 3.2: Sample relation verb-predicate mapping.

Relation Mapping Finally, for each relation verb v in the dataset, we replace it with the predicate label r consisting of the lemma of the most frequent relation in the cluster of v[33]. Otherwise, we map it to itself if v was an outlier and not clustered. For the news dataset, we slightly modify the heuristics and map each v to a predicate label r consisting of the most frequent lemma among the "exemplars" relations returned by HDBSCAN for the cluster of v.

Thus, the three distinct triples shown in the last example of Section 3.5.2 would be merged and the resulting triple would be:

$< BLEND360, BUY, EngagementFactory >$[34].

Table 3.2 shows some sample mappings from relations extracted from the tweet collection to their associated predicate labels, consisting of the lemma of the most frequent relation in their clusters.

3.8 Evaluation

We perform a twofold evaluation of our Triplétoile pipeline, based on data from the tweet collection: a. we manually assess the truthfulness of a sample set of statements, estimating Precision, Recall and F-measure; b. we evaluate our pipeline Precision against a number of alternative methods.

[33] `https://hdbscan.readthedocs.io/en/latest/api.html`. Notice that as HDBSCAN can generate clusters of arbitrary forms, it does not hold a notion of cluster centroid and there are typically multiple 'most representative' data points in a cluster, based on soft clustering.

[34] A CSV file with a sample of the most frequent normalized triples for the tweet dataset, together with the originally matched relations can be found in the project repository at `https://github.com/zavavan/dtm_kg/blob/master/data-collection/twitter/sampleNormalizedTriples.tsv`

Human Expert Assessment We randomly select 483 statements, equally distributed among high-support (support greater or equal to 5) and low-support triple groups and we have each triple assessed by three evaluators as True or False. The True label was assigned when all the following criteria were satisfied:

- the subj and obj entities are linked by a relation in the tweet text;
- the assigned relation label entails the relation verb in the tweet text;
- the spans of the subject/object of extracted triples include the syntactic head of the relation's subject/object[35].

We calculated the average pair-wise Cohen κ inter-rater agreement and the Fleiss κ_F agreement of all the 3 raters [FQ15], resulting in a score of 0.61 and 0.558, respectively. These values generally indicate a substantial level of agreement, although one annotator featured an outlier rating on a specific category of cases.

The majority vote-based assessment of the three annotators yielded a Precision of 0.96. In order to compute Recall, the three annotators were tasked to extract triples that they deemed correct from the same tweets containing the 483 selected statements. They extracted a total of 484 triples from a resulting set of 491 tweets (we considered the union of all the triples extracted by each annotator). Consequently, we were able to calculate the number of true positives ($TP = 464$), false positives ($FP = 19$), and true negatives ($TN = 20$) and compute a Recall of 0.95 and F1 score of 0.95. Individual rater estimates ranged from 0.90 to 0.96. These results indicate that the pipeline can extract triples with good precision from noisy text like tweets, while at the same time missing only a few triples.

Comparative Evaluation For a comparative evaluation vis-à-vis alternative triple extraction methods, we randomly sampled 500 tweets from the 100k-sized original dataset and used our pipeline to extract candidate entities. We then merged this set of entities with the one generated by the DyGIEpp Extractor [WWLH19].

DyGIEpp is an joint NER-RE framework, similar to the one outlined in Section 2.5.3. In order to detect entities, DyGIEpp uses a feed-forward neural network on textual span representations and computes a score for each entity type; an entity is detected considering the highest score for an entity type if a minimum threshold is met.

We then employed four alternative methods to identify relationships between these input entities and to extract triples from the 500 tweets. Specifically, we compared:

- *OpenIE Extractor*, the IE tool of the Stanford Core NLP suite, described in Section 2.4.2;

[35] For example, a triple $< 78\%\_of\_\#healthcare, USE, Digital\_Transformation >$ would be marked as False if extracted from the text *"78% of #healthcare organisations deploy #DigitalTransformation"* as the syntactic head is "organisations".

Extraction Method	Precision
OpenIE Extractor	0.52
PoST Extractor	0.17
Dependency-based Extractor	0.77
Entity and Relationship Refiner	0.31
Triplétoile	**0.82**

Table 3.3: Precision of the triples extracted from a set of alternative methods from a collection of 500 tweets, using a combination of Triplétoile and DyGIEpp input entities.

- *PoST Extractor*, a module built on top of the Stanford Core NLP suite that uses PoS tags to find all verbs that exist between two candidate entities in a sentence to extract verb relations, using a window of max token distance 15 between the entities;
- *Dependency-based Extractor*, a module that extracts dependency trees using the dependency parser of the Stanford Core NLP suite, maps entities previously extracted using DyGIEpp into the sentence tokens, and exploits 12 hand-crafted paths[36] to find verb connecting entities.
- *Entity and Relationship Refiner*, a module that applies *Entity and Relationship Handlers* as described in [DOR+22b] to the set that includes *OpenIE Extractor*, *PoST Extractor*, and *Dependency-based Extractor* triples.

The number of extracted triples from the dataset ranged from 339 for the Dependency-based Extractor to a maximum of 1,015 for the PoST Extractor, which does not impose any filter on the dependency relations connecting verbs that exist between pairs candidate entities. After PoST Extractor, Triplétoile is the one generating the largest number of triples (663) among the methods using the extended range of candidate entities, with OpenIE Extractor producing 588 triples Entity and Relationship Refiner reaching 348 triples. In order to use these numbers as an indirect assessment of the relative Recall levels of the different pipelines, we manually assessed also the Precision on a limited random sample of 150 triples generated by each method, using the same human assessment setup described earlier and reaching a strong κ_F agreement coefficient of 0.86[37]. We report the results in Table 3.3, with the precision of our pipeline largely outperforming all the alternative methods.

Overall, we conclude that the proposed method is able to extract semantically accurate triples from noisy text data such as micro-blogging posts, while featuring a competitive recall against other NLP methods. We do not arguably expect a decrease in accuracy decay on the more standard language style of news data, although we lack a

[36] `https://github.com/danilo-dessi/SKG-pipeline/blob/main/resources/path.txt`

[37] Notice that these test sets are not generated from the same tweet subset for each pipeline. Notice also that the random sampling was done without using any information on the triple support, which was not available for the alternative pipelines.

Subject Entity	Relation	Object Entity
pandemic	accelerate	digital_transformation
artificial_intelligence	impact	insurance_sector
microsoft	buy	riskiq
data-driven_insight	drive	decision-making
agile_business	demand	effective_marketing_capability
hootsuite	buy	ai_chatbot_firm
automl	generate	data-driven_insight
image_classification	use	transfer_learning
new_belgium_brewing	implement	digital workflow_place_solution
e-rupi	back	existing indian rupee
82%_of_cio	implement	new_technology
image_recognition_framework	use	artificial_intelligence
microinsurance	close	africa_insurance_gap
hsbc_qatar	introduce	mobile_payment
ford_motor_company	explore	blockchain_technology

Table 3.4: A sample of statements extracted from the tweet collection by the Triplétoile pipeline.

dedicated evaluation for it. Therefore, we deploy our method for the generation of large scale Knowledge Graphs from both use case text collections.

3.9 Digital Transformation Social Media Monitor Knowledge Graph

The *DTSMM_KG* (Digital Transformation Social Media Monitor Knowledge Graph) comprises approximately 22,270 statements (triples). We represented all statements extracted from the tweets using an *DTSMM_KG* ontology (with namespace prefix *dtsmm-ont*) we created for this purpose. Table 3.4 shows a statement sample.

Each statement is reified into *dtsmm-ont:Statement* class instances, which defines a specific claim extracted from a given number of tweets. Namely, each *dtsmm-ont:Statement* object includes:

- the reification of the triple itself via *rdf:subject*, *rdf:predicate* and *rdf:object* predicates;
- a data property *dtsmm-ont:hasSupport* reporting the number of tweets supporting the claim;
- a number of object property instances *dtsmm-ont:comesfromTweet* ranging over ontology instances of type *dtsmm-ont:Tweet* (which was inherited from *schema:SocialMediaPosting*) supporting the claim;

```
dtsmm-ont:statement_10100 a dtsmm-ont:Statement,
 rdf:Statement ;
dtsmm-ont:negation false ;
dtsmm-ont:comesfromTweet dtsmm:tweet_1424266328882429952 ;
...
dtsmm-ont:hasSupport 6 ;
rdf:subject dtsmm:multi_page_document_classification ;
rdf:predicate dtsmm-ont:use ;
rdf:object dtsmm:machine_learning .
```

Figure 3.6: A shortened example of reification for a Statement concerning the instance *machine_learning*, grounded by 6 tweets, with the three dots referring to the hidden *dtsmm-ont:comesfromTweet* predicates.

- A boolean data property *dtsmm-ont:negation* flagging whether a negation of the claim's predicate was parsed from the source text.

Figure 3.6 shows a shortened example of a claim reification having the *DTSMM_KG* ontology's instance *machine_learning* as *rdf:object* and support equal to six.

Figure 3.7 instead illustrates a sub-graph of *DTSMM_KG* containing a few sample triples having the instance *machine_learning* as subject. Here, we report just the statements, hiding claim reification for the sake of readability.

The linking of the *DTSMM_KG* instances to the DBpedia ontology (see Section 3.6) is implemented by using the *owl:sameAs* predicate, while the "relatedness" link between candidate entities and the DBpedia entries are encoded using SKOS predicate *skos:related*[38]. Overall, *DTSMM_KG* provides 2,857 *owl:sameAs* links and 3,309 *skos:related* links to to DBPedia entries.

The resulting data have been made publicly accessible under Creative Commons Attribution 4.0 International (CC BY 4.0) license[39] within the Joint Research Centre Data Catalogue[40], as well as within the European Data portal[41], the official data repository of the European Commission. The direct link to the Digital Transformation knowledge graph, available in Terse RDF Triple Language (Turtle), is `https://jeodpp.jrc.ec.europa.eu/ftp/jrc-opendata/CC-COIN/se-tracker/DTSMM_KG.ttl`.

3.9.1 Knowledge Access Example: Graph RAG application

Access to *DTSMM_KG* can be useful in application scenarios where one needs to bridge the gap between the wealth of information available in real-time data streams, like so-

[38] `https://www.w3.org/2004/02/skos/`

[39] Creative Commons Attribution 4.0 International license: `https://creativecommons.org/licenses/by/4.0/`

[40] `https://data.jrc.ec.europa.eu/dataset/f7be47f7-49a2-44e8-9dc8-043735af4139`

[41] `https://data.europa.eu/88u/dataset/f7be47f7-49a2-44e8-9dc8-043735af4139`

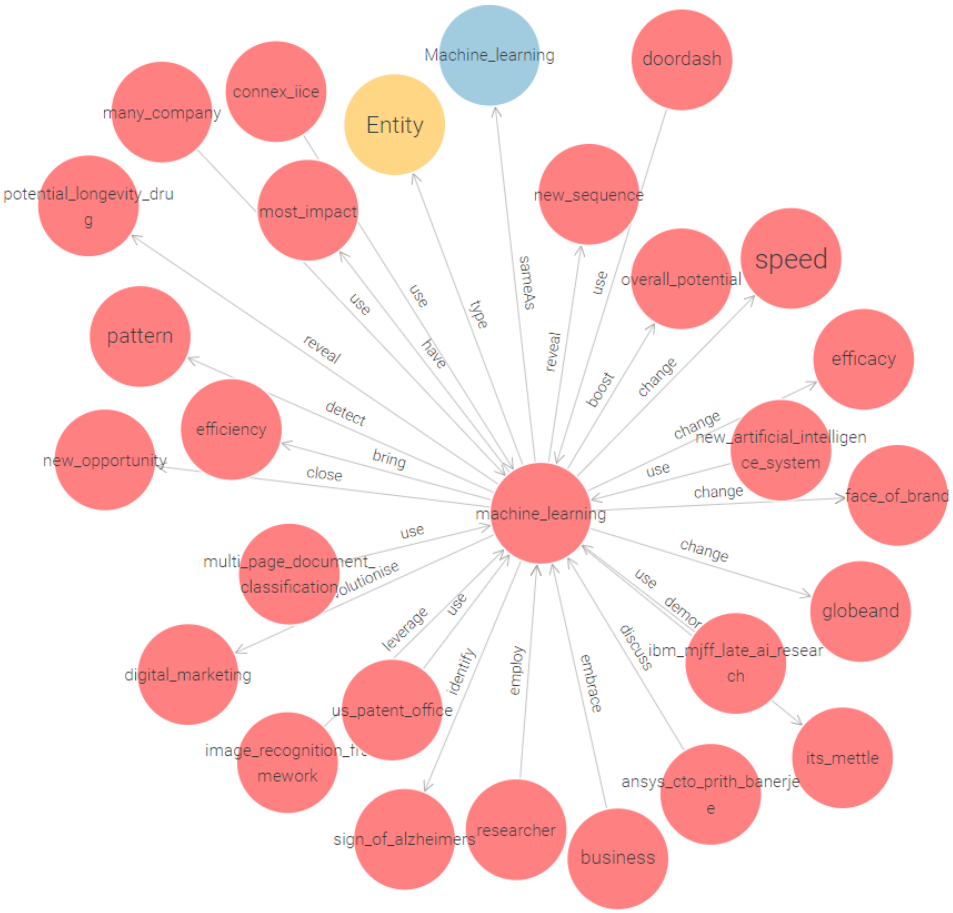


Figure 3.7: A sample *DTSMM_KG* subgraph showing a few claims for the instance *machine_learning*.

cial media, and more static, conventional sources. For example, it might serve as a critical resource for enriching Graph-based Retrieval-Augmented Generation architectures [LPP+20, ETC+25], in order to dynamically pulling in contextual information via KG querying during the generation process, thus enhancing the quality and relevance of the results.

As a simplified Question Answering example, assume one is supplying the following question to a RAG system:

Is Microsoft dedicating resources to computer security technologies? (3.3)

and assume a NER model is able to recognize the entities *Microsoft* and *Computer Security* from the text. *DTSMM_KG* can be queried to detect whether *Microsoft* entities are declared into its ontology:

```
SELECT DISTINCT *
FROM <DTSMM_KG>
WHERE { <http://dtsmmkg.org/dtsmmkg/resource/microsoft> ?p ?o . }
```

which would produce the following resulting triple (in RDF Turtle format):

```
@prefix dtsmm: <http://dtsmmkg.org/dtsmmkg/resource/> .
@prefix dtsmm-ont: <http://dtsmmkg.org/dtsmmkg/ontology#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .

dtsmm:microsoft a dtsmm-ont:Entity ;
 owl:sameAs <http://dbpedia.org/resource/Microsoft> .
```

informing us that the resource *Microsoft* is defined and equal to the well-known DBpedia entity `http://dbpedia.org/resource/Microsoft`.

This would allow to infer additional information, external to our knowledge-base, via the DBpedia SPARQL endpoint[42] using the query:

```
SELECT DISTINCT *
FROM <DTSMM_KG>
WHERE { <http://dbpedia.org/resource/Microsoft> ?p ?o . }
```

which returns 960 knowledge triples about Microsoft[43].

[42] Available at `https://dbpedia.org/sparql`

[43] One can see see all the triples by browsing directly DBpedia to the URL `http://dbpedia.org/resource/Microsoft`.

This existing knowledge can be enriched with new relations extracted from *DTSMM_KG*. For example, looking for triples involving the subject Microsoft with an "acquire" predicate type (i.e. `http://dtsmmkg.org/dtsmmkg/ontology#acquire`) via the SPARQL query:

```
prefix dtsmm: <http://dtsmmkg.org/dtsmmkg/resource/>
prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

SELECT DISTINCT *
FROM <DTSMM_KG>
WHERE {
 ?statement rdf:subject dtsmm:microsoft .
 ?statement rdf:predicate dtsmm-ont:acquire .
 ?statement rdf:object ?object .
}
```

we would get to know that Microsoft has acquired companies like *Cloudknox_Security*, *CyberX* and *RiskIQ*. In SPARQL we might then ask for information about this last:

```
SELECT DISTINCT *
FROM <DTSMM_KG>
WHERE { <http://dtsmmkg.org/dtsmmkg/resource/riskiq> ?p ?o . }
```

with Turtle result as follows:

```
@prefix dtsmm: <http://dtsmmkg.org/dtsmmkg/resource/> .
@prefix dtsmm-ont: <http://dtsmmkg.org/dtsmmkg/ontology#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix skos: <http://www.w3.org/2004/02/skos/core#> .

dtsmm:cybersecurity_firm_riskiq a dtsmm-ont:Entity ;
 owl:sameAs <http://dbpedia.org/resource/RiskIQ> ;
 skos:related <http://dbpedia.org/resource/Computer_security> .
```

The results would tell us that this is a Computer Security company. If we now would supply via RAG the existing 960 DBpedia knowledge triples on Microsoft plus the extracted relation triples deriving from our KG in-context to a LLM (in this example we used OpenAI GPT-4 Turbo[44]), and then ask our initial question 3.3, specifying to be

[44] `https://platform.openai.com/docs/models/gpt-4-turbo-and-gpt-4`

```
dhnewskg-ont:statement_90694 a rdf:Statement ;
dhnewskg-ont:comesfromNewsArticle dhnewskg:lba0000020030305dz34000cl ;
dhnewskg-ont:hasSupport 1 ;
dhnewskg-ont:negation false ;
rdf:object dhnewskg:receptor ;
rdf:predicate dhnewskg-ont:block ;
rdf:subject dhnewskg:drug_tamoxifen .
```

Figure 3.8: A sample reification for a statement concerning the *DTSMM_KG* resources `dhnewskg:drug_tamoxifen` and `dhnewskg:receptor`.

brief, we would get the following answer from the system:

Yes, Microsoft is dedicating substantial resources to computer security technologies, as evidenced by its acquisitions of companies like RiskIQ, a leader in global threat intelligence and attack surface management, and CyberX, which specializes in securing IoT devices.

where the latter information comes exactly from *DTSMM_KG*.

In summary, when generating textual content, the RAG model can then reference the most recent updates and developments in Digital Transformation encapsulated within our knowledge graph, which acts as a pool of novel knowledge taken from social media that RAG models can tap into.

3.10 Digital Health News Knowledge Graph

From the DNA news dataset we generated the Digital Health News Knowledge Graph (referred to as *DHNEWS_KG*), a KG comprising roughly 431k distinct (non-reified) triples, connecting 186k unique entities via a total of 1866 generalized relations. In the corresponding ontology, designed to describe *DHNEWS_KG* (*dhnewskg-ont* namespace prefix), each extracted claim is successively reified into instances of the `dhnewskg-ont:Statement` class. Figure 3.8 provides an example of a claim reification, with the ontology instance `dhnewskg:drug_tamoxifen` serving as `rdf:subject` and the `dhnewskg-ont:hasSupport` data property reporting the number of news articles supporting the claim.

A sample of generated (un-reified) statements is illustrated in Table 3.5, together with their support. The support distribution of the triples has a marked long-tail pattern, with a few key statements occurring frequently and a vast majority matched only a few times.

DHNEWS_KG inherits entity typization of entities linked to DBpedia via `owl:sameAs` predicate. Table 3.6 lists the 20 predominant DBpedia-inherited types

Table 3.5: Sample statements from *DTSMM_KG*, with their support.

Subject Entity	Relation	Object Entity	Support
Italy	report	coronavirus death	374
clinical trial	involve	patient	128
interactive graphic	track	global spread	90
fitch	undertake	sensitivity analysis	75
administration	approve	drug	47
meningitis immunity	fight	endometrial cancer	35
dow chemical	develop	drug	28
drug tamoxifen	reduce	risk	17
fibrocell	announce	fda acceptance	2
zealand pharma	announce	fda acceptance	1

```
PREFIX dhnewskg: <http://dhnewskg.org/dhnewskg/resource/>
PREFIX dhnewskg-ont: <http://dhnewskg.org/dhnewskg/ontology#>
SELECT ?statement
FROM <DHNEWS_KG>
WHERE { ?statement a rdf:Statement .
 ?statement rdf:subject dhnewskg:biogen . }
```

Figure 3.9: Query returning all *DHNEWS_KG* statements with the graph entity `dhnewskg:biogen` as `rdf:subject`.

within the graph. All not-linked entities are classified into the generic `dhnewskg:Entity` type. Of the total set of unique *DHNEWS_KG* entities, around 8% have been linked to DBpedia entries using 14975 `owl:sameAs` and 33345 `skos:related` predicates. Overall, 23.8% of all triples had either subject or object entities linked to DBpedia.

We made *DHNEWS_KG* publicly available via data access endpoints. Namely, we set up a Virtuoso SPARQL endpoint for this purpose where *DHNEWS_KG* can be queried, and analytical information on target entities, attributes, and relations can be retrieved in user-specified data formats[45]. As an example, a SPARQL query like the one in Figure 3.9 can be run to return all the 480 statements from the '*DHNEWS_KG*' graph having the target entity `dhnewskg:biogen` as subject, where `dhnewskg:biogen` is a graph resource linked to the DBpedia entry for the American multinational biotechnology company Biogen Inc.

Finally, we built aggregated analyses on top of *DHNEWS_KG* and made it available through a collection of interactive visualizations, accessible in the Hugging

[45] `https://api-vast.jrc.service.ec.europa.eu/sparql/`. Currently, the access is password protected, with credentials available upon request to authors. Provisional credentials for the reviewing process: (dtsmm_user, dtsmm_user_2024).

Table 3.6: Number of matches and unique matches of the 20 most represented DBpedia entity types in *DHNEWS_KG*.

DBpedia Entity Type	#Matched Entities	#Unique Entities
DBpedia:Organisation	20050	2640
DBpedia:Company	15667	1736
DBpedia:Country	12124	324
DBpedia:Disease	7881	918
DBpedia:Person	6594	2611
DBpedia:ChemicalSubstance	6583	1378
DBpedia:Drug	5927	1180
DBpedia:Politician	4069	1187
DBpedia:Work	1872	567
DBpedia:MonoclonalAntibody	1258	128
DBpedia:GovernmentAgency	1123	107
DBpedia:AdministrativeRegion	1088	185
DBpedia:City	971	205
DBpedia:Bank	789	128
DBpedia:Biomolecule	754	165
DBpedia:Group	729	109
DBpedia:AnatomicalStructure	656	151
DBpedia:ChemicalCompound	631	191
DBpedia:ArchitecturalStructure	593	183
DBpedia:Gene	586	124
dhnewskg-ont:Entity	800527	185653

Face Space `https://huggingface.co/spaces/zavavan/Digital_Health_News_Analytics_Dashboard`.

The visualizations across all the dashboard panel allow filtering per geographical macro-area (values *Europe/US/EU-US* values), enabling the detection of significant trends in digital health technologies within the European and US contexts.

The Top Entity Types bar plots in the dashboard show the predominant DBpedia-inherited entity types within the graph for triples tagged with Europe, US, and EU-US region codes via their article support.

For a subset of predominant types, the Top Key Entities plots track the occurrence of several key entities per year, with occurrence indicating that the entity is either the Subject or Object of an extracted triple in the KG. One can point out here how some major pharmaceutical industry corporations seem to have an impact on a global scale (both for Europe and the US), while major information technology giant corporations show different impacts in the Digital Health industry in the two contexts.

Lastly, within the Entity Chord Diagrams panel we present the most frequently connected entity pairs within the KG through chord illustrations, serving as both Subjects

and Objects of predicative triples. The size of the chords corresponds to the support of the depicted relation[46].

[46]For the sake of visualization, we pre-filtered for relations with a minimum number of 20 occurrences in the dataset.

Chapter 4

Mapping the AECO Research Landscape

4.1 Building Scientific Knowledge Graphs

KGs have proved effective for representing research knowledge discussed in scientific papers and patents across several different domains [DORR+21, SBWL21, XXT23]. New generation "scientific KGs" have shifted from representing purely bibliographic information of research publications to support the construction of extensive networks of machine-readable information about entities and relationships pertaining to a certain domain, enabling fine-grained semantic queries over large scientific text collections of the form: "retrieve all methods that are used for Indoor Air Remediation in the time range t".

Therefore, they can support downstream analytical services like technology trend analysis. For example, [LHOH18] uses the statistics of relation triples of type $< Method; Used-for; Task >$ automatically extracted from paper abstracts to reconstruct historical trends of the top applications of target methods such as *artificial neural networks* in different areas like speech recognition and computer vision.

The generation of large-scale and accurate knowledge graphs of scientific knowledge presents various challenges, specific to the content and style of scholarly communication. Keeping aside the architectures proposing assistant tools for human expert KG editing ([JOF+19]) which are typically not scalable, systems based on automatic IE pipelines have to solve generally more demanding tasks than in the general domain, due to the following reasons:

1. the domain entities to be detected are generally *terms*, rather than named entities (e.g. *Geothermal Heat Pump System* as opposed to a named Person entity like e.g. *Bill Gates*). Terms are more abstract concepts that are linked to each other by (possibly multiple) subclass relations. For example, a *Geothermal Heat Pump System* is an instance of the more general class *Heating, Ventilation, and Air Conditioning System*. Consequently, there are multiple ways to refer to the same

terms at various levels of granularity, making mention detection more difficult.

2. terms are often referred to via domain-specific and often ambiguous acronyms (e.g. *HVAC system* for *Heating, Ventilation, and Air Conditioning system*);

3. extracting information from scientific articles requires extracting relations across sentences using co-reference resolution, while the application of standard sentence-level IE systems would result in lower relation coverage and would generate sparse KGs.

Methods for automatic generation of scientific KGs typically build upon supervised Relation Extraction (RE) models for capturing fine-grained relations between scientific entities in text [ZDY+23]. In the scholarly domain, the most influential entity and relation schema specification for this task is defined by the SciERC initiative [LHOH18][1].

The SciERC annotation schema defines six types of scientific entities, namely *Task*, *Method*, *Metric*, *Material*, *OtherScientificTerm* and *Generic* and five directed relation labels (*Used-for*, *Evaluate-for*, *Part-of*, *Feature-of*, *Hyponym-Of*) plus two symmetric relations (*Compare* and *Conjunction*)[2]. Moreover, co-reference links are annotated between identical scientific entities within and across sentences.

As an illustration of this schema, Figure 4.1.a shows the annotation of an NLP paper fragment, where the term "probabilistic context-free grammar (PCFG)" is detected as *Method* and linked to the *Method* "MORphological PArser MORPA" via *Used-for* relation, encoding the fact that the former is applied within the latter. Notice also that an entity of type *Generic* (the pronoun "it") is annotated as it is involved in a relation and is co-referred with another entity (that is the MORPA parser, edge not shown here).

The SciERC initiative published also a training dataset, comprising annotations of 500 scientific abstracts, taken from conference/workshop proceedings in various AI sub-communities. This work has laid the ground for several recent efforts towards the creation of large-scale scholarly KG infrastructures, such as CS-KG 2.0 [DOB+25] and AI-KG [DORR+20]. However, the availability of trained IE tools and training dataset has limited until now these initiatives to Computer Science, AI and related domains.

As part of a joint initiative on innovation intelligence analytics for the Architecture, Engineering, Construction and Operation (AECO) research and industry, carried out in collaboration with the Institute of Data Science and Artificial Intelligence of the University of Navarra, we designed and experimented with a hybrid workflow aimed at making sense of the recent research agenda in the AECO domain by identifying dominant research areas and tracking their evolution over time, providing quantitative grounding for insights on trending research directions and research foresight.

The proposed method organizes the vast domain of AECO research into macro-topics: then, for each macro topic, we leverage information extraction technologies

[1] `https://nlp.cs.washington.edu/sciIE/`

[2] The meanings of the entity and relation types are largely self-explanatory; however, for a detailed description, please refer to the full SciERC specifications: `https://nlp.cs.washington.edu/sciIE/annotation_guideline.pdf`.

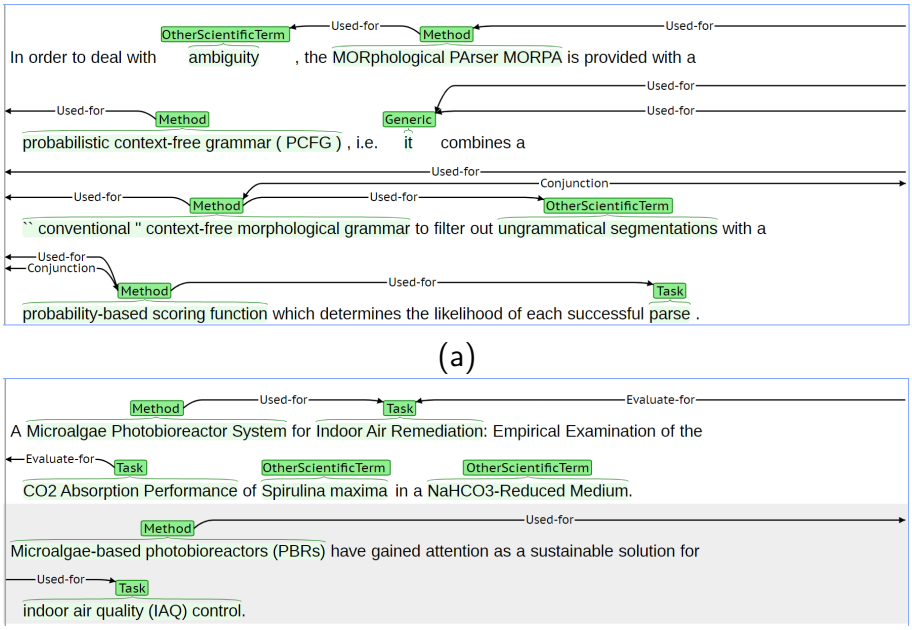


Figure 4.1: Sample Entity and Relation annotation from the SciERC dataset (a) and from an AECO paper abstract [HPKY23] (b), following the SciERC annotation schema. The latter shows a *Method* "Microalgae Photobioreactor System" and *Task* "Indoor Air Remediation" extracted respectively as the head and tail of a *Used-for* relation, resulting in the triple ⟨Microalgae Photobioreactor System;*Used-for*;Indoor Air Remediation⟩, which encodes the claim that a "Microalgae Photobioreactor System" method is applied to solve/achieve the task of "Indoor Air Remediation".

backed by deep learning models and LLMs to detect and track domain research entities such as Tasks and connect them with the Methods that have been proposed over time to solve them. To the best of our knowledge, this is the first attempt to apply semantic technologies to support data-driven innovation and technology intelligence at a large scale in the AECO domain.

4.2 Methodology Outline

The methodology adopted in this study is structured into four main phases, as illustrated in Figure 4.2. First, we collect research papers from OpenAlex using its API (Data Collection). Subsequently, we use the BERTopic architecture and domain expertise to consolidate a set of optimized macro topics (Topic Modeling). Finally, for each macro topic, we carry out two parallel processes: a Bibliometric Analysis, generating collaboration network visualizations, and an Information Extraction analysis, using an adapted version of the SKG pipeline [DOR+22c] and producing trend analyses backed

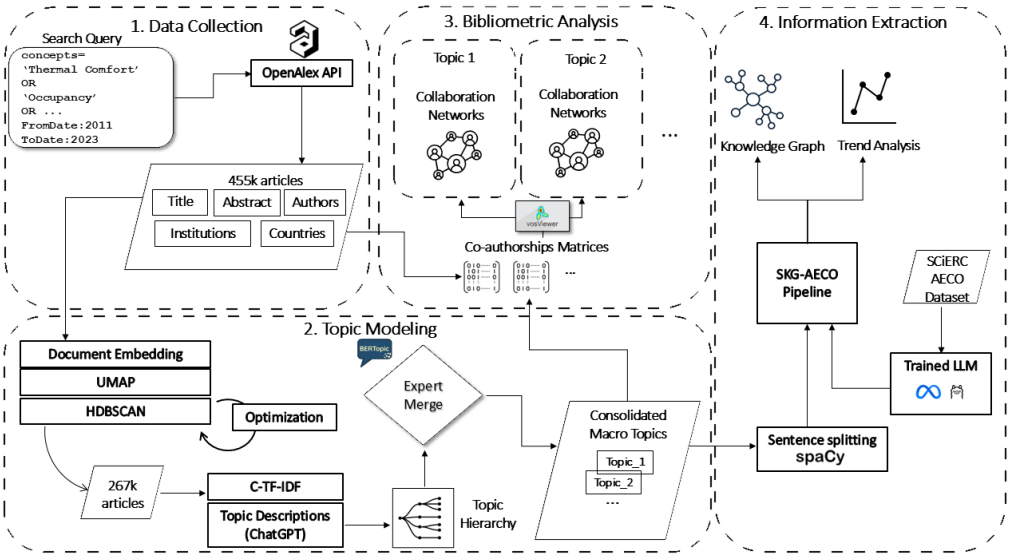


Figure 4.2: Flowchart of the text processing pipeline used in this research. The pipeline is structured into four main phases: 1. Data collection from the OpenAlex API; 2. Topic Modeling using optimization of BERTopic architecture and expert tuning; 3. Bibliometric analysis of collaboration networks for each consolidated topic; 4. Extraction of scientific Knowledge Graphs and generation of trend analysis for each consolidated topic, using the SKG-AECO pipeline.

by relational graphs. We provide here a detailed description of phases 1, 2 and 4, which are more closely related to the subject of the present thesis.

A complete collection of interactive visualizations for all the analyses conducted in this study is made publicly available in a dashboard accessible at: `https://huggingface.co/spaces/zavavan/AECO_Tech_Dashboard`.

Details on how to interact with and customize the single visualization panels are provided within each corresponding section in the rest of the chapter.

All the code and resources generated in this project and enabling the reproducibility of the analysis are shared in a public GitHub repository (see Appendix C for details).

4.3 Data Collection

The research publications used for our analysis were sourced from the OpenAlex[3], a fully open and high-coverage scientific graph database [PPO22].

Created in 2022 as a replacement of the discontinued Microsoft Academic Graph (MAG) [SSS+15], OpenAlex consists of a large directed graph connecting five types of scholarly entities, namely "*works*" (234M) , "*authors*", "*venues*" (i.e., journals, conferences, etc.), "*institutions*", and "*concepts*". The nearly 65k OpenAlex "*concepts*" are

[3] `https://docs.openalex.org/`

topic labels automatically assigned by a text classifier based on titles and abstracts of the publications. They are organized into a 5-level hierarchy with 19 root-level concepts.

For our analyses, we sampled a base dataset by querying the OpenAlex API, retrieved all English-language papers published between 2010 and 2023 that were tagged with at least one concept label from a set of expert-selected terms, representing the entire AECO domain. The complete list of these concept tags, along with the corresponding coverage in terms of retrieved works, is provided in the "*resources*" folder of the code repository.

This resulted in a collection of around 466k publications. For all the following text-based analyses, we process titles and abstracts of the papers.

Figure 4.3 below shows monthly time series of the total number of publications and the number of publications per country, for the top 15 countries by number of publications in the dataset[4]. These top 15 publishing countries account for around 50% of the total number of papers.

A complete visualization of country-based publication time series is available in the "Publication Trends" panel of the web dashboard.

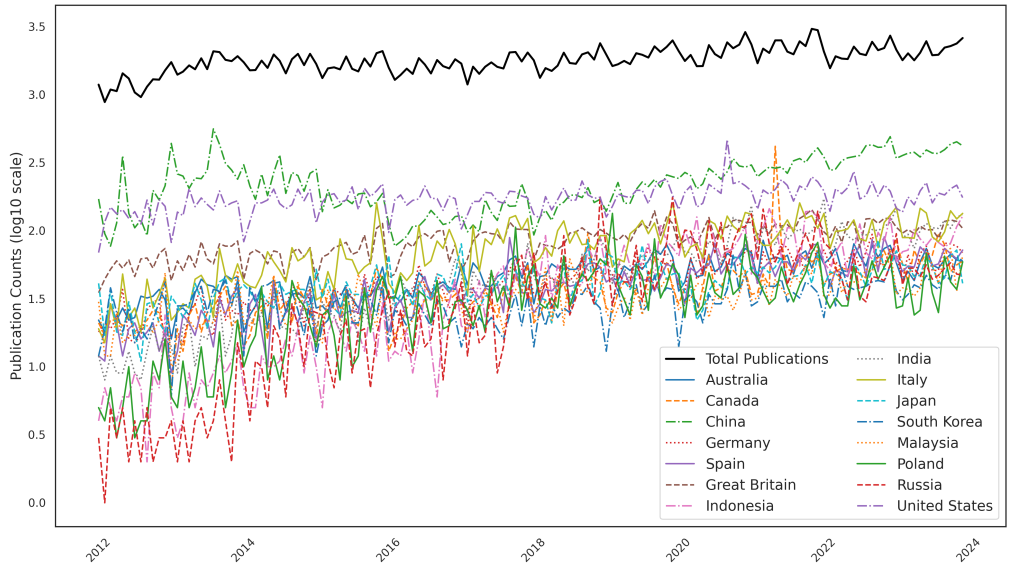


Figure 4.3: Log scale monthly time series of the overall publications and publications per country, for the 15 top publishing countries.

[4]In the OpenAlex database, the publication countries are the countries of the institutional affiliations that the publication authors claimed in the context of that work, extracted using a combination of matched institutions and parsing of the raw affiliation strings, in case specific institutional affiliation metadata are missing

4.3.1 SciERC AECO Dataset

From this base dataset, we sampled around 150 items (concatenated Title and Abstract), pre-processed and sentence split them using Spacy's English transformer pipeline *en_core_web_trf-3.6.1*[5] and sampled a final set of 1,016 sentences. We annotated the sentences for scientific entities and relations, based on a subset of the SciERC annotation schema from Section 4.1. Namely, we annotated for scientific entities of type *Task*, *Method* and *Metric*) and for the relations of type *Used-for* and *Evaluate-for*. The annotations were carried out independently by two domain experts using the Brat annotation tool [SPT+12] and we measured the inter-annotator positive specific agreement relative to the entity detection sub-task ([HR05][6]), obtaining a mean F1 score of 0.73. This indicates a significant agreement between the human annotators, although some marginal task ambiguity emerged (see Section 4.6).

Figure 4.1.b illustrates a sample visualization of SciERC AECO annotations, while Table 4.1 summarizes some statistics of the dataset. We publicly share the current version of the dataset (called SCIERC AECO) in the Hugging face dataset hub: `https://huggingface.co/datasets/zavavan/scierc_aeco`.

	Train	**Test**
Sentences	816	200
Negative Sentences	536	124
`Task` Entities	512	123
`Method` Entities	388	89
`Metric` Entities	51	22
`Used-for` Relations	196	42
`Evaluate-for` Relations	18	8

Table 4.1: Summary counts of the SciERC AECO dataset.

4.4 Detecting Topic Clusters

In order to enhance the granularity of the KG-based research trend analysis we describe in Section 4.5, we integrate an upstream optimized topic model in the presented pipeline. This scalable approach makes the downstream IE module more sensible to low-level signals which would not emerge at the full collection level. As we will show in Section 4.5, computing topic-specific statistics of the generated triples allows to extract very specific *Task* and *Method* entities, such as highly specialized tools and software packages.

Topic modeling is an unsupervised machine learning technique used to automatically discover abstract topics within a collection of documents. We use the BERTopic neural

[5] `https://github.com/explosion/spacy-models/releases/tag/en_core_web_trf-3.6.1`

[6] This corresponds to classical Cohen K inter-rater agreement, in tasks like NER where the number of negative cases is undefined.

topic modeling algorithm to identify the main latent topics emerging from our paper collection [Gro22].

The main feature setting apart BERTopic from more classical unsupervised topic modeling methods, such as LDA, is the use of contextual and fixed-sized dense vector representations of the input documents through embedding models, instead of static "Bag-of-Word" representations, which fail to account for semantic relationships among words.

The algorithm is highly modular and the sequence of processing modules can be split into two main parts. The the first one responsible for document clustering and encompasses:

1. a document embedding step, creating multidimensional vector representations of documents using the encoding of pre-trained embedding models;

2. dimensionality reduction of the document embedding vectors.

3. clustering of the compressed embeddings and detection of groups of semantically similar documents.

Once document clusters are created, the goal of the topic model is to detect and represent latent themes in each cluster, which is done sequentially by:

1. computing a distribution with respect to document clusters of the word/terms in the whole dataset vocabulary;

2. extracting of the best-representing terms for each cluster.

This general BERTopic pipeline can be customized as each module is largely independent of the others. This allows us to optimize for a module configuration that achieves enhanced performance in our AECO domain dataset.

Namely, for the document embedding step, we encode the concatenation of Title and Abstract of each research paper into a vector representation generated by the Sentence Transformer model "all-mpnet-base-v2"[7].

We compute document clusters by optimizing the hyperparameters of a combination of the UMAP and HDBSCAN algorithms, similarly to what described in Section 3.7.

In this case, hyperparameter grid-search is carried out optimizing with respect to the average coherence of the resulting topics. Assuming a topic is defined as a set $T = \{t_1, t_2, ...t_n\}$ of terms (tokens or n-grams) that best describe the latent theme of a document cluster, the topic coherence we used here estimates the reciprocal support between topic terms as the average cosine similarity between the probability vector of

[7] `https://huggingface.co/sentence-transformers/all-mpnet-base-v2`. This embedding model maps sentences and paragraphs to a 768-dimensional dense vector space and is optimized for semantic similarity tasks, providing strong out-of-the-box performance even on technical or semi-specialized domains.

each term $t_i \in T$ and the probability vector of all other terms $t_j \in T$ in the topic, that is:

$$\text{Coherence}_{c\_v}(T) = \frac{\sum_{i=1}^{n} \tilde{m}_{\text{cos}}(\{t_i\}, \{t_j \in T \mid t_j \neq t_i\})}{n}$$

where for each t_i, t_j:

$$\tilde{m}_{\text{cos}}(t_i, t_j) = \text{cosine\_sim}(\vec{\mathbf{v}}_{\text{prob}}(t_i), \vec{\mathbf{v}}_{\text{prob}}(t_j))$$

and the probability vector of a term subset is calculated as:

$$\vec{\mathbf{v}}_{\text{prob}}(W') = \{ \sum_{w_i \in W'} \text{prob}(w_i, w_j) \}_{j=1,2..,|N|}$$

where $\text{prob}(w_i, w_j)$ is estimated based on term co-occurrence counts over a moving sliding window in the dataset corpus [RAJS22]. This method of assessing topic coherence has been shown to correlate fairly well with human judgment [SBA+22].

Based on this metric, we proceed as follows. First, to discard marginal topic clusters in the original paper collection (possibly due to inaccuracy of the openalex concept tag filtering), we run BERTopic with default settings on the whole dataset, manually inspect the result topics descriptions, discard out-of-domain topics and filter out all papers belonging to discarded topics[8], ending up with 267,319 items.

Subsequently, we find an optimal hyperparameter configuration for a 10,000 random document sample in order to estimate the optimal level of granularity of the clustering. Figure 4.4 plots the average topic coherence against the number of clusters for a subset of best-performing UMAP and DBSCAN hyperparameter configurations, showing that optimal coherence scores are reached for a configuration distributing the data sample into slightly more than 30 clusters.

Using this parameter settings on the entire dataset results in a total of 52 clusters, which are visualized in a reduced 2-dimensional space in Figure 4.5, with the descriptive labels for each cluster being generated by fine-tuning the topic representations via the use of a LLM (see further down in the Section for details)[9]. An interactive visualization of the topic modeling data map is made available in the "Topic Map" panel of the web dashboard, where the titles of article data points from the entire dataset can be accessed by hovering over them, while clicking on single data points redirects to the corresponding OpenAlex paper entry page[10].

The BERTopic's default topic representation is based on the c-TF-IDF weighting schema, a class-based adaptation of the classical TF-IDF that estimates the statistical

[8]Larger discarded clusters were mostly related to biological ecosystems, transportation, signal processing and textile industry.

[9]For simplicity, we only show the descriptive labels for the 10 largest clusters and use numerical labels for the others. The full list of topic labels is stored in the *OptimizedAECOTopicLabels.tsv* file in the folder "*resources*" folder of the code repository.

[10]A search function also allows to retrieve single papers based on keywords in the Title and Abstract.

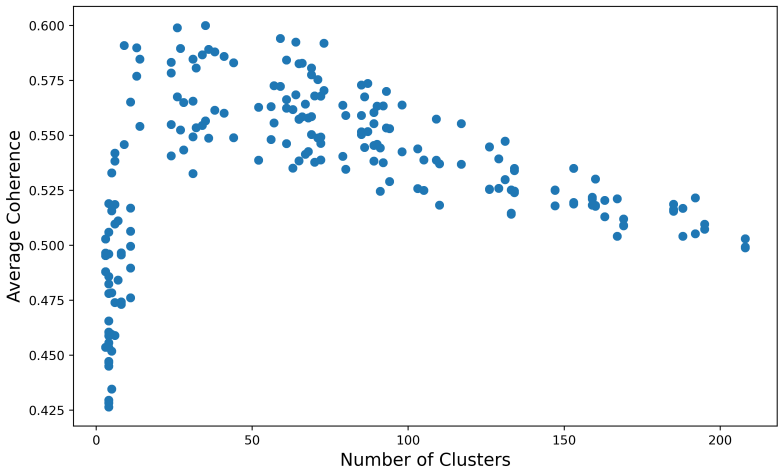


Figure 4.4: Average topic coherence values against the number of clusters for a subset of best-performing UMAP-HDBSCAN hyperparameter settings.

relevance of a term t for a class c as:

$$W_{t,c} = \mathsf{TF}_{t,c} \cdot \log\left(1 + \frac{A}{\mathsf{TF}_t}\right)$$

where $\mathsf{TF}_{t,c}$ is the frequency of the term t in the concatenation of all documents comprising the topic class c, A is the average number of terms per class and TF_t is the total frequency of t over all classes.

In our settings, the vocabulary comprises the 1,000 most frequent unary terms (excluding English stop words) and an initial topic representation for each class is generated, that includes 10 vocabulary terms with the highest c-TF-IDF score. These terms are used to sample a subset of 10 most representative documents per class. Then, following the KeyBERT keyword extraction technique ([Gro20]), both base terms and representative documents are embedded with the Sentence Transformer model, and vectors are compared to generate a 10-term fine-tuned topic representation. Finally, for each class, the 10 fine-tuned keywords and 10 documents are passed in as variables for a text generation prompt to the OpenAI "gpt-3.5-turbo" model (aka ChatGPT, [RNSS18]) API, using parameters $temperature = 0.0$ and $diversity = 0.6$. The prompt is shared at the URL: `https://github.com/zavavan/AECO_KG_Pipeline/blob/main/resources/gpt3.5_topic_representation_prompt.txt`.

Figure 2 in Appendix C shows a heat map of the coherence scores for the optimized topics, labeled using ChatGPT-generated descriptions.

The average topic coherence score is 0.613, with values ranging from 0.46 to 0.74 (standard deviation 0.067), which typically denotes fairly consistent topics [RBH15, LNB14]. However, by embedding the c-TF-IDF representations of the topics and projecting them onto a reduced 2-dimensional embedding space (as shown

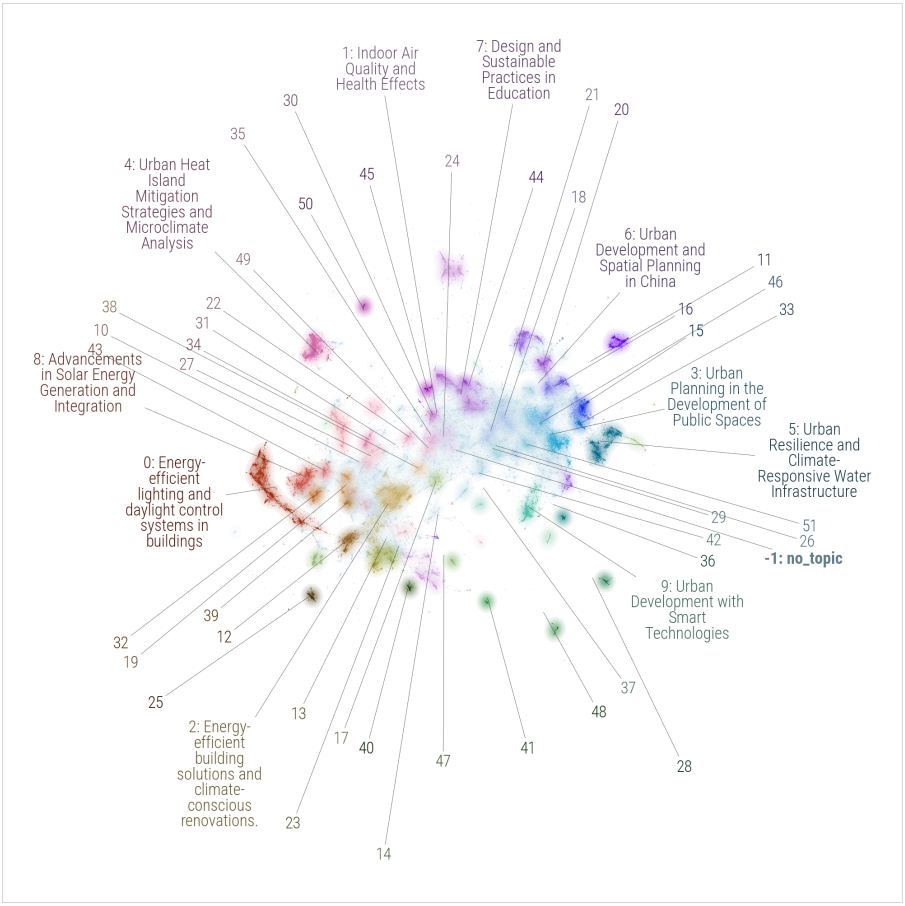


Figure 4.5: Reduced 2-dimensional visualization of the optimized 52 topic clusters of research papers, embedded using a Sentence Transformer model. The indicator lines originating from cluster labels (laid out here in rings around the data map, for clarity) point to each cluster's medoids. The *-1:no_topic* label denotes the set of outlier articles.

in Figure 3 in Appendix C), one can notice that there are macro groups of clusters that share several semantic features and stand apart from each other.

We verified this observation by carrying out a manual inspection of the hierarchical structure of the generated clusters. Figure 4 in Appendix C shows the dendrogram representation of the optimized clustering. One can notice, for example, that the sub-tree colored in green at the top of the dendrogram contains rather contiguous topics (indexes 0,2,4,12,13,14 and 25) all related to energy efficiency technologies for building thermal comfort. We leverage such a hierarchical representation to merge several topic subsets placed close to each other at higher levels of the dendrogram, resulting in a number of macro clusters (16) very close to the one emerging from Figure 3. Therefore, we consolidate the topic modeling using these 16 macro clusters, which are listed in Table 3 in Appendix C together with cluster size, natural language description labels generated by ChatGPT, and the topic base representations comprising the most relevant

10 terms.

4.4.1 Analysis

Sixteen dominant macro topics in AECO research emerged from the topic-modeling analysis, reflecting a diverse set of research interests centered around sustainability, urban planning, and technological advancements. The prominence of energy efficiency, smart cities, and sustainable construction indicates a strong focus on climate-aware design. The inclusion of BIM, solar integration environments, and acoustics suggests growing interest in digitalization and environmental quality in built spaces. The resilience of cities to climate change and the preservation of traditional architecture highlight a balance between innovation and heritage conservation. Finally, topics like urban agriculture and heat transfer show interdisciplinary expansions beyond conventional AECO research.

Figure 4.6 visualizes the evolution over time of the 16 identified AECO macro topics by showing semi-annual time series of the absolute numbers of publications per topic, in the time range 2012- early 2024. Energy Efficiency and Thermal Comfort remains the dominant topic, showing continuous growth until 2023, reflecting global sustainability efforts. Indoor Air Quality and Sustainable Air Conditioning Systems follows closely, likely due to increasing worldwide air quality concerns in cities. Between 2018 and 2020, Smart City Development and Urban Resilience overtook Child-Friendly Urban Spaces, indicating a broader shift from socially oriented to technology-focused urban planning. Landscape Planning declined from 2013 to 2020, with a notable spike in interest in 2020. Meanwhile, Topics 10–15, including BIM and Sustainable Materials, show static trends, indicating fields with steady but limited research interest.

An interactive version of the plots can be accessed in the "AECO Macro Topic Trends" panel of the web dashboard.

4.5 Information Extraction Pipeline

Once the entire research paper dataset is partitioned into topic-coherent clusters, a semantically richer approach to exploring the AECO research landscape can be achieved by explicitly representing the content of the papers in each cluster through a KG of scientific entities [AKP+18], using the SciERC schema presented in Section 4.1.

As we mentioned earlier, SciERC data schema is backing the implementation of NLP pipelines for the automatic generation of accurate, large-scale scientific KGs, such as SCICERO [DOR+22c]. We describe in the following the steps we carried out for adapting SCICERO's pipeline, originally designed for the Computer Science domain, to extract (a subset of) SciERC ontology entities and relations from our collections of AECO paper abstracts. The resulting pipeline is named *SKG-AECO*. Moreover, we show how the resulting graph data infrastructure can be queried to generate analytical insights and trend analyses on the AECO domain. We will use macro topics 0, 6 and 12 as illustrative examples.

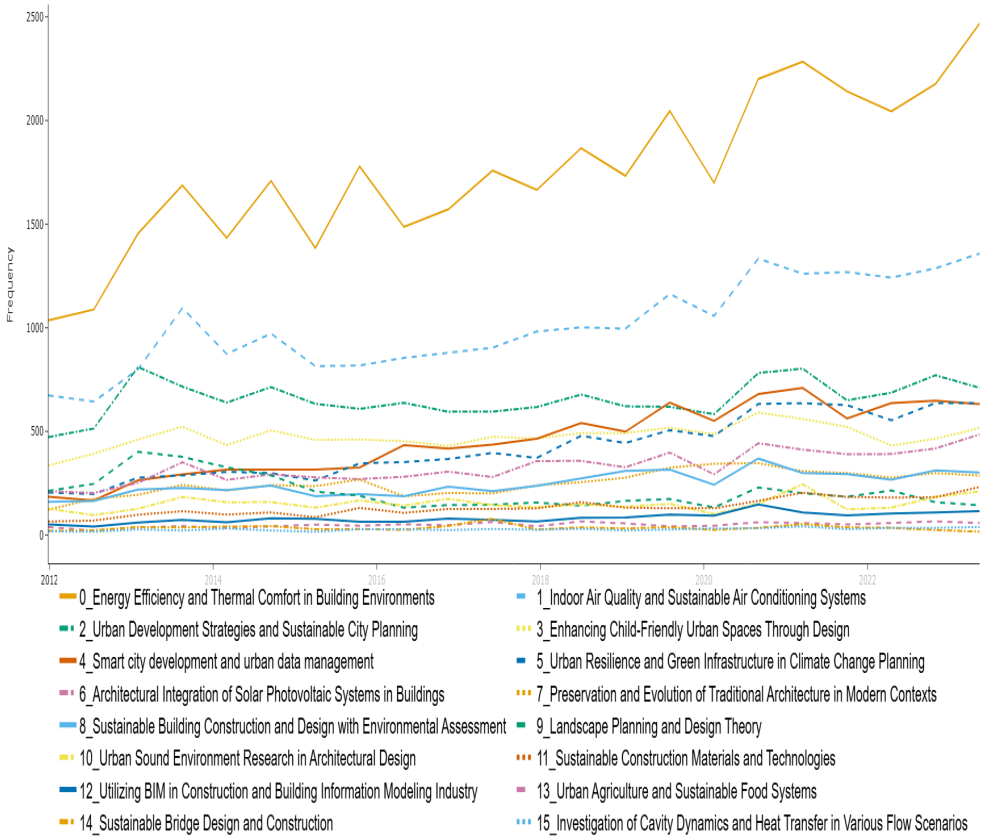


Figure 4.6: Evolution over time of AECO macro topics.

4.5.1 SCICERO

SCICERO is a hybrid NLP pipeline using a blend of deep learning trained models, rule-based heuristics and semantic web techniques to generate a large-scale KG [DORR+22] of 41M statements among 10M unique entities from a dataset of 6.7M abstracts in Computer Science[11]. It comprises three main components:

1. A set of extraction modules responsible for generating candidate entities and candidate relations connecting them;

2. Entity and relationship handling, where entities are normalized and merged by linking to external knowledge bases.

[11]The full ontology underlying this KG includes entity types `Task`, `Method`, `Material`, `Metric` and `OtherEntity` and 179 object properties.

3. Triple validation, where final output triples are selected based on consistency with the most reliable ones (high support triples[12]) and an ontological schema.

For further details on the SCICERO pipeline, refer to the original paper [DOR+22c]. The main source of candidate triples are models that have been trained in scientific text collections in the Computer Science domain[13]. However, state-of-the-art models for scholarly Relation Extraction, trained on the SciERC dataset comprising AI/ML articles, do not perform well on new target domains, such as AECO.

For example, in a previous study ([ZGSC24]) we experimented with the SOTA architecture SpERT (see Section 2.5), by re-training it on SciERC train split (1861 sentences) using SciBERT (cased) embeddings [BLC19a] and testing it on our small out-of-domain gold standard SciERC AECO. We observed that the performance degrades drastically from around 70% to as little as 18% for NER, and from 50% to 7% for RE. In the same study, we also indirectly proved that few-shot learning methods for ChatGPT generate low quality data for the SciERC task. In other words, a domain adaptation process is needed for the KG extraction models to work effectively on AECO data.

Therefore, we enhance the original pipeline with an Information Extraction module trained on target entity and relation instances from the SciERC AECO dataset.

4.5.2 LLMs for structured information extraction

Typically, deep learning RE models require a large number of training data, whose annotation requires costly domain expertise [ZDY+24]. However, more recently, decoder-only LLMs, pre-trained on massive open-domain data volumes, have proven able to learn complex information extraction tasks in scientific domains from a small number of examples [BMR+20]. Here, we adapt a method successfully applied in the material science domain ([DDL+24]) to the fine-tuning of an LLM for a joint NER/RE task using only a few hundred structured prompt examples and no explicit schema definition. For each of the 816 SciERC AECO training set instances, an instruction tuning prompt-completion pair is generated as shown in Figure 4.7, where the LLM is prompted to extract target entities and relations from an input sentence and it is simply presented with sample output in the form of json-style annotation.

Using the resulting dataset, we perform instruction fine-tuning on the smallest release of the Llama 2 foundation model (`meta-llama/Llama-2-7b-hf` [TMS+23b]) using Hugging Face's `Trainer` class and `PeftModel` class implementation for wrapping the base Llama model to trainable PEFT weight decomposition matrices. Training is done on 6 epochs with a 8 batch size. We reached a minimum validation loss of 0.162, with an average of 0.3% unparsable completions, indicating that the trained model achieves good performance on the task and provides structurally consistent output.

[12] Support refers to the number of papers from where a triple was extracted, or the number of methods that generated it.

[13] Namely, the Dygiepp model ([WWLH19]) and a number of relation extraction modules building on top of Dygiepp-generated entities.

Zero-Shot Prompt

```
{  "sentence_text": "Water bio-remediation through a probiotic
                        layer system.",
   "Tasks": { "T1": "Water bio-remediation" },
   "Methods": { "T2": "probiotic layer system"},
   "Metrics": {},
   "Used-for": [
      { "T2": "T1"  } ],
   "Evaluate-for": [],
   "relevant": "true"
}
```

```
{  "role": "user",
   "content": "From the following Text, extract non-overlapping
   entities of type Tasks, Methods and Metrics and extract
   Used-for relations between Methods and Tasks
   and Evaluate-for relations between Metrics and Methods.
   Text: Water bio-remediation through a probiotic layer
   system."},
 {"role":"assistant",
 "content": {"Tasks": {"T1": "Water bio-remediation"},
                    "Methods": {"T2": "probiotic layer system"},
                    "Metrics": {},
                    "Used-for": [{"T2": "T1"}],
                    "Evaluate-for": []
                    }
}
```

Figure 4.7: Sample conversion of a SciERC AECO training instance into a prompt-completion pair for instruction fine-tuning using json-style annotation formalism. $T1$, $T2$ are entity indexes. Notice that no definitions of Entity and Relation semantics are provided in the instruction part of the prompt.

4.5.3 SKG-AECO

Figure 4.8 depicts in detail the *SKG-AECO* processing pipeline.

On the total sample of 48,972 articles from macro-clusters 0, 6, and 12, the *SKG-AECO* pipeline extracts a total of around 700k raw triples, with a contribution of 22.9% of triples coming solely from the added LLM module (referred to as E_{LLM} in the Figure). The triples undergo a multi-step process of validation, filtering, and merging, so that entities and relation predicates that refer to the same concepts are mapped to a unique representation.

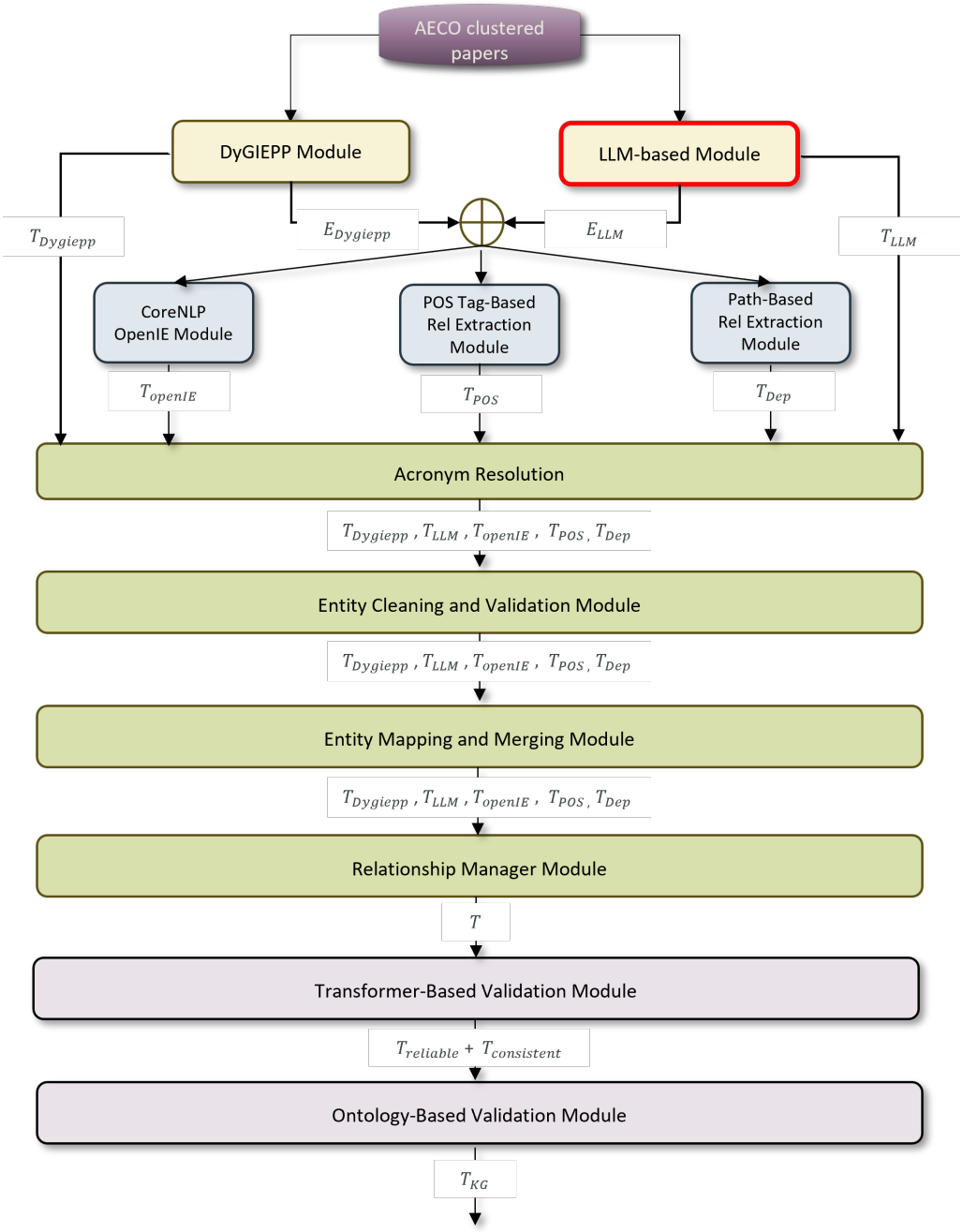


Figure 4.8: A detailed flowchart of the *SKG-AECO* pipeline.

Cluster-level acronym resolution At the article level, acronyms are solved and mapped to a standard form by exploiting the regular expression patterns where they appear together with their extended forms, as in: *computational fluid dynamics (CFD)* or *building information modeling (BIM)*. However, this limits the recall as highly standardized acronyms in a domain (like *BIM*) typically appear by themselves, outside any

of those expanded pattern. By processing documents per topic cluster, we can take advantage of the acronyms' low ambiguity level and safely apply the acronym mappings collected globally, at the cluster level. In this way, we raise by 21.1% the number of acronym-mapped entities, resolving 19,798 acronyms out of a total of 706,614 raw entities (2.8%).

Linking to external knowledge bases By linking to external knowledge bases, different candidate entities bearing the same meaning can be merged into a single entity. For example, *"building insulation material"* and *"insulating building material"* are both linked to the same Wikidata[14] entry `http://www.wikidata.org/entity/q28942423` and thus merged together. The mapping is performed in two parallel ways: first, we collect the Wikidata entries mapped from all OpenAlex *concept* metadata associated with our entire research papers dataset, generating a dictionary resource of 12,700 AECO domain entries. Then, if a candidate entity string matches the main label or alternative label (i.e., labels linked by the property *rdfs:altLabel*) of a Wikidata entry in the dictionary, it is mapped to that entry. Finally, the longest variant of all entities mapped to the same Wikidata entry is chosen as the representative label, and all other variants are replaced by it in the generated triples. Secondly, we run a SPARQL query retrieving Wikidata entities corresponding (via *owl:sameAs* link) to DBpedia entities that belong to any of the subject categories *dbc:Architecture*, *dbc:Building*, *dbc:Construction*, *dbc:Engineering*, *dbc:Operations* or their sub-categories (via *skos:broader* relation).

Out of 222,633 candidate entities, 3,790 (1.7%) and additional 765 (0.34%) are mapped to Wikidata through the first and second method, respectively[15].

Transformer-based merging In the SCICERO pipeline, entities that are not linked to external resources undergo a merging process based on cosine similarity of their corresponding vector representations, derived from *paraphrase-distilroberta-base-v2* embedding model[16]. After some tests, we empirically lowered the similarity merging threshold to 0.75, while keeping the remaining mechanism unchanged.

Triple selection By the entity merging above and relation ontology mapping mechanisms (as detailed in the SCICERO paper), the raw triples get reduced to 307,627 generalized triples. As a last step before graph generation, triples get further refined by taking:

- the set $T_{reliable}$ of reliable triples, that is triples extracted from a minimum number of articles s_1 and by a minimum number of extractor tools s_2, with s_1 and s_2 empirically set to 3 and 2, respectively;

[14] `https://www.wikidata.org/wiki/Wikidata:Main_Page`

[15] As computational methods are pervasive across the AECO domain, we also keep from the original SCICERO pipeline the module mapping entities onto the Computer Science Ontology (CSO [SOTM19]). This generates an additional 1,129 (0.5%) entity mappings.

[16] `https://huggingface.co/sentence-transformers/paraphrase-distilroberta-base-v2`

- an additional subset $T_{consistent}$ of non-reliable triples which are classified as consistent with $T_{reliable}$ by a transformer-based binary classifier[17] fine-tuned on around 16,000 positive examples sampled from $T_{reliable}$ and 16,000 synthetically generated negative examples.

The union of $T_{reliable}$ and $T_{consistent}$ forms the set of 208,462 final triples T_{KG}, about 29% of the initial raw triple set.

4.6 Evaluation

In order to assess the reliability of our IE approach, we performed an evaluation of the precision of the generated triples. Out of the entire range of triple output by the original SCICERO pipeline, ⟨*Method; Used-for; Task*⟩ and ⟨*Method; Used-for; Method*⟩ are the types that we will use for trend analysis in see Section 4.7.2. Therefore, we restrict the evaluation to these two types of SciERC triples.

We randomly sampled 300 triples, evenly distributed from the high-support and low-support groups (support ≥ 5 and ≤ 2, respectively), and assigned them to a total of 12 independent, AECO domain expert annotators. Each triple was assessed by three annotators as True or False, where a True label was assigned when: 1) the triple Subject and Object entities are linked by a *Used-for* relation in the text sentence; 2) the triple Subject and Object are fully extracted, i.e. they do not contain less information than what is stated in the text; 3) both the triple Subject and Object are correctly classified as Task/Method.

We calculated inter-rated agreement using Gwet's AC1 coefficient [Gwe08]. AC1 generalizes to annotation tasks where each item is assessed by only a subset of the annotators pool, while additionally using a corrected baseline of expected (chance) agreement that is more robust to label class imbalance[18], than the standard Cohen's κ and Krippendorff's α. The resulting agreement score is 0.361 (overall observed agreement 0.670, Gwet chance agreement 0.484), which is commonly viewed as a "fair" agreement level but still signals a residual element of unreliability of the annotators' judgments.

Consequently, we performed an outlier analysis by computing per-annotator Cohen's κ agreement with respect to the majority vote labels and identified a subgroup of 3 unreliable annotators with lower κ values in the range 0.35-0.38, (versus ≥ 0.54 values for most of the annotators). By removing this subgroup, the resulting AC1 score rises to 0.507, which is typically interpreted as "moderate" agreement.

Overall this indicates that, while not outstanding in absolute terms, the validation reliability is driven down by poor compliance to annotation instructions by a few outlier annotators, rather than by an inherent difficulty or subjectivity of the task.

Table 4.2 displays the Precision score values for the overall 300 manually triples, as well as for the subsets of high support, low support, and for the ones generated solely

[17] Based on *scibert_scivocab_uncased* [BLC19b].

[18] In our case, 58.7 and 41.2% True and False, respectively.

by the AECO-customized LLM.

Extraction Method	Precision
Random Triples	0.695
High Support	0.758
Low Support	0.637
LLM-generated	0.683

Table 4.2: Triple evaluation over a set of 300 triples.

Overall, the precision of the method is in line with the results from the original SCICERO pipeline and robust across all configurations.

Analogously to the original SCICERO pipeline, triple precision correlates with triple support, with high support triples reaching a 0.76 precision level [DOR+22c].

Through error analysis, we found that a significant part of the error rate was due to the Transformer-based entity merging method. In fact, we boosted entity merging by lowering the similarity threshold from 0.9 to 0.75, in order to enhance entity generalization and mitigate data sparseness for the subsequent trend analysis, given the relatively low size of our cluster datasets. However, this accounted for a significant share of invalid triples, such as for example $\langle urban\,heat\,island; uses; urban\,green\,space\rangle$ which was wrongly extracted from the sentence *"Urban greening through local to landscape management is a proposed strategy to combat UHI and improve environmental justice in city neighborhoods."*. In this case, the entity *combat UHI* was overgeneralized to *urban heat island* via acronym resolution and cosine similarity-based merging.

Finally, the novel LLM component shows a Precision level comparable to the SCICERO native extraction modules. Consequently, from the integration of the module, one can estimate a significant addition of true positive triples and a consequent recall gain.

4.7 Results

4.7.1 AECO Research Knowledge Graph

From the entire triple set generated by deploying the *SKG-AECO* pipeline, we select the subset of 255,764 triples of type ⟨*Method; Used-for; Task*⟩ and ⟨*Method; Used-for; Method*⟩ and make it publicly available as a preliminary version of the *AECO* knowledge graph.

The *AECO Research Landscape Knowledge Graph* (hereafter *AECO-KG*) connects 15,037 and 22,240 unique *Task* and *Method* entities via the object property *uses-Method* defined in [DOR+22c]. The ontology describing *AECO-KG* is an adaptation from SCICERO's[19], where classes and properties are defined in the namespace

[19] `https://scholkg.kmi.open.ac.uk/cskg/ontology#`

http://aeco-research.org/aecokg/ontology# (prefix *aeco-ont*), instances in the namespace http://aeco-research.org/aecokg/resource/ (prefix *aeco*) and the source papers from where triples originated are typed as *aeco-ont:OpenAlexPaper* entities with a data property *aeco-ont:hasOpenAlexURL* associating them with their unique URL in the OpenAlex platform.

Each claim encoded by one *AECO-KG* triple is reified into an instance of the *aeco-ont:Statement* class and associated with the collection of entities of type *aeco-ont:OpenAlexPaper* it was generated from (using the property *provo:wasDerivedFrom*[20]) and the cardinality of such collection (*aeco-ont:hasSupport*).

The *AECO-KG* graph together with its ontology definitions are accessible as a single RDF format serialization within the European Data portal[21]. Furthermore, we have set up a Virtuoso SPARQL endpoint where *AECO-KG* can be queried, and analytical information on target entities, attributes, and relations can be retrieved in user-specified data formats[22]. As an example, a SPARQL query like the one in Figure 4.9 returns the list of Tasks that are claimed to be solved using the green roof architectural solution (*aeco-ont:green_roof*), according to 1395 extracted triples. The query additionally extracts the URLs of the (899) research papers generating those triples.

```
PREFIX aeco-ont: <http://aeco-research.org/aecokg/ontology#>
PREFIX aeco: <http://aeco-research.org/aecokg/resource/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX provo: <http://www.w3.org/ns/prov#>

SELECT ?subject ?url
WHERE {
   ?stmt aeco-ont:subject ?subject ;
   aeco-ont:predicate aeco-ont:usesMethod ;
   aeco-ont:object aeco:green_roof .
   ?subject rdf:type aeco-ont:Task .
   ?stmt provo:wasDerivedFrom ?paper .
   ?paper aeco-ont:hasOpenAlexURL ?url . }
```

Figure 4.9: Sample SPARQL query returning *aeco-ont:Task* entities that are claimed to be applying the Method *aeco:green_roof* according to the *AECO* graph statements, together with the URLs of the papers supporting the claims.

[20]PROV-O - https://www.w3.org/TR/prov-o/

[21]https://data.jrc.ec.europa.eu/dataset/996d2a1b-69c9-4b27-b9b1-e0913f7f2d77

[22]The service is accessible at https://api-vast.jrc.service.ec.europa.eu/sparql/.

4.7.2 Trend Analysis

For each target macro-cluster, we perform trend analysis based on the output triples of type ⟨*Method; Used-for; Task*⟩ and ⟨*Method; Used-for; Method*⟩ extracted from the research papers belonging to that cluster. A complete visualization of trend analysis time series is available in the "Research Tasks and Methods Trends" panel of the web dashboard.

Figure 4.10 shows the historical trends of the most frequently occurring `Task` entities in triples from micro-cluster 0, representing the evolution of key applications in this AECO topic area. *Energy performance* has consistently been the most researched task throughout the period. The second most prominent task is *thermal comfort*, showing a steady upward trend that reflects the sustained focus on occupant-centric performance. While *daylighting system design* was initially among the top three, in recent years it has been surpassed by emerging tasks such as *urban heat island* and *outdoor thermal comfort*.

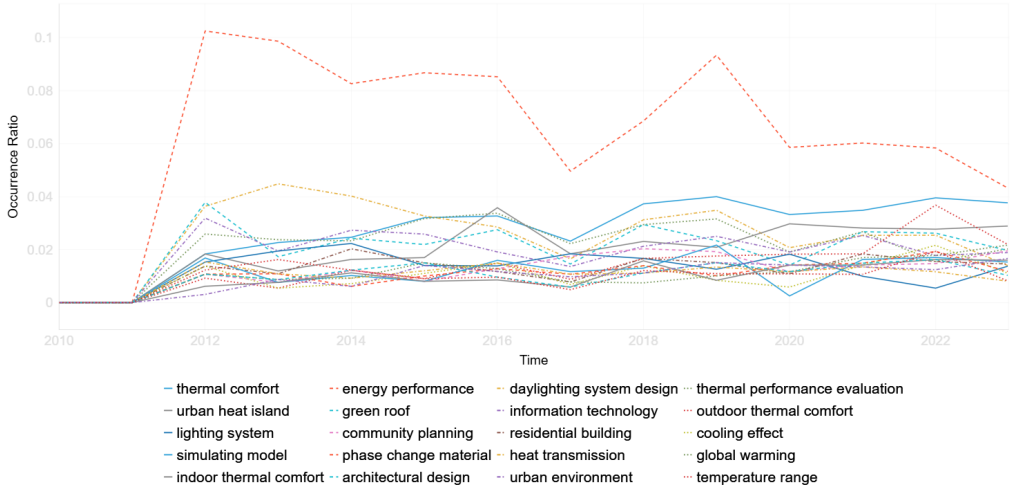


Figure 4.10: Trend analysis of the top 20 Tasks for macro-cluster 0, with the Tasks listed in the legend at the bottom of the plot. The y-axis measures the ratio of articles mentioning the Tasks to the overall number of articles in the cluster.

Analogously, Figure 4.11 shows the historical trends of the most frequent *Method* entities in cluster 0 triples, capturing the evolution over time of key technological solutions in this AECO topic area. *Information technology* remains the most widely used method across the entire period, with peaks around 2016 and continued relevance thereafter. It often appears in sentences where it functions as a tool or enabler, representing how the study was conducted (e.g., use of sensors, data systems, digital tools). From 2016 onward, *urban heat island* methods gained prominence, temporarily overtaking all others between 2018 and 2020. It often becomes a framing method, a contextual variable that structures simulations, models, or measurements. *Global warming* also shows a steady increase, becoming the third most used method in recent years. It acts as a guiding con-

text for scenario-based analysis or long-term forecasting, especially in energy demand, comfort evaluation, or mitigation strategies. Additional methods, such as *daylight* and *urban green space*, display consistent but lower usage over time.

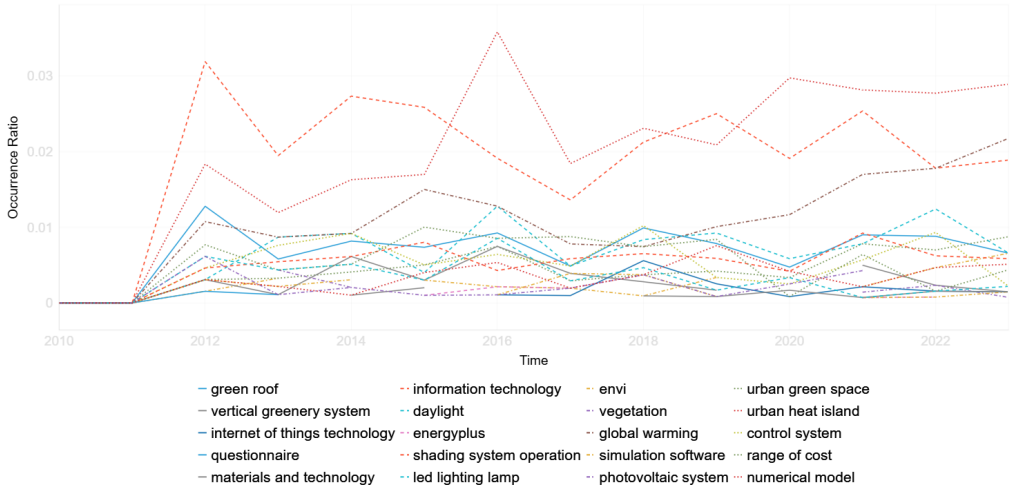


Figure 4.11: Trend analysis of the top 20 Methods for macro-cluster 0, with the Methods listed in the legend at the bottom of the plot. The y-axis measures the ratio of articles mentioning the Method to the overall number of articles in the cluster.

Chapter 5

Building Causality Graphs from Biomedical Text

5.1 Causality in Biomedical Text

Causal reasoning is a core aspect of human cognition, and extracting cause-effect relationships from text has long been recognized as a critical task in NLP, particularly for building causal networks that support prediction and decision-making [YHP22a].

Quantitative vs. Qualitative Approaches Approaches to causal analysis can be broadly classified into two categories: *qualitative* and *quantitative* methods. Qualitative approaches typically frame the problem as a task of Causal Relation Extraction (CRE), where a single-label, binary or multi-class causal relation between candidate entities is detected as claimed in an input text, optionally together with some additional variables affecting the causal effect. On the other hand, quantitative approaches aim to assess the strength of causal links and manage the uncertainty inherent in causal inference [CJSF24]. These latter approaches often build on theoretical models such as the Potential Outcome Framework [Rub74] and Graphical Models [PM18].

For example, in an observational study with K units indexed as $i = 1, ..., K$, with units partitioned by the values of a treatment assignment variable Z_i (where $Z_i = 1$ in the treatment group and $Z_i = 0$ for a control group), the Potential Outcome model aims to estimate the difference between these groups on an outcome variable Y_i, that is the Average Treatment Effect:

$$ATE = \mathbb{E}(Y_i(1) - Y_i(0)) \tag{5.1}$$

where $Y_i(1)$ is the Y value for the unit i if i is in the treatment group and $Y_i(0)$ is the Y value for the unit i if i is in the control group. In other words, as the difference between the potential outcomes for each single unit i, denoted as:

$$\Delta_i = Y_i(1) - Y_i(0) \tag{5.2}$$

can never be directly observed (one patient can either receive the treatment or not), one can resort to estimating the average difference for a population sample.

In a settings where one is not directly designing and managing a clinical trial but is relying instead on passively collected textual observational data such as Electronic Health Records (EHRs), generated through routine care and monitoring, the population sample will consist of text units T_i, where each T_i must be mapped to a vector of p pre-treatment covariates $X_i = (X_{i1}, X_{i2}, ..., X_{ip})$[1], by applying some text analysis tools [MKCM24].

In contrast, qualitative approaches bypass the uncertainty estimate and aim to directly identify potential causal relations between variables, such as entities or events in text. This can result in oversimplification of the causal effect estimate. For example, by default crucial intervening factors are not taken into account such as *confounders*[2]. While techniques have been recently proposed to derive more complex causal variables from text analysis (see [MKCM24]), this is outside the scope of this study, which entirely falls within the framework of a qualitative representation of causal relationships.

Annotation schemas for qualitative Causal Relation in NLP commonly draw from the cognitive theory of force dynamics [Wol17], which conceptualizes causality as dynamic force interactions between entities, aligning better with human intuition.

In the biomedical domain, the ability to accurately identify causal relationships between events or entities in text is critical to advance knowledge discovery [SP25, SK22]. Causal relationships underlie fundamental biomedical insights, forming clinical decision-making for example in treatment recommendations. However, manually analyzing vast amounts of biomedical literature and clinical texts is infeasible and has triggered the development of automated approaches for the extraction of causal relationships, enabling researchers and healthcare professionals to identify potential risk factors, understand disease progression, and assess treatment effectiveness more efficiently and accurately [AM21, VJR+23, PMP+23, SCR+25].

Text Corpora A range of diverse unstructured observational data, including EHRs, clinical notes, and online drug reviews, can serve as valuable sources for causal inference experiments. These data sources capture real-world patient experiences and medical events, offering opportunities to generate inexpensive causal effect estimates when Controlled Randomized Trials (CRT) are not practically or ethically feasible [SMV+21, MKCM24, FCea24].

At the same time, the potential of leveraging such unstructured textual data is hindered by the complexity and variability of biomedical texts, which often contain diverse linguistic patterns and domain-specific terminologies, as well as implicit causal statements that are not immediately obvious without context [Kil16, LC05, AM21]. The variety of text categories that can be used as observational data include:

1. *Electronic Health Records:* these include admission, progress and discharge notes,

[1] Such as patient age, gender, previous conditions, etc.

[2] A confounder is a variable that causally influences both the treatment and the outcome and can generate a spurious association between the two, if not accounted for.

summarizing hospitalizations, stays, diagnoses, interventions, prescribed medications, and follow-up plans. They are characterized by an highly technical language, specialized medical terminology, abbreviations, acronyms (even institution-specific), as well as telegraphic style including fragmented sentences and shorthand forms (e.g. *"pt c/o cp x3d"* for *"patient complains of chest pain for 3 days "*). Finally, they contain non-standardized Protected Health Information tokens.

2. *Drug Reviews*, as gathered from online health forums or social media platforms, are typically non-curated text featuring layperson vocabulary (e.g. *"heartburn"* for *"gastroesophageal reflux"*), non-standard spelling/grammar, commercial or slang drug names, subjective language, together with irrelevant anecdotal comments and vague temporal references (*"after a few weeks I felt dizzy"*).

The diversity in how causal information is expressed across these text collections makes it difficult to develop models that can consistently perform well across diverse datasets and scenarios [THH+23, YHP22b, FKM+22, LLB+24]. Additionally, the scarcity of comprehensive, labeled corpora specific to causal relationships limits the ability to train models effectively [AM21, XZLZ20, YPA+22, VJR+23].

The goal of this study is to evaluate the effectiveness of advanced open-source LLMs for CRE in the biomedical domain. While CRE can be approached using general RE techniques, identifying causal relationships presents distinct semantic challenges, as it often requires capturing implicit reasoning, temporal order, and contextual nuance beyond surface-level associations. Therefore, we systematically compare a range of LLM architectures and learning/inference strategies in the context of CRE. We benchmark these methods against two strong baselines using BERT and ClinicalBERT encoders, conducting experiments on the MIMICause dataset [KRH+22], a well-established resource for CRE in clinical text. To assess the cross-domain adaptability of these models, we also evaluate them on two additional datasets with partially different text characteristics and causal relationships classification schemas.

5.1.1 Task Definition

Regardless of the specific characteristics of the input text, our CRE task of identifying causal relations can be formulated as a single-label multi-class relation classification problem:

$$f : (X, e_1, e_2) \rightarrow Y \tag{5.3}$$

where $Y = \{y_1, ...y_n\}$ is the relation label, X is an input text sequence, e_1 and e_2 are non-overlapping, continuous token subsequences of X representing the entities between which the causal relation is to be identified, that is:

$$X = [x_1, x_2, ...x_{n-1}, x_n], \tag{5.4}$$

$$e_1 = X[i : j] \text{ with } i \leq j \text{ and } i, j \in [1..n], \tag{5.5}$$

$$e_2 = X[k:l] \text{ with } k \leq l \text{ and } k, l \in [1..n], \tag{5.6}$$

$$j < k \text{ or } l < i, \tag{5.7}$$

where the last statement asserts that the entities must not overlap and one must precede the other.

In the following, we present a comprehensive empirical analysis of how the various LLM-based learning strategies presented in Section 2.6 perform in the CRE task, testing across a range of SOTA, open-source decoder-only models, providing practical guidance and actionable insights for researchers and practitioners working with biomedical texts and applications. Moreover, we gauge on the portability of CRE-capable fine-tuned models across different types of textual collections within the biomedical domain.

5.2 Datasets

5.2.1 MIMICause

We benchmark various models and learning strategies to identify causal narratives within clinical notes using the MIMICause dataset [KRH+22], using its Hugging Face distribution[3]. MIMICause is derived from a collection of de-identified discharge summaries sourced from the MIMIC-III (Medical Information Mart for Intensive Care-III) clinical database [JPS+16][4], which are further annotated for the nine types of biomedical entities illustrated in Table 5.1 with corresponding examples.

The MIMICause annotation schema states that *"a causal relationship/association exists when one or more entities affect another set of entities"* [KRH+22]. Eight directed relation types between pairs e_1, e_2 of entities are defined, where the order of the entity tags determines the direction of causality: $Cause(e_1, e_2)$, $Cause(e_2, e_1)$, $Enable(e_1, e_2)$, $Enable(e_2, e_1)$, $Prevent(e_1, e_2)$, $Prevent(e_2, e_1)$, $Hinder(e_1, e_2)$, $Hinder(e_2, e_1)$. Additionally, an $Other$ relation class encompasses instances where either a non-causal interaction or no relationship at all exists between a given pair of biomedical entities. Consequently, MIMICause benchmark models CRE as a 9-class relation classification problem, i.e. $Y = \{0, ..., 8\}$.

The semantics of the four relation labels is the following:

- $Cause$: e_i is the direct and primary reason for the occurrence of e_j, like in the sample sentence: *"He was treated with <e1>ATRA</e1> and subsequently suffered from <e2>ATRA syndrome</e2> with an acute elevation in his WBC to >50"*, where the entity ATRA is the causative agent of ATRA syndrome.

- $Enable$: e_i facilitates or contributes to the occurrence of e_j in combination with other factors. This relation reflects uncertainty or partial involvement, like in the sample sentence: *"<e2>Rash</e2>- scattered papules on arm and swollen*

[3] `https://huggingface.co/datasets/pensieves/mimicause`

[4] Harvard's DBMI Data Portal: `https://portal.dbmi.hms.harvard.edu/projects/n2c2-nlp/`

Table 5.1: The entity types annotated in the *n2c2 2018 shared task*.

Entity Type	Examples
Drug	*morphine, ibuprofen*
Adverse Drug Event (ADE)	*nausea, seizures*
Reason	*vitamin K deficiency*
Dosage	*2 units, stress dose*
Strength	*10 mg, 60 mg/0.6 mL*
Form	*capsule, syringe, tablet*
Frequency	*daily, twice a day, Q4H*
Route	*transfusion, oral, gtt*
Duration	*for 10 days, chronic, 2 cycles*

eyelids. Unclear if from levaquin or <e1>morphine</e1>", where morphine is suspected to be one of multiple agents that may have led to the rash.

- $Prevent$: the emergence or application of e_i leads to the eradication, prevention or decrease of e_j, like in the sample sentence: *"<e2>Docusate Sodium</e2> 100mg Capsule Sig: One (1) Capsule PO BID (2 times a day) as needed for <e1>constipation</e1>"*, where the entity Docusate Sodium is intended to prevent the onset of constipation.

- $Hinder$: e_i weakens, delays, or limits the severity or frequency of e_j, in combination with other factors. For example, in the sample sentence, *"She was treated with home fentanyl 25mcg patch for pain control, home lidocaine patch with <e1>morphine</e1> for <e2>breakthrough pain</e2> Medications on Admission"* a combination of lidocaine and morphine is administered to reduce the severity or frequency of breakthrough pain.

For more details on the definitions of the causal relation schema, refer to the original paper [KRH+22].

Each row in MIMICause dataset comprises a text, a pair of entities explicitly mentioned in the text, and a causal relation label. Causal relations can link entity pairs within the same sentence or, in rare cases, spanning a few sentences in the input text. These relationships may be explicitly signaled by lexical causal connectives, such as "due to", or they may be implicit, requiring inference from the broader context.

The MIMICause dataset comprises 2,714 examples, with the train-dev-test split in the Hugging face distibution summarized in Table 5.2.

5.2.2 Adverse Drug Event Dataset

A second dataset we use to validate the generalization of our evaluation results is the Adverse Drug Event corpus (hereafter ADE), a popular benchmark of medical case reports, sourced from MEDLINE abstracts and annotated with mentions of drugs and

Table 5.2: Distribution of causal relation labels over train, eval and test splits of the MIMICause dataset. The second column contains the numerical equivalents of relation labels in the adopted Hugging Face distribution.)

MIMICause Relation	HF label	# Train Instances	# Eval Instances	# Test Instances
Cause(e1,e2)	0	254	64	36
Cause(e2,e1)	1	266	67	37
Enable(e1,e2)	2	126	31	17
Enable(e2,e1)	3	126	32	18
Prevent(e1,e2)	4	188	47	26
Prevent(e2,e1)	5	179	45	25
Hinder(e1,e2)	6	111	28	15
Hinder(e2,e1)	7	133	33	19
Other	8	570	142	79

adverse drug events (conditions) [GRR+12]. Namely, the distribution we used is the subset named *Ade_corpus_v2_drug_ade_relation* from the Hugging Face ADE-Corpus-V2 repository[5], which contains 6,821 case report sentences, each annotated with *drug* and *effect* entities, together with their character position within the text.

ADE assumes a binary classification schema of the relations between drugs and adverse events: the events are either effects caused by the drug, or they are not related, with no further sub-categorization of the relation. Moreover, ADE only contains positive instances, that is, examples of causal relation from the annotated *drug* to the annotated *effect*.

To align this binary classification setup with the causal relation classification schema of the fine-tuned models, trained on MIMICause, a relation label mapping is necessary. Specifically, MIMICause defines two types of directed positive causal links from E_1 to E_2: $Cause(E_1, E_2)$ and $Enable(E_1, E_2)$. The latter refers to situations in which a drug contributes to the occurrence of a condition in combination with other factors, according to MIMICause's definition of $Enable$. This is treated as a positive causal instance in the ADE dataset, as illustrated in the first two rows of Table 5.3, where bupivacaine and lidocaine are both contributing factors to methemoglobinemia. Conversely, $Prevent(E_1, E_2)$ and $Hinder(E_1, E_2)$ are interpreted as negative links, indicating the absence of a causal relationship between E_1 and E_2, and so are all inverse-direction relations (e.g., $Cause(E_2, E_1)$, $Enable(E_2, E_1)$, $Prevent(E_2, E_1)$), as well as the $Other$ class.

We will show in Section 5.6 how inference on ADE examples is performed using prompt-based conversion. Here, we just observe that inference output labels are mapped to 1, if they are equal to $Cause(e_1, e_2)$ or $Enable(e_1, e_2)$, and to 0 otherwise.

From the original ADE corpus, we generate two test sets. First, we use a 20% fixed random split of the full ADE[6]. Additionally, in order to evaluate across both positive and

[5] `https://huggingface.co/datasets/ade-benchmark-corpus/ade_corpus_v2`

[6] Notice that ADE does not provide an official train-test split. The index range of the collected split referencing the original ADE dataset is shared at the link `https://drive.google.com/file/d/1dIEhzoq_DN1_chKDxQbP2eUA5d0nN7hH/view?usp=drive_link` for full reproducibility.

Table 5.3: Positive examples of drug reaction case reports from the ADE dataset and two synthetic negative instances. The negative instances in rows four and five are generated by entity pair sampling from the sentence in row three.

text	drug	effect	label
Methemoglobinemia after axillary block with bupivacaine and additional injection of lidocaine in the operative field.	bupivacaine	Methemoglobinemia	1
Methemoglobinemia after axillary block with bupivacaine and additional injection of lidocaine in the operative field.	lidocaine	Methemoglobinemia	1
Cutaneous sarcoidosis during interferon alfa and ribavirin treatment of hepatitis C virus infection: two cases.	ribavirin	cutaneous sarcoidosis	1
Cutaneous sarcoidosis during interferon alfa and ribavirin treatment of hepatitis C virus infection: two cases.	ribavirin	infection	0
Cutaneous sarcoidosis during interferon alfa and ribavirin treatment of hepatitis C virus infection: two cases.	interferon	hepatitis	0

negative classes, we generate a synthetic set of negative examples using the following procedure. First, we collect the sets of all *Drug* and *Effect* entities in the entire ADE corpus. Subsequently, for each example in ADE, we generate all candidate combinations of all *Drug-Effect* entities matched within the example's text and compute the set difference with all occurrences of the same *Drug-Effect* combination for the same sentence in ADE (positive set). As ADE annotation can not be guaranteed to follow a strict Closed World Assumption, unannotated entity pairs in a sentence can not be assumed *a priori* to be true negatives. Therefore, we manually validated a small fraction of this initial negative sampling and removed error-prone entities[7]. We found out that upon removal of these spurious cases, ADE annotation is quite systematical and entities in a sentence not tagged in any causal link are actually not linked and therefore usable in negative samples. Consequently, we consolidated a set of 400 silver standard negative examples and finally created a balanced test sample of 800 instances. Rows 3 through 5 in Table 5.3 illustrate a positive example ($Cause$ or $Enable$ relationship) from ADE and two synthetic negative examples generated from the same sentence, respectively.

5.2.3 Drug Review Dataset

We created a drug review test set building upon the open-source *Drug Reviews (Druglib.com)* dataset, available within the UCI Machine Learning Repository[8].

The *Drug Reviews* dataset[9], introduced by Gräßer et al. [GKMZ18] to study sentiment analysis of drug experience, contains patient reviews on specific drugs along with

[7]Namely, entities that do not appear in positive combinations, but are sub-strings of entities that do.

[8]UCI Machine Learning Repository, available at `https://archive.ics.uci.edu/`

[9]*Drug Reviews*: https://archive.ics.uci.edu/dataset/461/drug+review+dataset+druglib+com

Table 5.4: Sample reviews with target entity metadata from the *Drug Reviews* dataset.

Drug	Condition	Review
ciprofloxacin	urinary tract infection	i had a urinary tract infection so bad that when i pee it smells but when i started taking ciprofloxacin it worked it's a good medicine for a urinary tract infections.
ziana	acne	when i first started using ziana, i only had acne in between my eyebrows, chin, and the nose area. my acne worsened while using it and then it got better. but after about 4 months of using it, it became ineffective. so i now have acne between my eyebrows, chin, cheeks, forehead, and the nose area. its great at first but after a while it made my face even worse than before i used the product.
nuvaring	birth control	i tried the nuvaring. this was my first form of any birth control. this was very easy to put inside and very easy to take out. i didn't feel the ring ever. i thought it was amazing until i started to get huge deep pimples. they were impossible to get rid of.

related conditions and was obtained by crawling online pharmaceutical review sites[10].

In the *Drug Reviews* dataset, the target Drug and Condition metadata entities are not always explicitly mentioned in the review text. For compliance with the instruction prompt setup of our benchmark evaluation in Section 5.3, we first filtered a subset of around 19,200 *Drug Reviews* instances where both Drug and Condition entities are matched within the text. Table 5.4 lists a few examples of reviews from this subset. Subsequently, for our evaluation we collected a random sample of 40 reviews for each possible relation: $Cause$, $Prevent$, $Hinder$, $Enable$ and $Other$, yielding an overall set of 200 relations to be validated.

Then, to evaluate the correctness and directionality of causal relations extracted from the drug reviews by the models, we conducted an annotation exercise involving three annotators per relation type[11]. The coders were instructed to simply read the drug review text and assess whether the output relationship was correct, with options to mark it as *True* if the relation *(E1 causal_rel E2)* was supported by the text, *False* if not, or *Swapped Entities* if the releation type was correct but direction was opposite (*E2 causal_rel E1*).

We calculated the average pair-wise Cohen κ inter-rater agreement [McH12] of all three raters, resulting in a value of 0.739, as well as the Fleiss κ_F agreement [FQ15], resulting in a value of 0.728, both indicating a substantial level of agreement among the annotators. For detailed figures see Table 5.8 in Section 5.6.

[10]The dataset is distributed under a CC BY 4.0 license, which allows using the data for research purposes as well as sharing and adapting it for any purpose.

[11]The annotators involved in the evaluation were specialists in the Digital Health field and entirely independent and external to this study.

5.3 Benchmarking Learning Methods

In this section, we first introduce the baseline approaches used for comparison with the learning methods under analysis. Next, we shortly describe the range of LLMs we experiment with. Finally, we present the results obtained for all model configurations and implemented methods.

5.3.1 Baselines

We compared our implemented learning methods with the two baseline architectures employed in [KRH+22], both leveraging BERT-based text encoders combined with fully connected Feed Forward Network (FFN) classifier layers. We will refer to them as BERT+Ent and Clinical-BERT+ENT. In both architectures, the sentence encoding vector is augmented via vector concatenation with the token-average context vectors of the two target entities before being passed through the FFN and softmax classification layers. Among these baselines, the architecture incorporating the domain-specific Clinical-BERT encoder, denoted as `Clinical-BERT+Ent` in Table 5.5, yields the best performance[12].

5.3.2 Large Language Models

We tested and evaluated five main families of state-of-the-art, open-source LLMs with decoder-only architectures, namely Mistral, Llama (ver 2 and ver 3), Gemma, and DeepSeek. In order to provide meaningful comparisons between models and to operate within the constraints of limited compute resources, we opted for using small-to-mid-range models across the various architectures. We also applied quantization throughout our experiments, as described in Table 4 in Appendix D. All the models tested in this study are released as open-source (with varying usage licenses) and accessible through Hugging Face via the Transformer library, hereby making our study fully reproducible from a user perspective.

Mistral: We used Mistral-7B-v0.1[13], a pre-trained generative text model with 7 billion parameters, released under Apache 2.0 license. By leveraging Grouped-Query Attention (GQA) [ALTdJ+23] and Sliding Window Attention mechanisms (SWA) [BPC20], Mistral-7B-v0.1 provides increased inference speed and reduced memory requirements for the decoding of longer token sequences [JSM+23b], while outperforming the Llama-2 13B model with almost half its parameters. As a base model with no instruct-tuning, we used Mistral-7B-v0.1 for fine-tuning experiments only.

[12]Note that the Clinical-BERT performance may be slightly overestimated as its embeddings were fine-tuned on text collections including the MIMIC dataset, albeit without explicit causal relation annotations.

[13]`https://huggingface.co/mistralai/Mistral-7B-v0.1`

Moreover, we used MistralOrca[14] for all of our in-context learning evaluations. MistralOrca is a 7 billion parameters model based on Mistral 7B architecture and further instruction-tuned by OpenOrca on a reproduction of the Orca dataset [MMJ+23], which focuses on imitating step-by-step reasoning and explanations from larger teacher models [LGW+23]. This makes it particularly suitable for tasks requiring explanation, reasoning, and multi-step problem-solving, such as the Chain-of-Thought and Tree-of-Thought methods.

Llama 2: The Meta Llama 2 family is a collection of pre-trained and fine-tuned generative text models, ranging in scale from 7 billion to 70 billion parameters, all with a 4k tokens context length [TMS+23a]. We experimented with instruction fine-tuning on the smallest-sized release of the Llama 2 foundation models, Llama-2-7b[15], which was pre-trained on 2 trillion tokens of data from publicly available sources.

Llama 3: The Meta Llama 3.1 family encompasses a collection of multilingual, pre-trained, and instruction-tuned decoder-only transformer models in 8B, 70B and 405B sizes, all featuring support for extended context lengths (128k tokens) and Grouped-Query Attention.

The Llama-3.1-8B-Instruct model we adopted in our experiments[16] is optimized for instruction-following tasks and has undergone Supervised Fine-Tuning (SFT) and Reinforcement Learning with Human Feedback (RLHF) [CLB+17] to align with human preferences for helpfulness and safety.

JSL-MedLlama-3-8B-v2.0 (hereafter referred to as *MedLlama*[17]), built upon the Llama 3 8B architecture, is an advanced model developed by the John Snow Labs specifically tailored for medical and healthcare applications, having undergone fine-tuning on extensive medical literature and datasets. By testing this model, we wanted to gauge the impact of domain-specific fine-tuning data on the CRE performance.

Gemma: Gemma is a family of lightweight SOTA open models from Google, well-suited for a variety of text generation tasks, including question-answering, summarization, and reasoning [Tea24]. The gemma-2-9b model we tested[18] was trained on an 8 trillion tokens dataset including Web Documents, code, and mathematical texts.

DeepSeek-Qwen-Distill: The DeepSeek-R1-Distill-Qwen-32B (referred to as *DeepSeek-Qwen-Distill* in the results Table 5.5), is an open-source, dense LLM based on the Qwen2.5-32B architecture and trained by Reinforcement Learning (RL) using the output reasoning data of the DeepSeek-R1 model [DAGY+25]. DeepSeek-R1 is

[14] `Open-Orca/Mistral-7B-OpenOrca`
[15] `meta-llama/Llama-2-7b-hf`
[16] `https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct`
[17] `https://huggingface.co/johnsnowlabs/JSL-MedLlama-3-8B-v2.0`
[18] `https://huggingface.co/google/gemma-2-9b`

a Mixture-of-Experts architecture (totaling 671B parameters) that recently gained popularity for reaching outstanding reasoning capabilities using cost-effective RL. By directly fine-tuning the smaller Qwen2.5-32B model on 800k reasoning examples curated with DeepSeek-R1, the reasoning patterns of the larger model have been successfully distilled into the smaller one, achieving new state-of-the-art results for dense models in reasoning tasks like MATH-500 and LiveCodeBench [DAGY+25].

By testing on DeepSeek-Qwen-Distill we aim to assess how powerful, general reasoning capabilities might affect LLM's performance on domain-specific CRE tasks.

5.4 Inference and Learning Strategies

To evaluate the ability of the LLMs to detect causal relationships, we designed a set of experiments based on five distinct learning paradigms, namely: i) instruction prompting, ii) in-context learning, iii) prompt chaining, iv) chain-of-thought, and v) instruction fine-tuning.

Prompt-based approaches have become a de facto standard for evaluating pre-trained LLMs across NLP tasks, as are known to effectively leverage the implicit knowledge encoded in LLMs. Moreover, prompt engineering supports high flexibility and reproducibility and allows rapid exploration of task-specific capabilities of LLMs, particularly for structured tasks like CRE [LYF+23b]. On the other hand, we test on instruction fine-tuning to assess the potential performance gains when adapting model parameters on task-specific data.

A consistent set of inference parameters are used across all inference strategies and models, with the exception of the *max_new_tokens* which is adapted to each prompting technique to accommodate intermediate reasoning steps. Inference parameters settings are listed in Table 4 in Appendix D. Essentially, we use greedy decoding to enforce deterministic token generation.

All the scripts for running inference, training and evaluating the LLMs considered in this benchmarking study are made publicly available in a code repository, described in the Appendix D.

Instruction Prompting We first employed an instruction prompting method that builds upon zero-shot learning for CRE [MCHS23]. Our approach aims to harness the background knowledge of the LLM by explicitly defining the semantics of the relation labels in the task. Specifically, we structured the prompt into four key components:

- *Task Instruction*: This section outlines the task to be solved.
- *Semantic Definition of Labels*: Here, the numerical labels from MIMICause are mapped to natural language definitions to clarify their meanings.
- *Output Formatting*: This part specifies the expected format of the output.

- *Input Data*: The input sentence, along with the target entities E1 and E2, whose relation is to be classified, is provided here.

This structured approach ensures that the LLM can effectively interpret and perform the RE task by leveraging both explicit guidance and its inherent knowledge.

Figure 5.1 provides a detailed illustration of the full prompt used in this method, along with a sample output generated by the model. To further encourage the model to contextualize its reasoning, we conclude the prompt with the instruction, *"Please explain your response"*. This approach aims to elicit the model's understanding of the input context and its decision-making process.

In-context Learning We implemented the iCL paradigm by integrating the same task instructions used in the instruction prompting and additionally generating in-context examples by randomly sampling K instances from the MIMICause training split[19]. For a sample illustration of a few-shot prompt, please refer to the Appendix D.

Prompt Chaining We implemented two variants of prompt chaining. The first adapts the SumAsk prompting technique proposed in [LWK23], decomposing the RE task into a 3-step prompt pipeline. This pipeline guides the LLM to:

1. Generate a summary of the relationship(s) between the target entities;
2. Reformulate each relation label into a Yes/No question about the target entities;
3. Use the summary from step 1 to answer the questions from step 2, returning the label corresponding to the most probable "Yes" answer.

In the second prompt chaining variant, we used a simpler 2-step prompt chaining approach. First, we ask the model to identify the relationship between the target entities. Then, we pass the model's output as an input variable to a second prompt, which asks the LLM to return the label that best describes the relationship.

Links to the full illustrations of both variants are provided in Appendix D.

Chain-of-Thought For zero-shot CoT, we append the usual *"Let's think step-by-step"* instruction at the end of a standard prompt.

For few-shot CoT, we tested two variants:

- *One-shot CoT*: A single reasoning example is randomly sampled from the 9 relation types.
- *9-shot CoT*: One reasoning example is provided for each of the 9 relation types.

[19]To ensure robust performance and avoid overfitting to specific training examples, the sampling of in-context examples is performed independently for each test instance inference, using a fixed random seed (42) to ensure reproducibility.

Zero-Shot Prompt

Task

```
Given a text enclosed in triple quotes and a pair of entities E1
and E2, classify the relation holding between E1 and E2.
```

Labels

```
The relations are identified with 9 labels from 0 to 8. The
meaning of the labels is the following:
0 means that E1 causes E2
1 means that E2 causes E1
2 means that E1 enables E2
3 means that E2 enables E1
4 means that E1 prevents E2
5 means that E2 prevents E1
6 means that E1 hinders E2
7 means that E2 hinders E1
8 means that E1 and E2 are in a relation different than any of
the previous ones.
```

Format

```
Given X the label that you predicted, for the output use the
format LABEL: X
```

Data

```
Text: ```mg po BID prn agitation in an effort to avoid sedation
associated with benzos.'''
E1: ```benzos''', E2: ```sedation''
Please explain your response.
```

Model Sample Output

```
LABEL: 4 The text suggests that the benzos (a type of medication) cause
or lead to sedation, which means that E1 (benzos) prevents E2 (sedation).
Therefore, the relation is labeled as 4.
```

Figure 5.1: Sample zero-shot prompt with explicit semantic definition of the MIMICause relation labels (top) and a LLM-generated response (bottom).

To mitigate example bias, we manually curated a pool of 3 reasoning examples per relation category based on the following criteria: i) each example contains a clearly annotated causal relation between entities; ii) sentence structures vary in complexity and length to simulate realistic clinical data diversity; and iii) the reasoning path between entities is interpretable and aligned with the expected relation label. All examples were independently reviewed by two clinical NLP experts to ensure quality and label con-

sistency. During inference, examples were randomly sampled from this pool to ensure diversity.

For the 9-shot CoT, we randomized the order of the reasoning examples, as their presentation order is known to influence prompt performance [LBM+22].

A sample one-shot CoT prompt illustration is pointed to in Appendix D.

Instruction Fine-Tuning For instruction fine-tuning, we first transformed the MIMICause training split into instruction prompts, as described in Figure 5.2.

```
Rash:<e2>Diffuse livido reticularis appearing rash</e2>.She
was seen by dermatologyand it was felt to be a reaction to her
<e1>vancomycin</e1>, and she was switched to daptomycin and started
on topical triamcinolone and hydrocortisone and the rash improved.
E1: vancomycin
E2: Diffuse livido reticularis appearing rash
Label: 0
```

```
Given a text enclosed in triple quotes and a pair of entities E1 and
E2, classify the relation holding between E1 and E2.
The relations are identified with 9 labels from 0 to 8. The meaning
of the labels is the following:
0 means that E1 causes E2
1 means that E2 causes E1
2 means that E1 enables E2
3 means that E2 enables E1
4 means that E1 prevents E2
5 means that E2 prevents E1
6 means that E1 hinders E2
7 means that E2 hinders E1
8 means that E1 and E2 are in none of the relations above.
For the output use the format LABEL: X
Text:```Rash:Diffuse livido reticularis appearing rash.She was
seen by dermatologyand it was felt to be a reaction to her
vancomycin, and she was switched to daptomycin and started on
topical triamcinolone and hydrocortisone and the rash improved.'''
E1:```vancomycin''', E2:```Diffuse livido reticularis appearing
rash'''
LABEL: 0
```

Figure 5.2: Transformation of a MIMICause datapoint into an instruction prompt for model fine-tuning.

Each generative model was then fine-tuned on this instruction dataset using the

`trainer` class from Hugging Face[20] [WDS+20]. We maintained a consistent set of hyperparameters across all experiments, with the primary settings listed in Table 5 in Appendix D. Since our primary focus is to evaluate CRE performance across various learning methods and pre-trained models, we have not performed extensive hyperparameter optimization.

Given the computational limitations of fully fine-tuning large generative models, we employed the Low-Rank Adaptation (LoRA) technique for Parameter-Efficient Fine-Tuning [YYK+23]. Using Hugging Face's `PeftModel` class, we wrapped the base LLMs into trainable PEFT models. The same LoRA hyperparameter configuration was applied across all experiments and reported in Table 5 in Appendix D. This resulted in a trainable parameter ratio of 0.04 relative to the original model size.

5.5 Results

Table 5.5 presents the F1 scores on the test split of the MIMICause dataset across all model configurations, in-context example settings, and learning strategies. Given the relatively poor performance of most techniques compared to instruction fine-tuning, we report F1 scores only for the following representative models:

1. *MistralOrca*, the best-performing general-purpose model not specialized in domain-specific text;
2. *DeepSeek-Qwen-Distill*, a next-generation model that has established a new state of the art in LLM reasoning capabilities;
3. *MedLlama*, evaluated using two strategies (9-shot CoT and Prompt Chaining).

Overall, fine-tuned models largely outperform encoder-only baselines, achieving substantially higher F1 scores even when starting from less domain-specific pretraining. This highlights their comparatively higher robustness and adaptability to the specialized clinical language.

Moreover, fine-tuned models significantly outperform all other LLM methods. Notably, even the lowest-performing fine-tuned model (*Llama-2-7B*) achieves a macro F1 score that is over 0.25 higher than the best-performing in-context learning configuration. This result is expected, given the domain-specific nature of the causal relation labels. Supporting this, we observed that many responses generated via CoT reasoning attempted to elaborate on the semantics of the relation but still produced incorrect labels, like for the entities *E1:4 days* and *E2:antibiotics* in the text:

> *"Following 4 days of IV antibiotics, the patient was narrowed to PO amoxicillin for enterococcus plus cipro for serratia."*

where one model response was:

[20] `https://huggingface.co/docs/transformers/en/main_classes/trainer`

Table 5.5: F1 performance values on the test split of the MIMICause dataset of various combinations of models, parameter sizes, learning examples, and learning methods, against the two encoder model Baselines (BL). The reported Macro F1 values are averaged over the nine MIMICause relation categories. In the Method column, FT denotes Fine-Tuning, SumAsk and 2-Chain correspond to the two implementations of Prompt Chaining, iCL represents In-context Learning and CoT refers to Chain-of-Thought. The best-performing configurations within each general learning category are highlighted in bold.

	Model	**Size**	**Method**	**N examples**	**Micro F1**	**Macro F1**
BL	BERT+Ent	110M	FT	1953	-	0.54
	Clinical-BERT+Ent	110M	FT	1953	-	0.56
zero-shot	MistralOrca	7B	InstPrompt	-	0.096	0.064
	DeepSeek-Qwen-Distill	32B	InstPrompt	-	0.267	0.172
	MistralOrca	7B	SumAsk	-	0.449	0.343
	DeepSeek-Qwen-Distill	32B	SumAsk	-	0.338	0.163
	MistralOrca	7B	2-Chain	-	0.591	0.348
	DeepSeek-Qwen-Distill	32B	2-Chain	-	**0.60**	**0.450**
	MedLlama	8B	2-Chain	-	0.304	0.210
	MistralOrca	7B	0-shot CoT	-	0.342	0.273
	DeepSeek-Qwen-Distill	32B	0-shot CoT	-	0.541	0.387
few-shot	MistralOrca	7B	iCL	1	0.055	0.033
	MistralOrca	7B	iCL	3	0.074	0.048
	MistralOrca	7B	iCL	9	0.089	0.073
	MistralOrca	7B	iCL	27	0.237	0.177
	MistralOrca	7B	9-shot CoT	9	0.405	0.278
	DeepSeek-Qwen-Distill	32B	9-shot CoT	9	**0.557**	**0.489**
	MedLlama	8B	9-shot CoT	9	0.230	0.203
fine-tuning	Mistral-7B-v0.1	7B	FT	1953	0.817	0.780
	MistralOrca	7B	FT	1953	0.843	0.817
	Llama-2-7b	7B	FT	1953	0.788	0.747
	Llama-3.1-8B-Instruct	8B	FT	1953	0.788	0.751
	MedLlama	8B	FT	1953	**0.847**	**0.829**
	gemma-2-9b	9B	FT	1953	0.825	0.810

> *"The relation holding between 4 days and antibiotics is that 4 days of antibiotics were given to the patient. The closest relation that best describes it is: 4 days enables antibiotics. The label associated to this closest relation is 2. The answer is: LABEL 2."*

Here clearly the correct label is 8, as there is no relation between the entities *E1:4*

days and *E2:antibiotics*.

As expected, zero-shot methods generally perform poorly. Among them, simple Instruction Prompting yields the lowest micro F1 score (0.096). However, more advanced prompting techniques such as Prompt Chaining and CoT provide substantial improvement.

Specifically, the 2-Chain approach achieves the highest performance among zero-shot techniques ($F1 = 0.45$), outperforming both CoT (0.387) and SumAsk (0.343). This suggests that decomposing the classification task into multiple reasoning steps helps activate latent knowledge from pretraining, leading to more accurate semantic labeling. These findings align with observations from [JGMW+22] and [LWK23].

While CoT shows a similar benefit, its effectiveness is specifically pronounced in models capable of longer reasoning chains and self-corrections, such as *DeepSeek-Qwen-Distill*.

For instance, given the entities *E1: Oxycodone-Acetaminophen* and *E2: pain* in the following input:

> *"Oxycodone-Acetaminophen 5-325mg Tablet Sig: One(1) Tablet PO Q6H (every 6 hours) as needed for pain."*

a zero-shot CoT prompt to *MistralOrca* led to this reasoning:

> *"The relation holding between Oxycodone-Acetaminophen and pain is that Oxycodone-Acetaminophen is a medication used to treat pain. The closest relation that best describes it is: Oxycodone-Acetaminophen prevents pain. The label associated with this closest relation is 4. The answer is: LABEL 4."*

However, in other instances, only *DeepSeek-Qwen-Distill* was able to infer the correct label through complex reasoning. This is illustrated in Figure 7 in Appendix D.

Other observed patterns of erroneous responses include cases where the reasoning sequence is sound, but eventually the mapping to the target label fails, or cases where a label for the inverse of the target relation is output. Examples 1. and 2. below, with their corresponding entities and CoT responses from the *Mistral-7B-OpenOrca* model, illustrate the first and second pattern, respectively:

1. E1:*Pneumonia*, E2:*levofloxacin*

   Text: *"Multifocal Pneumonia : She was admitted with multifocal pneumonia. She was started on levofloxacin and vancomycin."*

   CoT: *"The text says that the patient was admitted with multifocal pneumonia and was started on levofloxacin and vancomycin. This means that levofloxacin is a treatment for pneumonia.So, the correct label is:LABEL: 0"*

2. E1:*post-ATN diuresis*, E2:*fluids*

   Text: "*Patient was given continuous IV fluids and developed post-ATN diuresis complicated by one episode of hypernatremia.*"

   CoT: "*The text says that the patient was given continuous IV fluids and developed post-ATN diuresis.So, the relation between post-ATN diuresis and fluids is that fluids cause post-ATN diuresis.Therefore, the correct label is 0.LABEL: 0*"

Furthermore, we find that manually crafting artificial reasoning chains for few-shot CoT prompting provides performance gains primarily for highly capable models such as *DeepSeek-Qwen-Distill*. For other models, 9-shot CoT does not yield meaningful improvements over the zero-shot setting.

Few-shot in-context learning shows modest gains as the number of examples increases, with the overall performance remaining low. This indicates that LLMs struggle to generalize causal relations from limited in-context examples in the absence of task-specific parameter tuning.

In contrast, fine-tuning allows models to internalize the causal structure of the dataset, moving beyond reliance on general pretraining knowledge. Among all fine-tuned models, *MedLlama*, extensively pre-trained on medical literature and datasets, achieves the best performance. This result suggests that pretraining on domain-specific content helps build stronger representations of clinical language, including abbreviations and acronyms.

Nevertheless, pretraining alone does not suffice for effective CRE. As shown in Table 5.5, *MedLlama*'s performance in the 2-Chain and 9-shot CoT configurations remains limited.

To assess the statistical significance of these findings, we run a paired random permutation test [Pit37] comparing each fine-tune model with the best non-tuned options, namely DeepSeek-R1-Distil-Qwen-32 B with chain of thoughts (DeepSeek-R1-Distill-Qwen-32B 9-shot CoT) and zero-shot (DeepSeek-R1-Distill-Qwen-32B-chain)[21]. We found that all fine-tuned models produce significantly different predictions from those of the two DeepSeek models ($p < .01$ for all pairs) while differences between individual fine-tuning models were not statistically significant. Moreover, the two DeepSeek-based methods are not significantly different from each other ($p = 0.3177$), confirming that including in-context examples does not positively impact the model's performance in this specific task.

5.5.1 Relation Analysis

Examining more in-depth the classes of extracted MIMICause relations, Figure 8 in Appendix D presents the confusion matrix across the nine classes for the best-performing model (the fine-tuned *MedLlama*). The matrix reveals that most errors stem from semantically related or directionally reversed relations. In particular, $Prevent(e2, e1)$

[21]We use Holm method [Hol79] to correct p values for multiple comparisons.

and $Hinder(e1, e2)$ are occasionally interchanged, reflecting their semantic proximity as inhibitory causal types. Directional pairs (e.g., $Cause(e1, e2)$ vs. $Cause(e2, e1)$) also show moderate confusion, suggesting that interpreting syntactic cues of argument order remains partially challenging for decoder-only LLMs. Finally, the broad $Other$ category attracts a small fraction of ambiguous instances across all labels.

The box plots in Figure 5.3 summarize the distribution of F1 scores (Y-axis) against class frequencies (X-axis), grouped by all the fine-tuned models[22].

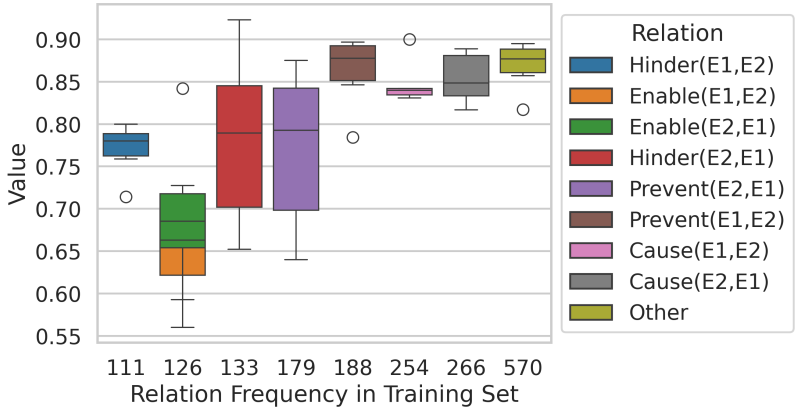


Figure 5.3: Training instance frequency analysis. Distributions of the F1 scores (Y-axis), collapsed by all fine-tuned models, on each of the nine classes in MIMICause, as a function of the each class' occurrence in the train split of the MIMICause dataset (X-axis).

Overall, the plot shows that performance tends to be either low or highly variable before reaching a threshold of approximately 188 instances, possibly reflecting a more significant influence of differing pre-training data on the fine-tuning phase across the different models. Conversely, the low variance and high average performance observed for the classes $Prevent(E_1, E_2)$, $Cause(E_1, E_2)$, $Cause(E_2, E_1)$, and $Other$ on the right-side of the plot (higher frequency relation classes) are coherent with some previous research showing how even simpler models like LSTMs could learn to generalize a combination of lexemes (in this case, the (`$E_1$, relation, $E_2$`) triple) after observing it for a specif amount of times in the training data [VSR+22].

In the Figure 5.4 below, each diagram disaggregates the data from Figure 5.3 by individual model, with each point in the curve representing the F1 score obtained by a model for different classes, here indicated solely by frequency[23].

The figure helps to better understand how each model exhibits a distinct response pattern, with model performance stabilizing after exposure to a varying threshold cen-

[22] Two classes, `Enable(`E_1`, `E_1`)` and `Enable(E2, E1)`, have the same number of instances (126), which explains why there are two plots corresponding to the 126 instances count.

[23] Note that the error bar refers to the `Enable` relation, which is the only relation in MIMICause for which |`relation(`E_1`, `E_2`)`| == |`relation(`E_2`, `E_1`)`| (see Figure 5.3).

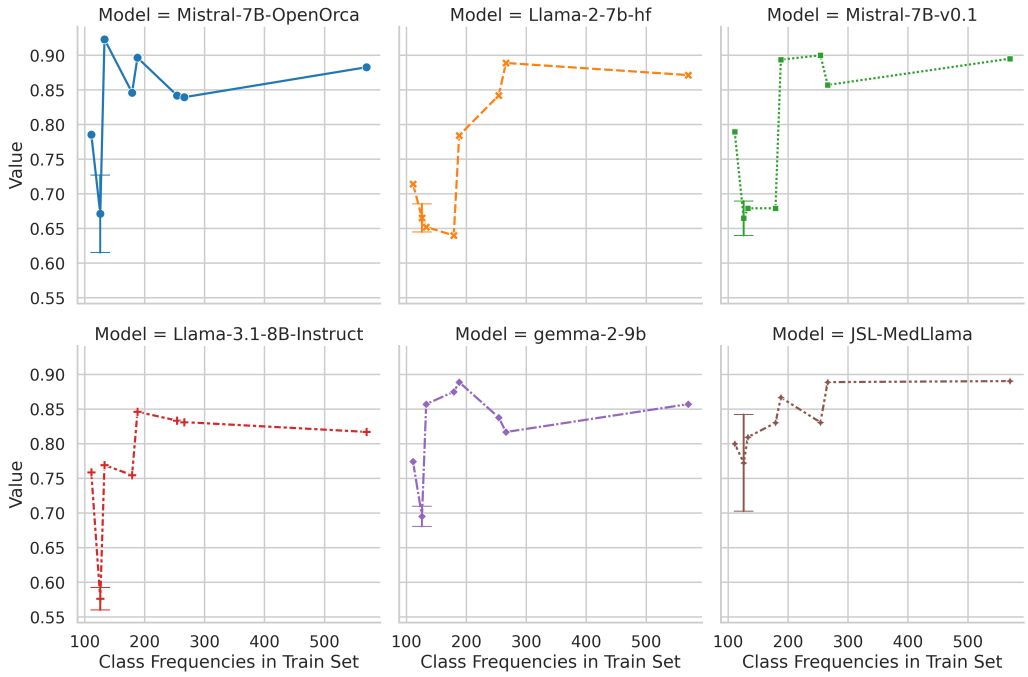


Figure 5.4: Training instance frequency analysis per model: F1 scores (Y-axis) obtained by the different fine-tuned models on each of the nine classes in MIMICause, as a function of the each class' occurrence in the train split of the MIMICause dataset (X-axis).

tered between 200-300 instances of training data, after which the task becomes notably easier.

Among the evaluated models, MedLlama demonstrates greater stability and consistency across all classes of MIMICause, with performance levels centered around 0.8 for harder relations such as $Enable(E_1, E_2)$ or $Hinder(E_1, E_2)$, and around 0.9 for relations $Enable(E_1, E_2)$, $Prevent(E_1, E_2)$ and $Other$.

Overall, our analysis suggests that model performance is not directly correlated with the frequency of a specific label in the training set, but rather by the interaction among components within each $(E_1,$ `relation`, $E_2)$ triple, as highlighted by the fact that classes sharing the same relation (e.g., $Prevent(E_1, E_2)$ and $Prevent(E_2, E_1)$) exhibit remarkably different results. Additionally, the results do not indicate pure memorization, as performance does not follow a binary 0–100% pattern but instead fluctuates within the mid-to-high F1 score range.

5.5.2 Prompt Design Sensitivity

Additionally, we wanted to explore how the performance within each learning strategy is robust across variations on prompt design choices. Therefore, we conducted a sensitivity analysis applied to the instruction prompting and few-shot methods defined in Section 5.4. For each such method, a set of significant prompt design dimensions is

identified and, for each dimension, we apply a range of four diverse variations from the base instruction prompt illustrated in Figure 5.1. For each prompt variation, the MistralOrca model is evaluated and micro F1 scores are computed over the MIMICause test split. We also report standard deviation over the range of F1 scores.

While instruction prompting turned out to be the least effective method in Section 5.5, testing on it enables us to gauge on the model robustness to prompt design variations independently from confounding factors characterizing the more complex reasoning chain methods. Moreover, the base instruction prompt is the core component of the iCL, CoT and instruction fine-tuning methods implemented in Section 5.4, so that the present robustness analysis arguably projects to those approaches as well[24].

Table 5.6: Micro F1 performance scores on the test split of the MIMICause dataset of five variants of the base instruction prompt illustrated in Figure 5.1 in Appendix D, for the MistralOrca model. Each row contains prompt design variants with respect to the target dimension indicated in the left column headings. The last column reports standard deviation over those variations.

	Prompt Dimension	v0	v1	v2	v3	v4	Std Dev (%)
instr. prompt	**Persona**	0.096	0.100	0.189	0.141	0.141	± 28.35
	Phrasing	0.096	0.059	0.063	0.052	0.107	± 32.44
	Definitions	0.096	0.115	0.0929	0.171	0.237	± 43.18
	Format	0.096	0.130	0.126	0.167	0.118	± 20.20
	Explain	0.096	0.078	0.163	0.115	0.089	± 30.92
	Ordering	0.096	0.223	0.282	0.301	0.278	± 35.37
few-shot	**Selection**	0.193	0.122	0.178	0.230	0.167	± 22.09
	Ordering	0.078	0.115	0.089	0.141	0.223	± 44.75

Table 5.6 reports the performance scores, where v0 represents the base prompt (no variation), while v1 through v4 denote variations relative to a prompt design dimension. We shortly introduce in the following the meaning and rationale of such dimensions along with some examples. The complete set of prompt variations is made available in the folder *prompt_analysis* in the code repository (see Appendix D).

1. Persona setting (referred to as Persona in Table 5.6): this is the process of defining a target role of the LLM in a prompt, typically via a system message or a sentence prefixed to the prompt instruction. v0 base prompt does not include a Persona, while we explore a few variants of domain-specific Persona setting clauses, identifying the model either as human-like assistant (e.g. *"You are a clinical NLP expert specializing in medical text annotation"* or as a system (e.g. *"You are a knowledgeable clinical NLP model ... "*).

[24] Among the two models evaluated with instruction prompting we opted for testing on MistralOrca here given our computational infrastructure constraints.

2. Instruction phrasing (Phrasing): We vary the tone and style of the task instructions, from more formal with explicit input markup specification to more direct and informal.

3. Label definition phrasing (Definitions): we compare the original label definitions and numerical label mapping in the base instruction prompt with alternative formattings and orderings, as well as alternative wording using lexical synonyms. v4 prompt implements the official, verbose definition of the MIMICause relation labels from the original paper [KRH+22].

4. Output formatting instruction (Format): we vary the instruction for output formatting, testing for json and xml variants, alternating instruction styles as well.

5. Explanation requirement (Explain): we test with dropping the closing request for output label explanation ("Please explain your response."), as well as with demanding different levels of detail.

6. Instruction ordering (Ordering): here, we test alternative ordering of the component parts of the base prompt, namely Task Instruction, Label Definition, Formatting Instruction and Input Data.

For the few-shot method, we measure robustness with respect to example selection strategies and to the ordering of the selected examples. Throughout these evaluations, we use an optimized prompt deploying the best scoring configurations from the previous analysis and a 9-shot settings, with exactly one example per label (see the exact prompt in the *prompt_analysis* folder in the code repository). This way, we can test whether performance gain from prompt design optimization propagates also through the few-shot method. The nine examples are selected using five strategies:

1. Randomized (v0): one randomized example per label type, sampled independently for each inference instance;

2. Prototype (v1): selecting, for each relation type, the medoid example among all example embedding vectors for that relation[25];

3. Max_Diversity (v2): selecting the nine label representative examples that maximize the sum of pairwise embedding vector distance scores, therefore sampling for variety in the train set;

4. Min_Diversity (v3): selecting the nine label representative examples that minimize the sum of pairwise embedding vector distance scores;

5. Input_Similarity (v4): For each inference instance, it selects the nine label instances with highest embedding similarity to it.

[25]Using *all-MiniLM-L6-v2* Sentence Transformer over a concatenation of the example's text and entity strings.

Finally, in order to test for example ordering, we fix the set of 9 examples from the Min_Diversity strategy and re-shuffle the order for each variation v0 through v5.

Overall, we observe that instruction prompt exhibits a relatively high variance across prompt design settings, with standard deviation ranging from 20% to 43%. This shows that prompt optimization can significantly boost the performance of this inference method. Nonetheless, the range of performance scores remains below the level of the reasoning-based zero-shot methods evaluated in Section 5.5 for the same model.

Label definition and prompt structure ordering are the most effective dimensions of prompt optimization. We find that enriching the definitions of the MIMICause causal relations with synonym and paraphrase expressions elicits the model's pre-training knowledge enhancing its interpretation of the input text. Interestingly, we also observe that moving the input data increasingly closer to the instruction clause within the prompt improves the performance, with the best result obtained when data are prefixed to the prompt (v4). Unsurprisingly, removing the closing request for explanation of the predicted label deteriorates the performance, while strengthening the requests elicits CoT-type output, boosting the results. Finally, while the impact of Persona setting has been observed to be inconsistent for factual Question Answering tasks in an extensive study by [ZPL+24], we find that it systematically improves F1 scores on our CRE task.

Few-shot method is relatively more robust to example selection strategies, with the only strategy outperforming random selection being Min_Diversity. We observe that in-context examples do not enhance the model performance even in combination with an optimized prompt, consistently with what found in Section 5.5.

5.5.3 Training Hyperparameters

While fine-tuning proved to be the most effective method for our CRE task, in order to test how robust the method is to variations across hyperparameter settings in the training process, we performed a sensitivity analysis applied to *MedLlama*, the best performing architecture, focusing on three main parameters controlling model instantiation and the training process, namely: LoRA rank, training learning rate and training batch size, indicated respectively as $rank$, lr and bs in Figure 5.5. The parameters ranged over the values: $rank = [4, 8, 16]$, $lr = [1e-5, 2e-5, 5e-5]$, $bs = [4, 8]$.

To limit computational resource use, we tuned on a subset of 1,000 instances (around 50%) of the MIMICause train split. We applied a step-based evaluation strategy with an early stopping approach and patience parameter equal to 3 and a *max_steps* cap of 700.

Figure 5.5 shows the evolution of the loss values on MIMICause validation set (489 instances) of *MedLlama* for 12 intervals of 50 steps (starting from 50 and ending with 700 steps), during training with a total of 18 parameter configurations listed on the top-right corner. We found that, while starting from three clusters of very distant performance levels after the first 50 steps, all model configurations eventually converge towards an interval of < 5 points of validation loss, with values ranging from 0.171 to 0.220. We also evaluated the micro F1 scores on the MIMICause test split of the best and

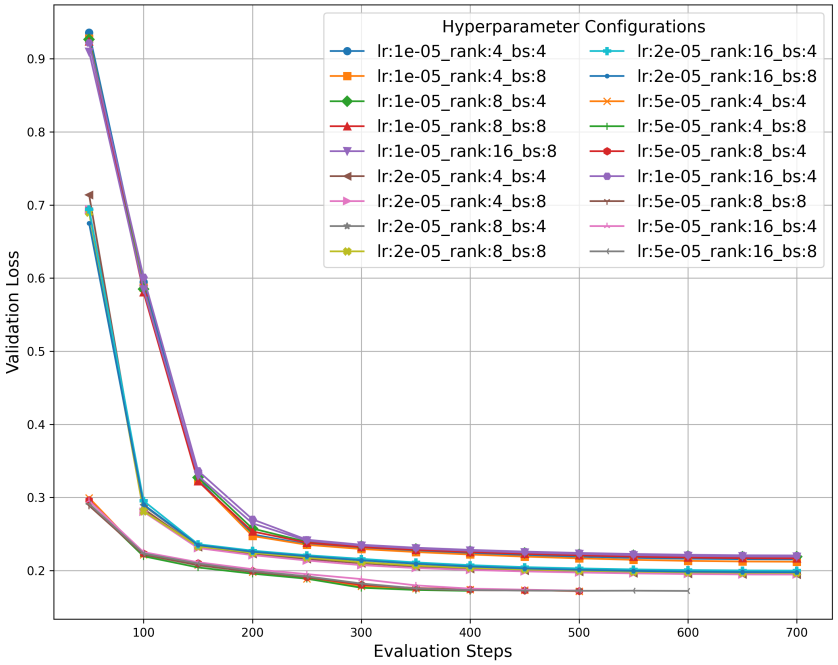


Figure 5.5: Sensitivity analysis of the best performing fine-tuned model, *MedLlama*, to variation over Lora rank, learning rate, and training batch size hyperparameters (respectively $rank$, lr, and bs in the Figure). The plots show the validation loss values for 12 intervals of 50 evaluation steps, for a total of 18 model configurations listed.

worst performing configurations identified in the present hyperparameter search, finding a performance delta of around 0.04 (0.698 vs. 0.661). Although not fully comprehensive, these findings suggest an overall robustness of the fine-tuning method applied to the CRE task.

5.6 Result Generalization

We release the best performing model resulting from our study, renamed as CLiMA (Causal Linking for Medical Annotation) and make it publicly available as LORA adapters, with associated training scripts and hyperparameters settings, in the code repository of this study (see Appendix D) as well as in the Hugging Face repository: `https://huggingface.co/unica/CLiMA`.

We use CLiMA to test the generalization capability of the fine-tuning method to CRE from medical note MIMICause data through the ADE and Drug Reviews datasets.

ADE In Table 5.7 we report the F1 scores on the 20% sample of ADE positive examples and the balanced random sample of positive and negative instances, for all the decoder-only models fine-tuned on the MIMICause benchmark. The performance level of all the

Table 5.7: F1 scores of the fine-tuned models on a 0.2 random sample and a synthetic, 800-sized balanced sample of ADE case reports.

Model	20% ADE	ADE balanced	
	F1	macro F1	micro F1
Mistral-7B-v0.1	0.956	**0.850**	**0.851**
MistralOrca	0.959	0.766	0.772
Llama-2-7b	0.913	0.698	0.712
Llama-3.1-8B-Instruct	0.978	0.685	0.708
gemma-2-9b	0.948	0.789	0.793
CLiMA	**0.981**	0.838	0.84

models is well aligned with the evaluation on the MIMICause test set, indicating that the causal relation understanding resulting from the instruction fine-tuning generalizes well across partly differing datasets and classification schemas. A slight increase in the F1 scores of some of the models is even noticed, arguably due to the simplified classification task.

Although not strictly comparable, all the models outperform the SOTA encoder-only architectures trained directly on ADE data [HBG24] on the positive example prediction, and a few of them also in the balanced test set. Noticeably, some of the domain-general and lower-sized models, such as *Mistral-7B-v0.1*, reach the best score on the balanced test set.

Drug Reviews Table 5.8 summarizes the Precision scores of CLiMA on the Drug Review sample described in Section 5.2.3, aggregated per causal relation group. As discussed earlier, we calculated a majority vote among groups of three annotators for all 200 causal relationships identified by our algorithm. The human evaluation setup was designed exclusively to validate the correctness of the relationships identified by the model, rather than to identify a complete ground truth of all possible relationships present in the text; therefore, we were only able to compute precision, while we lack knowledge of false negatives required to compute recall or F1-score.

CLiMA achieves an overall precision of 0.73. If we disregard the directionality of the extracted relationships between pairs of entities, the precision slightly increases to 0.76. In both cases, the obtained precision is quite satisfactory, as it closely aligns with the algorithm's overall performance on the original MIMICause test dataset, for which it was specifically trained. This demonstrates the robustness and generalization capabilities of the fine-tuning method.

Looking at the IAA coefficients, we notice that the raters found annotating the `Enable` and `Hinder` relations more challenging. This observation is consistent with the analysis presented in Section 5.5.1.

Table 5.8: Precision scores of CLiMA on the *Drug Reviews (Druglib.com)* data sample aggregated for relation groups, together with average pair-wise Cohen κ and Fleiss κ_F IAA coefficients among human annotators.

Relation Type	Cohen κ	Fleiss κ_F	Precision
Cause	0.706	0.707	0.70
Enable	0.591	0.576	0.60
Prevent	0.831	0.808	0.78
Hinder	0.763	0.746	0.60
Other	0.770	0.741	0.97
Overall	0.739	0.728	0.73

Drug	Causal_Relation	Condition	Support
mirena	Cause	birth control	60
accutane	Cause	acne	33
viibryd	Enable	depression	5
methotrexate	Enable	psoriasis	2
lexapro	Prevent	anxiety	269
vyvanse	Prevent	adhd	142
pristiq	Hinder	depression	17
buspar	Hinder	anxiety	11
nexplanon	Other	birth control	406
aviane	Other	birth control	52

Table 5.9: Sample statements for the 5 causal relation categories extracted by the MedLlama model, with their support values in the *Drug Reviews* dataset.

5.7 Drug Reviews Causal Graphs

Given the successful validation of CliMA on detecting causal relations from drug reviews, we deploy it on the 19,200 instance subset of *Drug Reviews* with metadata entities explicitly matched within the text, and generate a causal drugs knowledge graph (referred to as *CausalDrugsKG*) comprising a total of 19,200 triples, with roughly 3,000 distinct (non-reified) triples, connecting 1,149 unique Drug entities and 322 unique Condition entities via the five causal relation categories, i.e. *Cause*, *Enable*, *Prevent*, *Hinder* and *Other*. Drug and Condition entities are described within the namespace: `http://causaldrugskg.org/causaldrugskg/resource/` (prefix *csldrg*).

In the ontology designed to describe the KG (`http://causaldrugskg.org/causaldrugskg/ontology` namespace, with prefix *csldrg-ont*), each extracted claim is reified into an instance of the *csldrg-ont:Statement* class and associated with the collection of drug reviews it was generated from (using the property *provo:wasDerivedFrom*) and the number of source reviews (*csldrg-ont:hasSupport*). A sample of generated (unreified) triples is illustrated in Table 5.9, together with their supports, while Figure 5.6 illustrates a sample reified statement concerning the resource *causaldrugskg:accutane*

In order to make our causal KG interoperable with other biomedical resources, we

```
csldrg-ont:statement_2 a rdf:Statement ;
 provo:wasDerivedFrom csldrg:157784,
 ...
 csldrg:157915 ;
 csldrg-ont:hasSupport 33 ;
 csldrg-ont:object csldrg:acne ;
 csldrg-ont:predicate csldrg-ont:Cause ;
 csldrg-ont:subject csldrg:accutane .
```

Figure 5.6: A sample reification for the statement assessing a *csldrg-ont:Cause* relation between the instances *csldrg:accutane* and *csldrg:acne*, extracted from 33 reviews (only 2 are shown here for the sake of simplicity.)

linked the triples' Drug and Condition entity metadata with ontologies from the NCBO BioPortal [OPG+17]. The NCBO BioPortal is an open, community-developed repository that currently provides access to a library of as many as 1,190 biomedical ontologies and terminologies in subject matters such as pharmacology, public health, etc., and to over 93M class and property mappings across these ontologies [WNS+11].

The linking is performed by querying the NCBO BioPortal Annotator API[26] with each of the Drug and Condition entity terms and limiting the search to a restricted set of reference ontologies. This *ontologies* parameter was determined by intersecting the sets of ontologies returned by querying the NCBO BioPortal Recommender API[27] with the text of the drug reviews in our dataset.

Overall, 1057 Drug and 268 Condition entities (92% and 83%) in the KG are mapped onto one or many of 23 drug ontologies and 31 condition ontologies via *owl:sameAs* property, with each Drug and Condition entity being mapped onto an average of 5.6 and 8.6 ontologies, respectively. Table 5.10 shows the coverage of the top 10 ontologies, in terms of number of unique Drug and Condition entities in our causal KG.

We made publicly available[28] the automatically generated causal KG in Turtle and RDF serialization format within the European Data portal[29]. The direct link is: `https://jeodpp.jrc.ec.europa.eu/ftp/jrc-opendata/ETOHA/ETOHA-OPEN/CausalDrugsKG.ttl`. Furthermore, we have set up a Virtuoso SPARQL endpoint where *CausalDrugsKG* can be queried, and analytical information on target entities, attributes, and relations can be retrieved in user-specified data formats[30]. As an example, a SPARQL query like the one in Figure 5.7 returns the four statements from the graph having the

[26] `https://data.bioontology.org/documentation#nav_annotator`

[27] `https://data.bioontology.org/recommender`. This endpoint returns a ranked list of appropriate ontologies for a given input text, based on a weighted combination of coverage, acceptance, detail and specialization scores.

[28] Under Creative Commons Attribution 4.0 International (CC BY 4.0).

[29] `https://data.jrc.ec.europa.eu/dataset/acebeb4e-9789-4b5c-97ec-292ce14e75d0`

[30] `https://api-vast.jrc.service.ec.europa.eu/sparql/`

Condition Ontology	Coverage	Drug Ontology	Coverage
OCHV	246	RXNORM	869
IOBC	242	MESH	831
MEDDRA	232	NCIT	812
MESH	229	OCHV	789
MDM	227	MDM	687
SNOMEDCT	226	CHEBI	589
NCIT	212	IOBC	453
NIFSTD	184	SNOMEDCT	353
DOID	178	NDDF	351
MONDO	178	LOINC	294

Table 5.10: Coverage of the top 10 ontologies for Drug and Condition entities in the causal KG.

```
PREFIX csldrg: <http://causaldrugskg.org/causaldrugskg/resource/>
PREFIX csldrg-ont: <http://causaldrugskg.org/causaldrugskg/ontology#>
SELECT ?statement
FROM <CausalDrugsKG>
WHERE { ?statement a rdf:Statement .
 ?statement csldrg-ont:subject csldrg:accutane . }
```

Figure 5.7: Sample SPARQL query returning all *CausalDrugsKG* statements with the graph entity `csldrg:accutane` as *csldrg-ont:subject*.

target Drug entity *csldrg:accutane* as subject. For each result statement, a link to its corresponding URL in a Virtuoso Faceted Browser endpoint[31] is returned, allowing further navigation.

Data Analytics Dashboard We provide aggregated analyses of *CausalDrugsKG* through an interactive visualization dashboard accessible as an Hugging Face Spaces page at `https://huggingface.co/spaces/zavavan/CausalDrugsKG_Dashboard`. The dashboard's "Top Key Entities" panel shows the 30 most frequently occurring Drug and Condition entities in the graph, with over 15% of the extracted triples (out of the 19,200) having *birth control* as Condition, followed by *pain*, *depression* and *anxiety*. Clicking on the drug and condition names on the right-side legend redirects to the corresponding *CausalDrugsKG* entity page in the Virtuoso Faceted Browser.

The next plot illustrates the distribution of entity types within the KG, specifically focusing on conditions and drugs. This provides an overview of how frequently each entity type appears in the dataset. Another plot highlights the ontology coverage, showing the proportion of entities, again categorized as drugs and conditions, that have been

[31] `https://api-vast.jrc.service.ec.europa.eu/fct/`

successfully linked to biomedical concepts in external bio-ontologies. This helps assessing which standardized terminologies are represented within the KG and to what extent.

Finally, the "Causal Relation Chord Diagrams" panels for each of the five causal relations present the most connected entity pairs within the KG through chord illustrations, with the size of the chords encoding the frequency support of the depicted relations. Figure 5.8 contains a condensed visualization where we pre-filtered the relations with a minimum number of 70 occurrences. In the KG, the two relations with such a support are *Prevent* (in blue) and *Other* (in orange).

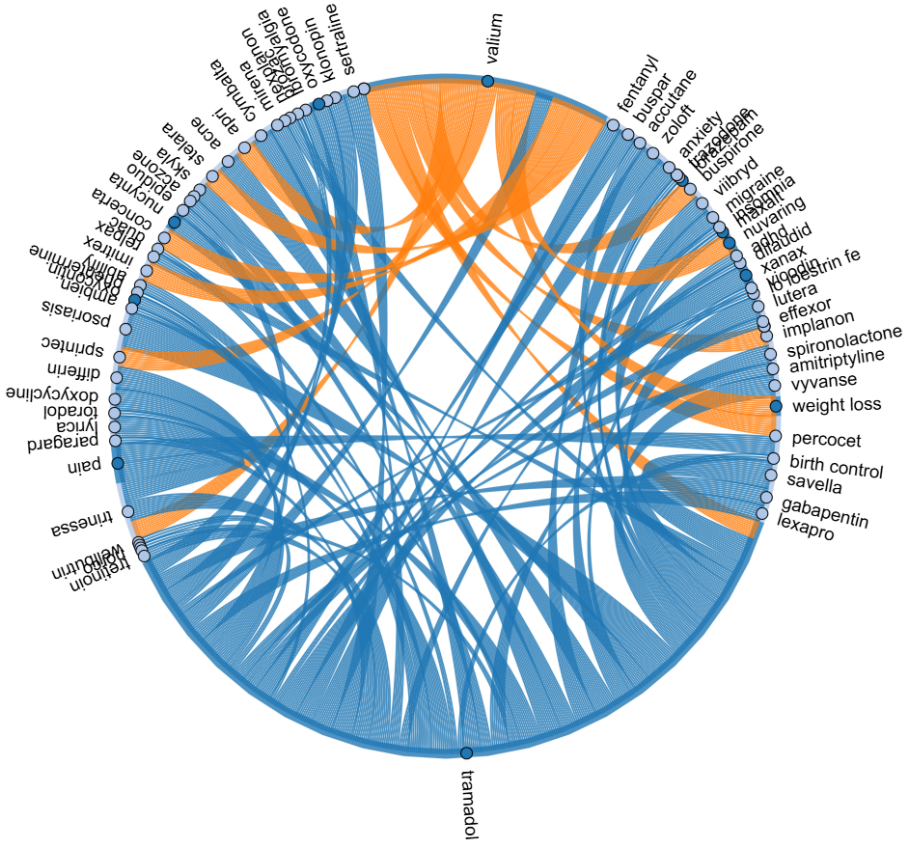


Figure 5.8: Chord diagram illustrating the strength (statement support) of relationships among top Drug and Condition entities in *CausalDrugsKG*.

Chapter 6

Conclusions

6.1 Research results

The increasing availability of unstructured data in natural language has opened unprecedented opportunities for automatic KG generation systems to extract complex knowledge structures and support actionable data analysis services for a wide range of domains and application scenarios. In this thesis, we experimented with the application of extractive techniques from NLP, ML and generative AI, coupled with SW data linking best practices, to the construction of interoperable knowledge infrastructures, supporting fine-grained data analytics and trend analysis.

For DT monitoring, we presented an unsupervised information extraction pipeline optimized to generate open-domain KGs from micro-blogging text, without relying on a target domain ontology schema in the extraction process. In a test tweet collection the pipeline proved to outperform some of the state-of-the-art methods, generating highly accurate triples, with around 12% of entities linked to DBpedia entries. We show the potential usefulness of the generated KG infrastructure for tracking relevant entities in the DT ecosystem, for example as a knowledge plug-in for RAG interfaces. When transferred to a news corpus, the algorithm has shown to scale linearly with the document set size and enabled the generation of a large scale KG of Digital Health, with around 8% of the 86k extracted entities linked to DBpedia entries of domain relevant types like *dbpedia:Disease*, *dbpedia:Company* and *dbpedia:Drug*. A preliminary data analytics interface to the graph highlighted its potential for generating insights on trends and key players in the digital health sector.

In the AECO research use case, we proved that leveraging deep text understanding abilities of instruction-tuned LLMs enables to customize an existing Information Extraction pipeline to out-of-domain data, with minimal annotation effort. Moreover, when applied to a large dataset of scientific papers, partitioned by an optimized neural topic model, the resulting KG generation pipeline allows the detection of insightful trends on predominant Tasks and Methods, specific to major macro-areas in the domain. The integration of an upstream topic model is beneficial for a fine-grained research trend

analysis, as it makes it more sensible to low signals which otherwise would not emerge at the full collection level. As an example, computing triple statistics for the macro-topic "Energy Efficiency and Thermal Comfort in Building Environments" allows to detect as predominant methods highly specialized tools such as the energy simulation package EnergyPlus™ or the 3D software model Envi-met, along with more generic concepts such as Information Technology. To the best of our knowledge, this establishes a novel framework that could be transferable to the large scale exploration of other research fields.

Finally, by extensively benchmarking LLM architectures and learning paradigms, we demonstrated that instruction fine-tuned models can reach strong performance levels on detecting causal relations from multi-type entities in biomedical text. We found that, while domain-specific pretraining enhances model capabilities compared to general purpose LLMs, medical LLMs show improved performance when they can leverage the complex causal structures specific to the train datasets. At the same time, these models seem to generalize a notion of causality across varying contexts, as it is observed by the robust performance across clinical notes, medical case reports and drug reviews. This highlights their potential for diverse biomedical applications. In fact, we designed an end-to-end pipeline for constructing a KG of drug-condition causal relationships mined from patient-authored drug reviews, deploying an LLM trained on the MIMICause dataset and off-the-shelf entity linking libraries.

Overall, these results help us addressing the research questions from Section 1.2:

Q1:*How NLP and Semantic Web technologies can be combined to extract knowledge from noisy user-generated text collections and represent it in interoperable formats?*

The DT monitoring use case showed that a suitable combination of lightweight pre-processing, standard NLP tools (like dependency parsing) with a word embedding-based clustering approach, enables to generate and link valid and informative triples from noisy social media posts, using no training data and minimal system engineering.

Q2:*Which NLP and ML techniques better fit different application scenarios?*

Use cases 2 and 3, while quite heterogeneous, both suggest that, in scenarios when a target entity-relation schema is provided, tuning pre-trained LLMs on labeled data, no matter how sparse, is the most effective option. In challenging domains such as clinical notes, this method largely outperforms SOTA encoder architectures (BERT) and in-context learning techniques[1]. In low-resource use cases like number 2, manually annotating a small training set is cost-effective compared to relying on zero- or few-shot model capabilities.

However, in open domain scenarios with no schema definition and no labeled data like use case 1, syntactical NLP methods are a scalable solution to bootstrap an emerging set of entities and relations.

Q3: *How can KG representations enhance the analysis of trends within a specific*

[1]As LLM training techniques, particularly for reasoning-intensive inference, are evolving fast, we underline that our conclusions are to be taken "as of" the time of our latest evaluation run, i.e. January 2025.

domain?

As we illustrate in the use case 2, KGs support the creation of trend analysis that are grounded on semantically transparent relations and can be tracked back to the text documents that generated them, like the *Task-Method* time series in Chapter 4.

Q4:*Can Generative AI techniques support the construction of scientific knowledge graphs of very abstract research concepts in technical domains, with only limited customization?*

The evaluation of the *SKG-AECO* pipeline in Section 4.5.3 proved that, by tuning a small- to mid-sized foundational LLM on sparse, high-quality labeled data, and by linking to suitable sections of large multi-domain KBs like Wikidata and DBpedia, a large scale graph of research concepts in a multidisciplinary and highly technical domain such as AECO can be built and made partially interoperable.

Q5:*Which Generative AI techniques and LLM architecture and training methods better adapt to the generation of causal graphs from medical texts?*

As we observed already, instruction fine-tuning shows stronger performance over few-shot learning or reasoning-intensive prompt techniques like CoT or prompt chaining, with also higher stability across relation labels.

6.2 Limitations and future developments

We finally discuss a few general limitations affecting some of presented use cases, outlining also directions of future developments.

6.2.1 Digital Transformation Monitoring

The task configuration of the DT monitoring use case limits a full exploitation of the generated data. As the generation pipeline for the *DTSMM_KG* graph does not rely on an ontology specification of the target domain for tailoring the entity and relation extraction process, the extracted entities[2] are currently not type-classified, which prevents the execution of more structured queries on the graph.

Therefore, we aim to work on an enhanced version of the pipeline that builds upon the entity and relation spans generated with the current approach and further categorizes them into a fine-grained schema tailored to the specific domain, leveraging contextual embedding vector representations.

For the *DHNEWS_KG*, we are considering extending the entity linking to technology KGs in the health domain, such as the recently released PKG 2.0 [XYX+25] that integrates data sources like research papers, patents and research projects. The analytical goal here is to detect insights about the societal and market impact of trending technologies, as mirrored in the news discourse, and reveal interesting patterns in the innovation process.

[2]Namely, the *emerging* entities that are not linked to DBpedia and can not inherit DBpedia type categories.

6.2.2 Scientific Knowledge Graphs

A recurrent limitation of KG generation pipelines is triple data sparseness.

We illustrate this in Figure 6.1, showing the log scale triple distribution over triple support levels, i.e. the number of documents from which a triple was extracted. One can notice as most of the data points cluster around the average support value (1.15) with a long tail stretching towards higher values.

Triple data sparseness turns out to be a significant bottleneck for the research trend analysis in Section 4.7.2, which requires the triples extracted by the IE module to be not only semantically correct but also abstract enough to detect a representative sample of triple instances over time.

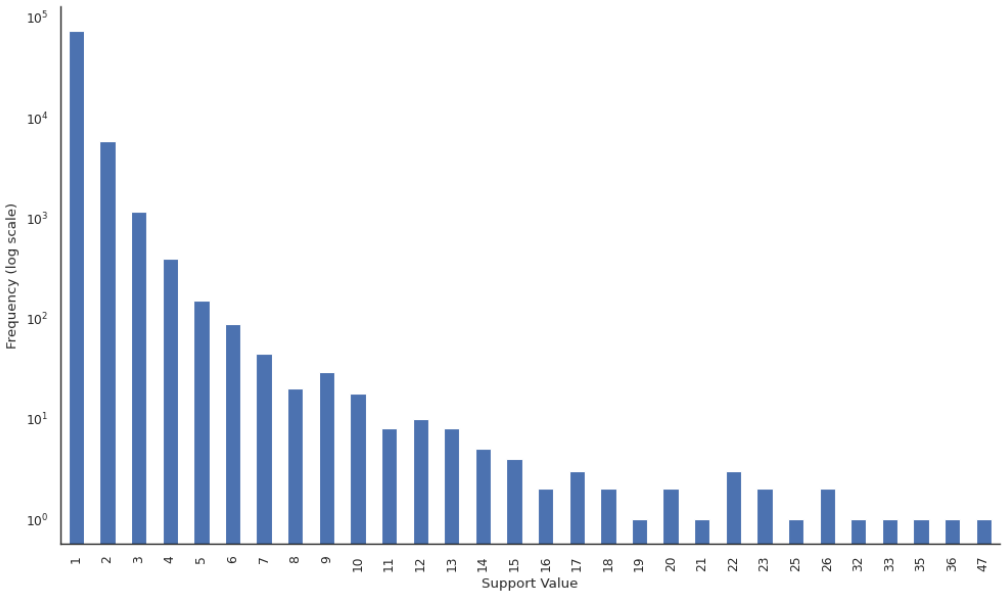


Figure 6.1: Log scale triple distribution over support, for the $\langle Method; Used-for; Task\rangle$ and $\langle Method; Used-for; Method\rangle$ triple set used in the trend analysis.

This right-skewed distribution is mainly due to poor entity generalization, with triple subject and object entity average support ranging between 4.43 and 4.91. In the case of the *SKG-AECO* pipeline, this stems from a sub-optimal performance of the entity merging mechanism described in Section 4.5.3. In fact, to reduce computational complexity, the Transformer-based merging only attempts to merge candidate entities that share at least one token, which prevents to merge near-equivalent concepts such as, e.g., *"photovoltaics"*,*"solar pvs"* and *"photovoltaic system"*. On the other hand, the lowered similarity threshold we applied to boost entity merging generates inaccurate merging, as discussed in Section 3.8. An alternative solution we are currently exploring consists of: 1. fine-tuning the underlying *paraphrase-distilroberta-base-v2* entity embedding model to our target domain by contrastive learning (using a subset of highly reliable merge pairs

as training data); 2. computing an optimized clustering on the entire set of candidate entities based on this domain-tuned model.

Another possible reason lies in the low recall of the LLM-based triple extraction module we integrated. In fact, the SciERC AECO dataset we curated for training the LLM module is based on a simplified SciERC schema that does not include entity co-reference resolution links. Given that we only process title and abstract of the research papers, this might significantly reduce the recall rate of the triple extraction process. As an example, in the abstract passage below the link between the Method *"Intelligent Lighting Control System"* and Task *"corridor lighting"* is missed because the cross-sentence co-reference to the Method entity (*"The system"*) is not recognized:
(2) *"Design of Intelligent Lighting Control System Based on Vivado Environment. The system can be applied to corridor lighting inside and outside, which includes [...]"*.

We plan to extend the annotation of the SciERC AECO dataset with entity co-reference and re-train the LLM module on this new release, testing for the tradeoff between the recall gain and the error rate that this might introduce.

Upstream of the *SKG-AECO* pipeline, one source of the triple sparseness issue is the relatively low size of the input clustered data. One limitation of our approach is that, differently than for relation clustering in Section 3.7, we optimize topic clustering based on topic coherence metrics only, without penalizing for the ratio of outlier data points discarded by HDBSCAN, therefore ending up by filtering out a significant fraction of the original data.

In order to characterize the relation between topic modeling quality and data representativeness, Fig. 5 in Appendix C shows the variation of average topic coherence with respect to the ratio of clustered data points[3] for the 10,000 random document sample of Section 4.4, over variations of the HDBSCAN hyperparameters. One can notice that the most coherent topics are stably discovered within a 25-40% range of outlier ratio, while coherence starts degrading when outlier ratio drops towards 0. This indicates a clear trade off between discovering compact, high-quality clusters and clustering higher portions of the data.

For application scenarios where accurate topic-based analysis is prioritized, a baseline option of simply discarding outlier documents may be applied, which can lead to data sparseness problems for the downstream cluster-based processing, depending on the size of the source dataset. The methodology employed in our experimental study may consequently limit the representativeness of the downstream analyses with respect to the original dataset.

However, this limitation can be mitigated by for example re-assigning outlier articles to discovered topics without re-computing the optimized topic model, by performing outlier topic imputation methods (e.g. based on best matching c-TF-IDF topic representations or embedding vector cosine distance). Further empirical testing is required to assess the impact of either approach on the topic representativeness of the downstream bibliometric and information extraction analyses.

[3] That is, 1 - outlier_ratio.

Finally, a major line of development of the present AECO research graph is a further integration of heterogeneous graph representations of the same domain. For instance, while not described in this thesis, we performed a bibliometric co-authorship analysis on the same AECO paper dataset, resulting in a graph of weighted collaboration triples between research institutions (see journal publication ii. in the Dissemination section). Integrating these sub-graphs would allow for example to selectively retrieve co-authorship relations relative to subject matters like the Method entities extracted from co-authored papers, hence supporting advanced queries like: *"Retrieve institutions collaborating on the application of method X for solving Task Y"*. Full integration of these relations into a comprehensive AECO research graph will contribute to a more fine-grained picture of the research innovation process.

6.2.3 Causality Graphs from biomedical text

The main limitation of the causal KG experiments relate to the limited generalization of the models' causal understanding.

While the fine-tuned MedLlama model demonstrated strong performance across both the ADE and Drug Reviews datasets, it was primarily trained on the low-sized MIMI-Cause dataset, which may not encompass the full spectrum of linguistic nuances present in broader biomedical texts. Additionally, the model's lower performance on the less frequent `Enable` and `Hinder` relations, indicates a need for larger and more diverse annotated datasets to improve model robustness and the learning of these complex relations. Furthermore, we fine-tuned models on a narrow domain due to the scarcity of available gold-standard datasets for training. This specificity might limit the model's applicability to broader biomedical contexts. To enhance generalization, future research could focus on developing gold-standard datasets from a wider array of health domains, allowing for the inclusion of more nuanced and fine-grained causal relations. This expansion would enable models to better capture the complexity and diversity of causal relationships in various biomedical fields.

We also acknowledge that our approach is primarily focused on the classification of causal relations rather than on deeper causal reasoning. While our models effectively extract causal links from biomedical texts, they do not yet engage in causal reasoning, which involves understanding the underlying mechanisms and implications of these relations. Expanding our work to include causal reasoning would enable the models to provide insights into the causal pathways and potential outcomes, offering a more comprehensive understanding of biomedical phenomena.

Additionally, our exclusive focus on relation classification problem left unsolved the task of biomedical entity detection and linking, which is a crucial building block of an end-to-end causal KG construction pipeline, and a challenging tasks due to the variety of technical terminologies, abbreviations, commercial labeling and even slang expressions found in clinical and patient-authored text collections [WPL$^+$16]. We plan to test with integrating existing biomedical NER algorithms for scaling the coverage of our prototype *CausalDrugsKG* graph.

In particular, we are currently benchmarking the applicability of causal graph generation pipelines for a use of pharmacovigilance on Adverse Drug Events from crowdsourced patient-authored reviews on off-label drug prescriptions.

Finally, we would like to conclude with a consideration on the security and privacy protection of the used data and the released models. Please note that all datasets used in this study are publicly available and have been de-identified in accordance with privacy regulations prior to their release. Namely, MIMICause, derived from MIMIC-III, underwent rigorous de-identification procedures including removal of protected health information (PHI) following HIPAA Safe Harbor guidelines[4], while the ADE corpus was constructed from anonymized MEDLINE case reports, and the Drug Reviews dataset contains only user-generated content without personal identifiers.

Regarding our released fine-tuned models, we acknowledge potential memorization risks inherent in large language models trained on clinical text. To mitigate these concerns, we recommend that users of our released checkpoints: (1) apply additional privacy-preserving techniques when deploying models in production environments, (2) avoid training on or exposing the models to any non-de-identified patient data, and (3) implement appropriate access controls and usage monitoring. Furthermore, our models should be used strictly for research purposes and require proper ethical review before any clinical deployment. We emphasize that while our training data is de-identified, users must remain vigilant about potential re-identification risks and ensure compliance with local data protection regulations (e.g. GDPR[5]) in their specific deployment contexts.

[4] https://www.hhs.gov/hipaa/for-professionals/special-topics/de-identification/index.html

[5] https://eur-lex.europa.eu/eli/reg/2016/679/oj/eng

Bibliography

[AGP+17] Harith Alani, Aldo Gangemi, Valentina Presutti, Diego Reforgiato Recupero, Andrea Giovanni Nuzzolese, Francesco Draicchio, and Misael Mongiovì. Semantic Web Machine Reading with FRED. *Semantic Web*, 8(6):873–893, 2017.

[AJPM15] Gabor Angeli, Melvin Jose Johnson Premkumar, and Christopher D. Manning. Leveraging linguistic structure for open domain information extraction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 344–354, Beijing, China, July 2015. Association for Computational Linguistics.

[AKP+18] Sören Auer, Viktor Kovtun, Manuel Prinz, Anna Kasprzik, Markus Stocker, and Maria Esther Vidal. Towards a knowledge graph for science. In *Proceedings of the 8th International Conference on Web Intelligence, Mining and Semantics*, WIMS '18, pages 1–6, New York, NY, USA, 2018. Association for Computing Machinery.

[ALTdJ+23] Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebrón, and Sumit Sanghai. GQA: Training Generalized Multi-Query Transformer Models from Multi-Head Checkpoints. In *EMNLP 2023 - 2023 Conference on Empirical Methods in Natural Language Processing, Proceedings*, pages 4895 – 4901, 2023.

[AM21] Abbas Akkasi and Mari-Francine Moens. Causal relationship extraction from biomedical text using deep neural models: A comprehensive survey. *Journal of Biomedical Informatics*, 119:103820, 2021.

[AMGOLOV20] Tareq Al-Moslmi, Marc Gallofré Ocaña, Andreas L. Opdahl, and Csaba Veres. Named entity extraction for knowledge graphs: A literature overview. *IEEE Access*, 8:32862–32881, 2020.

[BCM22] L. Barbaglia, S. Consoli, and S. Manzan. Forecasting with economic news. *Journal of Business & Economic Statistics*, 41(3):708–719, 2022.

[BESPV13] Anandhi Bharadwaj, Omar A. El Sawy, Paul A. Pavlou, and N. Venkatraman. Digital business strategy: toward a next generation of insights. *MIS Q.*, 37(2):471–482, June 2013.

[BFVX19] Carlos Barbosa, Lucas Félix, Vinícius Vieira, and Carolina Xavier. Sara - A Semi-Automatic Framework for Social Network Analysis. In *Anais Estendidos do XXV Simpósio Brasileiro de Sistemas Multimídia e Web*, pages 59–62, Porto Alegre, RS, Brasil, 2019. SBC.

[BH21] Fatima Batool and Christian Hennig. Clustering with the Average Silhouette Width. *Computational Statistics and Data Analysis*, 158, 2021.

[BLC19a] Iz Beltagy, Kyle Lo, and Arman Cohan. SciBERT: A pretrained language model for scientific text. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3615–3620, Hong Kong, China, November 2019. Association for Computational Linguistics.

[BLC19b] Iz Beltagy, Kyle Lo, and Arman Cohan. Scibert: A pretrained language model for scientific text. *arXiv preprint arXiv:1903.10676*, 2019.

[BM05] Razvan Bunescu and Raymond Mooney. A shortest path dependency kernel for relation extraction. In Raymond Mooney, Chris Brew, Lee-Feng Chien, and Katrin Kirchhoff, editors, *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 724–731, Vancouver, British Columbia, Canada, October 2005. Association for Computational Linguistics.

[BM15] Lutz Bornmann and Rüdiger Mutz. Growth rates of modern science: A bibliometric analysis based on the number of publications and cited references. *Journal of the Association for Information Science and Technology*, 66(11):2215–2222, 2015.

[BMR+20] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, and Dhariwal et al. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020.

[BPC20] Iz Beltagy, Matthew E. Peters, and Arman Cohan. Longformer: The long-document transformer, 2020.

[CBF20] Filippo Chiarello, Andrea Bonaccorsi, and Gualtiero Fantoni. Technical sentiment analysis. measuring advantages and drawbacks of new products using social media. *Computers in Industry*, 123:103299, 2020.

[CBM22] S. Consoli, S. Barbaglia, and S. Manzan. Fine-grained, aspect-based sentiment analysis on economic and financial lexicon. *Knowledge-Based Systems*, 247:108781, 2022.

[CCPB22] Marco Colagrossi, Sergio Consoli, Francesco Panella, and Luca Barbaglia. Tracking socio-economic activities in european countries with unconventional data. In *ACM International Conference Proceeding Series*, page 323 – 330, 2022.

[CHL+24] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tai, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. Scaling instruction-finetuned language models. *J. Mach. Learn. Res.*, 25(1), January 2024.

[CJSF24] Shaobo Cui, Zhijing Jin, Bernhard Schölkopf, and Boi Faltings. The odyssey of commonsense causality: From foundational benchmarks to cutting-edge reasoning, 2024.

[CLB+17] Paul F. Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 4302–4310, Red Hook, NY, USA, 2017. Curran Associates Inc.

[CP17] Philipp Cimiano and Heiko Paulheim. Knowledge graph refinement: A survey of approaches and evaluation methods. *Semant. Web*, 8(3):489–508, January 2017.

[DAGY+25] DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, and et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025.

[DCLT19] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

[DDL+24] John Dagdelen, Alexander Dunn, Sanghoon Lee, Nicholas Walker, Andrew S Rosen, Gerbrand Ceder, Kristin A Persson, and Anubhav Jain. Structured information extraction from scientific text with large language models. *Nature Communications*, 15(1):1418, 2024.

[DJHM13] 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, page 121–124, New York, NY, USA, 2013. Association for Computing Machinery.

[DKC+22] Jens Dörpinghaus, Sonja Klante, Martin Christian, Christof Meigen, and Carsten Düing. From social networks to knowledge graphs: A plea for interdisciplinary approaches. *Social Sciences & Humanities Open*, 6(1):100337, 2022.

[dMDS+14] Marie-Catherine de Marneffe, Timothy Dozat, Natalia Silveira, Katri Haverinen, Filip Ginter, Joakim Nivre, and Christopher D. Manning. Universal Stanford dependencies: A cross-linguistic typology. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Hrafn Loftsson, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 4585–4592, Reykjavik, Iceland, May 2014. European Language Resources Association (ELRA).

[DOB+25] Danilo Dessí, Francesco Osborne, Davide Buscaldi, Diego Reforgiato Recupero, and Enrico Motta. Cs-kg 2.0: A large-scale knowledge graph of computer science. *Scientific Data*, 12(1):964, 2025.

[DOR+22a] Danilo Dessì, Francesco Osborne, Diego Reforgiato Recupero, Davide Buscaldi, and Enrico Motta. CS-KG: A large-scale knowledge graph of research entities and claims in computer science. In Ulrike Sattler, Aidan Hogan, C. Maria Keet, Valentina Presutti, João Paulo A. Almeida, Hideaki Takeda, Pierre Monnin, Giuseppe Pirrò, and Claudia d'Amato, editors, *The Semantic Web - ISWC 2022 - 21st International Semantic Web Conference, Virtual Event, October 23-27, 2022, Proceedings*, volume 13489 of *Lecture Notes in Computer Science*, pages 678–696. Springer, 2022.

[DOR+22b] Danilo Dessì, Francesco Osborne, Diego Reforgiato Recupero, Davide Buscaldi, and Enrico Motta. SCICERO: A deep learning and NLP approach for generating scientific knowledge graphs in the computer science domain. *Knowledge-Based Systems*, 258:109945, 2022.

[DOR+22c] Danilo Dessí, Francesco Osborne, Diego Reforgiato Recupero, Davide Buscaldi, and Enrico Motta. Scicero: A deep learning and nlp approach for generating scientific knowledge graphs in the computer science domain. *Knowledge-Based Systems*, 258:109945, 2022.

[DORR+20] Danilo Dessì, Francesco Osborne, Diego Reforgiato Recupero, Davide Buscaldi, Enrico Motta, and Harald Sack. Ai-kg: an automatically generated knowledge graph of artificial intelligence. In *International semantic web conference*, pages 127–143. Springer, 2020.

[DORR+21] Danilo Dessì, Francesco Osborne, Diego Reforgiato Recupero, Davide Buscaldi, and Enrico Motta. Generating knowledge graphs by employing natural language processing and machine learning techniques within the scholarly domain. *Future Generation Computer Systems*, 116:253–264, 2021.

[DORR+22] Danilo Dessí, Francesco Osborne, Diego Reforgiato Recupero, Davide Buscaldi, and Enrico Motta. Cs-kg: A large-scale knowledge graph of research entities and claims in computer science. In *International Semantic Web Conference*, pages 678–696. Springer, 2022.

[ETC+25] Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitansky, Robert Osazuwa Ness, and Jonathan Larson. From local to global: A graph rag approach to query-focused summarization, 2025.

[EU20] Markus Eberts and Adrian Ulges. Span-based joint entity and relation extraction with transformer pre-training. In *ECAI 2020*, pages 2006–2013. IOS Press, 2020.

[FCea24] P. Fernainy, A.A. Cohen, and Murray E. et al. Rethinking the pros and cons of randomized controlled trials and observational studies in the era of big data and advanced methods: A panel discussion. *BMC Proc 18 (Suppl 2)*, 2024.

[FKM+22] Amir Feder, Katherine A. Keith, Emaad Manzoor, Reid Pryzant, Dhanya Sridhar, Zach Wood-Doughty, Jacob Eisenstein, Justin Grimmer, Roi Reichart, Margaret E. Roberts, Brandon M. Stewart, Victor Veitch, and Diyi Yang. Causal inference in natural language processing: Estimation, prediction, interpretation and beyond. *Transactions of the Association for Computational Linguistics*, 10:1138–1158, 10 2022.

[FQ15] Rosa Falotico and Piero Quatto. Fleiss' kappa statistic without paradoxes. *Quality and Quantity*, 49(2):463 – 470, 2015.

[GKMZ18] Felix Gräßer, Surya Kallumadi, Hagen Malberg, and Sebastian Zaunseder. Aspect-Based Sentiment Analysis of Drug Reviews Applying Cross-Domain and Cross-Data Learning. In *Proceedings of the 2018 International Conference on Digital Health*, DH '18, page 121–125, New York, NY, USA, 2018. Association for Computing Machinery.

[Gro20] Maarten Grootendorst. Keybert: Minimal keyword extraction with bert., 2020.

[Gro22] Maarten Grootendorst. Bertopic: Neural topic modeling with a class-based tf-idf procedure, 2022.

[GRR+12] Harsha Gurulingappa, Abdul Mateen Rajput, Angus Roberts, Juliane Fluck, Martin Hofmann-Apitius, and Luca Toldo. Development of a benchmark corpus to support the automatic extraction of drug-related adverse effects from medical case reports. *Journal of Biomedical Informatics*, 45(5):885–892, 2012.

[Gwe08] Kilem Li Gwet. Computing inter-rater reliability and its variance in the presence of high agreement. *British Journal of Mathematical and Statistical Psychology*, 61(1):29–48, 2008.

[HBG24] Moritz Hennen, Florian Babl, and Michaela Geierhos. ITER: Iterative transformer-based entity recognition and relation extraction. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 11209–11223, Miami, Florida, USA, November 2024. Association for Computational Linguistics.

[HKK+10] Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. SemEval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In Katrin Erk and Carlo Strapparava, editors, *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 33–38, Uppsala, Sweden, July 2010. Association for Computational Linguistics.

[HM17] Matthew Honnibal and Ines Montani. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear, 2017.

[HMQ+24] Wei Huang, Xudong Ma, Haotong Qin, Xingyu Zheng, Chengtao Lv, Hong Chen, Jie Luo, Xiaojuan Qi, Xianglong Liu, and Michele Magno. How good are low-bit quantized llama3 models? an empirical study. *arXiv preprint arXiv:2404.14047*, 2024.

[HMRHLA20] Andreas Hotho, Jose L. Martinez-Rodriguez, Aidan Hogan, and Ivan Lopez-Arevalo. Information extraction meets the semantic web: A survey. *Semant. Web*, 11(2):255–335, January 2020.

[Hol79] Sture Holm. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6:65–70, 1979.

[HPKY23] Myungho Han, Jinsuck Park, Inhan Kim, and Hwang Yi. A microalgae photobioreactor system for indoor air remediation: Empirical examination of the co2 absorption performance of spirulina maxima in a nahco3-reduced medium. *Applied Sciences*, 13(24), 2023.

[HR05] George Hripcsak and Adam S. Rothschild. Agreement, the f-measure, and reliability in information retrieval. *Journal of the American Medical Informatics Association*, 12(3):296–298, 2005.

[HSG04] Takaaki Hasegawa, Satoshi Sekine, and Ralph Grishman. Discovering relations among named entities from large corpora. ACL '04, page 415–es, USA, 2004. Association for Computational Linguistics.

[HYS20] Qi He, Jaewon Yang, and Baoxu Shi. Constructing knowledge graph for social networks in a deep and holistic way. In *Companion Proceedings of the Web Conference 2020*, WWW '20, page 307–308, New York, NY, USA, 2020. Association for Computing Machinery.

[HZL+11] Raphael Hoffmann, Congle Zhang, Xiao Ling, Luke Zettlemoyer, and Daniel S. Weld. Knowledge-based weak supervision for information extraction of overlapping relations. In Dekang Lin, Yuji Matsumoto, and Rada Mihalcea, editors, *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 541–550, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.

[JGMW+22] Bernal Jimenez Gutierrez, Nikolas McNeal, Clayton Washington, You Chen, Lang Li, Huan Sun, and Yu Su. Thinking about GPT-3 in-context learning for biomedical IE? think again. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 4497–4512, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics.

[JOF+19] Mohamad Yaser Jaradeh, Allard Oelen, Kheir Eddine Farfar, Manuel Prinz, Jennifer D'Souza, Gábor Kismihók, Markus Stocker, and Sören Auer. Open research knowledge graph: Next generation infrastructure

for semantic scholarly knowledge. In *Proceedings of the 10th International Conference on Knowledge Capture*, K-CAP '19, page 243–246, New York, NY, USA, 2019. Association for Computing Machinery.

[JPS+16] Alistair EW Johnson, Tom J Pollard, Lu Shen, Li-wei H Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. Mimic-iii, a freely accessible critical care database. *Scientific Data*, 3(1):1–9, 2016.

[JSM+23a] Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023.

[JSM+23b] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023.

[KDTZ26] Wei-Kun Kong, Yiping Duan, Xiaoming Tao, and Yueran Zu. Ride: Redensification-based intrinsic density estimation for knowledge graphs. *Pattern Recognition*, 169:111876, 2026.

[KGR+23] Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners, 2023.

[Kil16] Halil Kilicoglu. Inferring implicit causal relationships in biomedical literature. In *BioNLP 2016 - Proceedings of the 15th Workshop on Biomedical Natural Language Processing*, page 46 – 55, 2016.

[KRH+22] Vivek Khetan, Md Imbesat Hassan Rizvi, Jessica Huber, Paige Bartusiak, Bogdan Sacaleanu, and Andrew Fano. MIMICause: Representation and automatic extraction of causal relation types from clinical notes. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, page 764 – 773, 2022.

[LBM+22] Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8086–8098, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[LBS+16] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. Neural architectures for named entity recognition. In Kevin Knight, Ani Nenkova, and Owen Rambow, editors, *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270, San Diego, California, June 2016. Association for Computational Linguistics.

[LC05] Matthew Lease and Eugene Charniak. Parsing biomedical literature. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 3651 LNAI, page 58 – 69, 2005.

[LGW+23] Wing Lian, Bleys Goodson, Guan Wang, Eugene Pentland, Austin Cook, Chanvichet Vong, and "Teknium". Mistralorca: Mistral-7b model instruct-tuned on filtered openorcav1 gpt-4 dataset. `https://huggingface.co/Open-Orca/Mistral-7B-OpenOrca`„ 2023.

[LHOH18] Yi Luan, Luheng He, Mari Ostendorf, and Hannaneh Hajishirzi. Multi-Task Identification of Entities, Relations, and Coreference for Scientific Knowledge Graph Construction. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3219–3232, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.

[LIJ+15] Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N. Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick van Kleef, Sören Auer, and Christian Bizer. DBpedia–A large-scale, multilingual knowledge base extracted from Wikipedia. *Semantic Web*, 6:167–195, 2015.

[LLB+24] Li Lishuang, Mi Liteng, Zhang Beibei, Xiang Yi, Feng Yubo, Qin Xueyang, and Tang Jingyao. Biomedical event causal relation extraction by reasoning optimal entity relation path. In Maosong Sun, Jiye Liang, Xianpei Han, Zhiyuan Liu, and Yulan He, editors, *Proceedings of the 23rd Chinese National Conference on Computational Linguistics (Volume 1: Main Conference)*, pages 1087–1098, Taiyuan, China, July 2024. Chinese Information Processing Society of China.

[LLS+15] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. Learning entity and relation embeddings for knowledge graph completion. AAAI'15, page 2181–2187. AAAI Press, 2015.

[LNB14] Jey Han Lau, David Newman, and Timothy Baldwin. Machine reading tea leaves: Automatically evaluating topic coherence and topic model quality. In Shuly Wintner, Sharon Goldwater, and Stefan Riezler, editors, *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 530–539, Gothenburg, Sweden, April 2014. Association for Computational Linguistics.

[LPP+20] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, pages 9459–9474, Red Hook, NY, USA, 2020. Curran Associates Inc.

[LWK23] Guozheng Li, Peng Wang, and Wenjun Ke. Revisiting large language models as zero-shot relation extractors. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 6877–6892, Singapore, December 2023. Association for Computational Linguistics.

[LYF+23a] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Comput. Surv.*, 55(9), January 2023.

[LYF+23b] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM computing surveys*, 55(9):1–35, 2023.

[MB20] Claudia Malzer and Marcus Baum. A hybrid approach to hierarchical density-based cluster selection. In *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, volume 2020-September, page 223 – 228, 2020.

[MBSJ09] Mike Mintz, Steven Bills, Rion Snow, and Daniel Jurafsky. Distant supervision for relation extraction without labeled data. In Keh-Yih Su, Jian Su, Janyce Wiebe, and Haizhou Li, editors, *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 1003–1011, Suntec, Singapore, August 2009. Association for Computational Linguistics.

[MCCD13] Tomas Mikolov, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *International Conference on Learning Representations*, 2013.

[McH12] Mary L. McHugh. Interrater reliability: The kappa statistic. *Biochemia Medica*, 22(3):276 – 282, 2012.

[MCHS23] Yubo Ma, Yixin Cao, Yong Hong, and Aixin Sun. Large language model is not a good few-shot information extractor, but a good reranker for hard samples! In *Findings of the Association for Computational Linguistics: EMNLP 2023*, page 10572–10601. Association for Computational Linguistics, 2023.

[MHM20] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction, 2020.

[MKCM24] Reagan Mozer, Aaron R. Kaufman, Leo A. Celi, and Luke Miratrix. Leveraging text data for causal inference using electronic health records, 2024.

[MMJ+23] Subhabrata Mukherjee, Arindam Mitra, Ganesh Jawahar, Sahaj Agarwal, Hamid Palangi, and Ahmed Awadallah. Orca: Progressive Learning from Complex Explanation Traces of GPT-4, 2023.

[Mol25] Christoph Molnar. *Interpretable Machine Learning*. 3 edition, 2025.

[MRLARA18] Jose L. Martinez-Rodriguez, Ivan Lopez-Arevalo, and Ana B. Rios-Alvarado. OpenIE-based approach for Knowledge Graph construction from text. *Expert Systems with Applications*, 113:339–355, 2018.

[Ope23] OpenAI. Gpt-4 technical report, 2023.

[OPG+17] Christopher Ochs, Yehoshua Perl, James Geller, Sivaram Arabandi, Tania Tudorache, and Mark A. Musen. An empirical analysis of ontology reuse in BioPortal. *Journal of Biomedical Informatics*, 71:165 – 177, 2017.

[Pit37] E. J. G. Pitman. Significance tests which may be applied to samples from any populations. ii. the correlation coefficient test. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 4(2):225–232, July 1937.

[PM18] Judea Pearl and Dana Mackenzie. *The book of why: The new science of cause and effect*. Basic books, 2018.

[PMP+23] Sachin Pawar, Ravina More, Girish K. Palshikar, Pushpak Bhattacharyya, and Vasudeva Varma. Knowledge-based extraction of cause–effect relations from biomedical text. *Lecture Notes in Electrical Engineering*, 964:157 – 173, 2023.

[PPO22] Jason Priem, Heather Piwowar, and Richard Orr. Openalex: A fully-open index of scholarly works, authors, venues, institutions, and concepts, 2022.

[PSM14] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

[RAJS22] Matthias Rüdiger, David Antons, Amol M. Joshi, and Torsten-Oliver Salge. Topic modeling revisited: New evidence on algorithm performance and quality metrics. *PLoS ONE*, 17(4), 2022.

[RBH15] Michael Röder, Andreas Both, and Alexander Hinneburg. Exploring the space of topic coherence measures. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, WSDM '15, page 399–408, New York, NY, USA, 2015. Association for Computing Machinery.

[RN18] Alec Radford and Karthik Narasimhan. Improving language understanding by generative pre-training. 2018.

[RNSS18] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. *OpenAI Blog*, 2018.

[RP16] Petar Ristoski and Heiko Paulheim. Rdf2vec: Rdf graph embeddings for data mining. In *International Workshop on the Semantic Web*, 2016.

[RR09] Lev Ratinov and Dan Roth. Design challenges and misconceptions in named entity recognition. In Suzanne Stevenson and Xavier Carreras, editors, *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL-2009)*, pages 147–155, Boulder, Colorado, June 2009. Association for Computational Linguistics.

[RS16] P S Raji and Subu Surendran. RDF approach on social network analysis. In *2016 International Conference on Research Advances in Integrated Navigation Systems (RAINS)*, pages 1–4, 2016.

[RSG17] Stephan Rabanser, Oleksandr Shchur, and Stephan Günnemann. Introduction to tensor decompositions and their applications in machine learning, 2017.

[Rub74] Donald Rubin. Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of Educational Psychology*, 66, 10 1974.

[RvEV+16] Marco Rospocher, Marieke van Erp, Piek Vossen, Antske Fokkens, Itziar Aldabe, German Rigau, Aitor Soroa, Thomas Ploeger, and Tessel Bogaard. Building event-centric knowledge graphs from news. *Web Semant.*, 37(C):132–151, March 2016.

[RWC+19] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, pages 1–24, 2019.

[SBA+22] R. Sandhiya, A.M. Boopika, M. Akshatha, S.V. Swetha, and N.M. Hariharan. A review of topic modeling and its application. In *Handbook of Intelligent Computing and Optimization for Sustainable Development*, page 305 – 322. 2022.

[SBL22] L. Siddharth, Lucienne Blessing, and Jianxi Luo. Natural language processing in-and-for design research. *Design Science*, 8:e21, 2022.

[SBWL21] L. Siddharth, Lucienne T. M. Blessing, Kristin L. Wood, and Jianxi Luo. Engineering Knowledge Graph From Patent Database. *Journal of Computing and Information Science in Engineering*, 22(2):021008, 10 2021.

[SCR+25] Vivek Sriram, Ashley Mae Conard, Ilyana Rosenberg, Dokyoon Kim, T. Scott Saponas, and Amanda K. Hall. Addressing biomedical data challenges and opportunities to inform a large-scale data lifecycle for enhanced data sharing, interoperability, analysis, and collaboration across stakeholders. *Scientific Reports*, 15(1), 2025.

[SK22] Shohei Shimizu and Shuichi Kawano. Special issue: Recent developments in causal inference and machine learning. *Behaviormetrika*, 49(2):275–276, 2022.

[SM12] Charles Sutton and Andrew McCallum. An introduction to conditional random fields. *Found. Trends Mach. Learn.*, 4(4):267–373, April 2012.

[SMV+21] Xinpeng Shen, Sisi Ma, Prashanthi Vemuri, M. Regina Castro, Pedro J. Caraballo, and Gyorgy J. Simon. A novel method for causal structure discovery from EHR data and its application to type-2 diabetes mellitus. *Scientific Reports*, 11(1), 2021.

[SOTM19] Angelo A Salatino, Francesco Osborne, Thiviyan Thanapalasingam, and Enrico Motta. The cso classifier: Ontology-driven detection of research

topics in scholarly articles. In *Digital Libraries for Open Knowledge: 23rd International Conference on Theory and Practice of Digital Libraries, TPDL 2019, Oslo, Norway, September 9-12, 2019, Proceedings 23*, pages 296–311. Springer, 2019.

[SP25] Shahidur Rahoman Sohag and Syed Murtoza Mushrul Pasha. Exploring causal relationships in biomedical literature: Methods and challenges. *International Journal of Innovative Science and Research Technology (IJISRT)*, 9(12), January 2025.

[SPT+12] Pontus Stenetorp, Sampo Pyysalo, Goran Topić, Tomoko Ohta, Sophia Ananiadou, and Jun'ichi Tsujii. brat: a web-based tool for NLP-assisted text annotation. In Frédérique Segond, editor, *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 102–107, Avignon, France, April 2012. Association for Computational Linguistics.

[SSS+15] Arnab Sinha, Zhihong Shen, Yang Song, Hao Ma, Darrin Eide, Bo-June (Paul) Hsu, and Kuansan Wang. An overview of microsoft academic service (mas) and applications. In *Proceedings of the 24th International Conference on World Wide Web*, WWW '15 Companion, page 243–246, New York, NY, USA, 2015. Association for Computing Machinery.

[SYC+24] Shichao Sun, Ruifeng Yuan, Ziqiang Cao, Wenjie Li, and Pengfei Liu. Prompt chaining or stepwise prompt? refinement in text summarization. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Findings of the Association for Computational Linguistics: ACL 2024*, pages 7551–7558, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

[Tea24] Gemma Team. Gemma, 2024.

[THH+23] Fiona Anting Tan, Hansi Hettiarachchi, Ali Hürriyetoğlu, Nelleke Oostdijk, Tommaso Caselli, Tadashi Nomoto, Onur Uca, Farhana Ferdousi Liza, and See-Kiong Ng. RECESS: Resource for extracting cause, effect, and signal spans. In Jong C. Park, Yuki Arase, Baotian Hu, Wei Lu, Derry Wijaya, Ayu Purwarianti, and Adila Alfa Krisnadhi, editors, *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 66–82, Nusa Dua, Bali, November 2023. Association for Computational Linguistics.

[TMH+24] Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*, 2024.

[TMS+23a] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[TMS+23b] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[TT15] Suppawong Tuarob and Conrad S. Tucker. Quantifying Product Favorability and Extracting Notable Product Features Using Large Scale Social Media Data. *Journal of Computing and Information Science in Engineering*, 15(3):031003, 09 2015.

[Via19] Gregory Vial. Understanding digital transformation: A review and a research agenda. *The Journal of Strategic Information Systems*, 28(2):118–144, 2019. SI: Review issue.

[VJR+23] Jack T. VanSchaik, Palak Jain, Anushri Rajapuri, Biju Cheriyan, Thankam P. Thyvalikakath, and Sunandan Chakraborty. Using transfer learning-based causality extraction to mine latent factors for Sjögren's syndrome from biomedical literature. *Heliyon*, 9(9):e19265, 2023.

[VSP+17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. advances in neural information processing systems. *Advances in neural information processing systems*, 30(2017), 2017.

[VSR+22] Josef Valvoda, Naomi Saphra, Jonathan Rawski, Adina Williams, and Ryan Cotterell. Benchmarking compositionality with formal languages. In Nicoletta Calzolari, Chu-Ren Huang, Hansaem Kim, James Pustejovsky, Leo Wanner, Key-Sun Choi, and et al., editors, *Proceedings of the 29th International Conference on Computational Linguistics*, pages 6007–6018, Gyeongju, Republic of Korea, October 2022. International Committee on Computational Linguistics.

[WCB+11] George Westerman, Claire Calméjane, Didier Bonnet, Patrick Ferraris, Andrew McAfee, et al. Digital transformation: A roadmap for billion-dollar organizations. *MIT Center for digital business and capgemini consulting*, 1(1-68), 2011.

[WDS+20] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, and et al. Transformers: State-of-the-art natural language processing. In Qun Liu and David Schlangen, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, October 2020. Association for Computational Linguistics.

[WJL+22] Isaac Ronald Ward, Jack Joyner, Casey Lickfold, Yulan Guo, and Mohammed Bennamoun. A practical tutorial on graph neural networks. *ACM Comput. Surv.*, 54(10s), September 2022.

[WMWG17] Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. Knowledge graph embedding: A survey of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering*, 29(12):2724–2743, 2017.

[WNS+11] Patricia L Whetzel, Natalya F Noy, Nigam H Shah, Paul R Alexander, Csongor Nyulas, Tania Tudorache, and Mark A Musen. Bioportal: enhanced functionality via new web services from the national center for biomedical ontology to access and use ontologies in software applications. *Nucleic acids research*, 39(suppl_2):W541–W545, 2011.

[Wol17] Phillip Wolff. Force dynamics. In *The Oxford Handbook of Causal Reasoning*, pages 147–168. Oxford University Press, 06 2017.

[WPL+16] Chih-Hsuan Wei, Yifan Peng, Robert Leaman, Allan Peter Davis, Carolyn J Mattingly, Jiao Li, Thomas C Wiegers, and Zhiyong Lu. Assessing the state of the art in biomedical relation extraction: overview of the biocreative v chemical-disease relation (cdr) task. *Database*, 2016:baw032, 2016.

[WWLH19] David Wadden, Ulme Wennberg, Yi Luan, and Hannaneh Hajishirzi. Entity, relation, and event extraction with contextualized span representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5784–5789, Hong Kong, China, November 2019. Association for Computational Linguistics.

[XXT23] Y. Xiao, C. Xiao, and M. Thürer. A patent recommendation method based on kg representation learning. *Engineering Applications of Artificial Intelligence*, 126, 2023.

[XYX+25] Jian Xu, Chao Yu, Jiawei Xu, Vetle I Torvik, Jaewoo Kang, Mujeen Sung, Min Song, Yi Bu, and Ying Ding. Pubmed knowledge graph 2.0: Connecting papers, patents, and clinical trials in biomedical science. *Scientific Data*, 12(1):1018, 2025.

[XZLZ20] Jinghang Xu, Wanli Zuo, Shining Liang, and Xianglin Zuo. A review of dataset and labeling methods for causality extraction. In *COLING 2020 - 28th International Conference on Computational Linguistics, Proceedings of the Conference*, page 1519 – 1531, 2020.

[YHP22a] Jie Yang, Soyeon Caren Han, and Josiah Poon. A survey on extraction of causal relations from natural language text. *Knowledge and Information Systems*, 64(5):1161–1186, 2022.

[YHP22b] Jie Yang, Soyeon Caren Han, and Josiah Poon. A survey on extraction of causal relations from natural language text. *Knowl. Inf. Syst.*, 64(5):1161–1186, May 2022.

[YPA+22] Altynay Yerkhassym, Alexandr A. Pak, Iskander Akhmetov, Amir Yelenov, and Alexander Gelbukh. On causality problem in natural language processing field. *Computacion y Sistemas*, 26(4):1549 – 1556, 2022.

[YYK+23] Yu Yu, Chao-Han Huck Yang, Jari Kolehmainen, Prashanth G. Shivakumar, Yile Gu, Sungho Ryu Roger Ren, Qi Luo, Aditya Gourav, I-Fan Chen, Yi-Chieh Liu, Tuan Dinh, Ankur Gandhe Denis Filimonov, Shalini Ghosh, Andreas Stolcke, Ariya Rastow, and Ivan Bulyko. Low-rank adaptation of large language model rescoring for parameter-efficient speech recognition. In *2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, page 1–8. IEEE, December 2023.

[YYL+19] Yuan Yao, Deming Ye, Peng Li, Xu Han, Yankai Lin, Zhenghao Liu, Zhiyuan Liu, Lixin Huang, Jie Zhou, and Maosong Sun. DocRED: A large-scale document-level relation extraction dataset. In Anna Korhonen, David Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 764–777, Florence, Italy, July 2019. Association for Computational Linguistics.

[ZDY+23] Xiaoyan Zhao, Yang Deng, Min Yang, Lingzhi Wang, Rui Zhang, Hong Cheng, Wai Lam, Ying Shen, and Ruifeng Xu. A comprehensive survey on deep learning for relation extraction: Recent advances and new frontiers, 2023.

[ZDY+24] Xiaoyan Zhao, Yang Deng, Min Yang, Lingzhi Wang, Rui Zhang, Hong Cheng, Wai Lam, Ying Shen, and Ruifeng Xu. A comprehensive survey on relation extraction: Recent advances and new frontiers, 2024.

[ZGSC24] Vanni Zavarella, Juan Carlos Gamero-Salinas, and Sergio Consoli. A few-shot approach for relation extraction domain adaptation using large language models. *arXiv preprint arXiv:2408.02377*, 2024.

[ZLL+14] Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao. Relation classification via convolutional deep neural network. In *Proceedings of COLING 2014, the 25th international conference on computational linguistics: technical papers*, pages 2335–2344, 2014.

[ZPL+24] Mingqian Zheng, Jiaxin Pei, Lajanugen Logeswaran, Moontae Lee, and David Jurgens. When" a helpful assistant" is not really helpful: Personas in system prompts do not improve performances of large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 15126–15154, 2024.

[ZZC+17] Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D. Manning. Position-aware attention and supervised data improve slot filling. In Martha Palmer, Rebecca Hwa, and Sebastian Riedel, editors, *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 35–45, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

A KNOWLEDGE GRAPH CONSTRUCTION METHODS

	Hyperparameter	Explanation	Value Range
UMAP	*n_neighbors*	the number of neighboring data points UMAP will look at when attempting to learn the manifold structure of the data	[5,15,30]
	min_dist	the minimum distance apart that points are allowed to be in the low dimensional representation	[0.0,0.1]
	n_components	the dimensionality of the reduced space	[2,5,10,30]
HDBSCAN	*min_cluster_size*	affects the final number of generated clusters	[5,10,20,50,100,200]
	min_samples	the minimum number of points required in the neighborhood of a point for it to be considered a core point	[None,1,2,5]
	metric	used to calculate the distances	["euclidean","cosine"]

Table 1: Main hyperparameters searched for upon optimizing the UMAP-HDBSCAN interaction.

B DT MONITORING

Code Repositories: `https://github.com/zavavan/dtm_kg`
Main Programming Languages: Python
External Libraries: SpaCy, Scikit-learn, NetworkX, hdbscan, umap, Stanford CoreNLP, NLTK, Pandas

---

Embedding Model	Silhouette · Clustered Ratio	Num Clusters
BERT	0.9387	1107
BERT	0.9287	918
BERT	0.9171	1063
Sentence-BERT	0.6852	869
Sentence-BERT	0.6794	978
Sentence-BERT	0.6767	1050
GloVe	0.6505	327
GloVe	0.6362	**332**
GloVe	0.6345	511

Table 2: The table presents clustering score values and the number of output clusters for the top three performing UMAP-HDBSCAN configurations across three tested embedding models. It's worth noting that the dataset comprises a total of 29,335 relation instances for contextualized BERT and Sentence-BERT embeddings. In contrast, for static GloVe embeddings, we consolidated single occurrences of each relation form, resulting in a final set of 2,539 relations due to their context-independent vector representations.

C AECO

Code Repository: `https://github.com/zavavan/AECO_KG_Pipeline`.
Main Programming Languages: Python
External Libraries: BERTopic; Hugging Face's Transformers, PEFT, Datasets, Evaluate; VOSViewer; SpaCy, Pandas, Brat

---

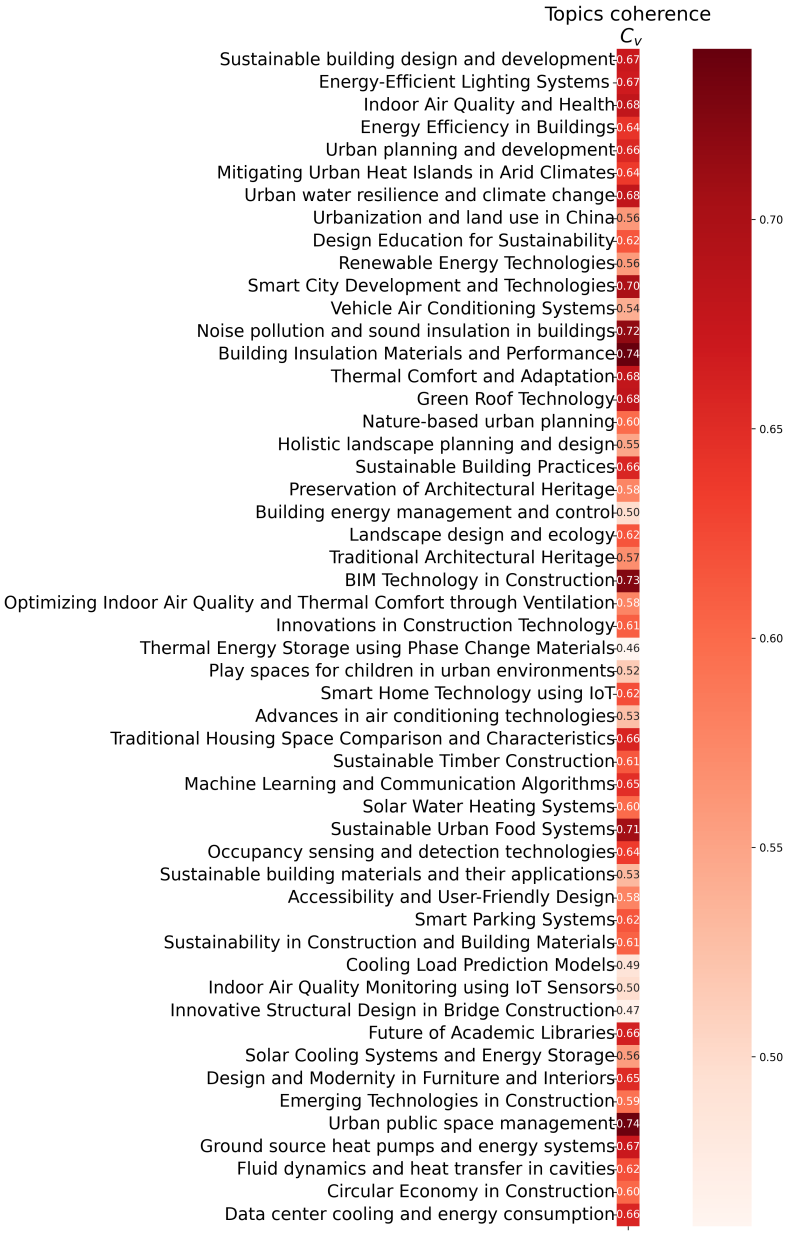


Figure 2: Topic Coherence score heat map for the optimized topic model.

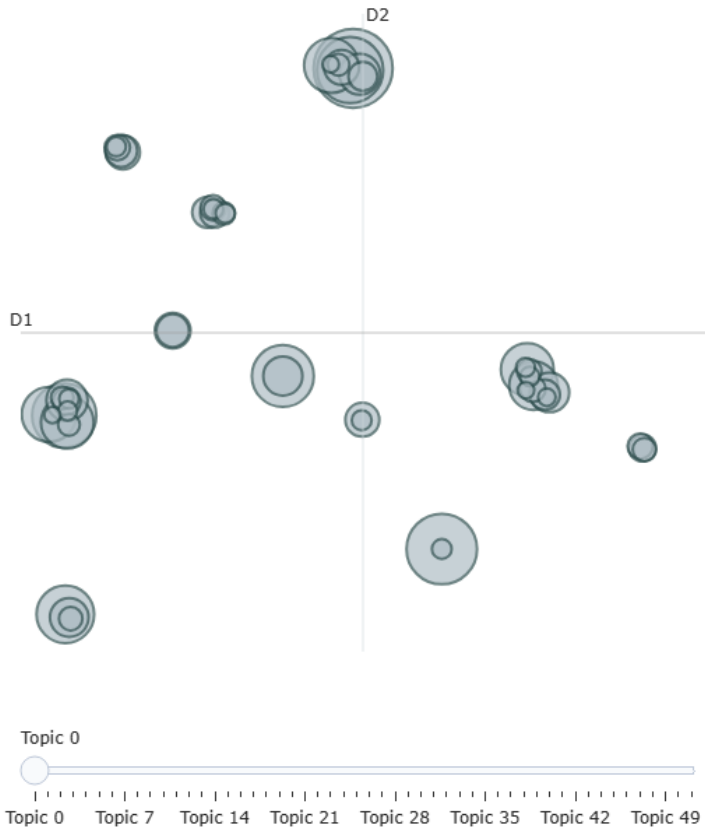


Figure 3: Optimized topics displayed in a reduced 2-dimensional embedding space, showing inter-topic distances.

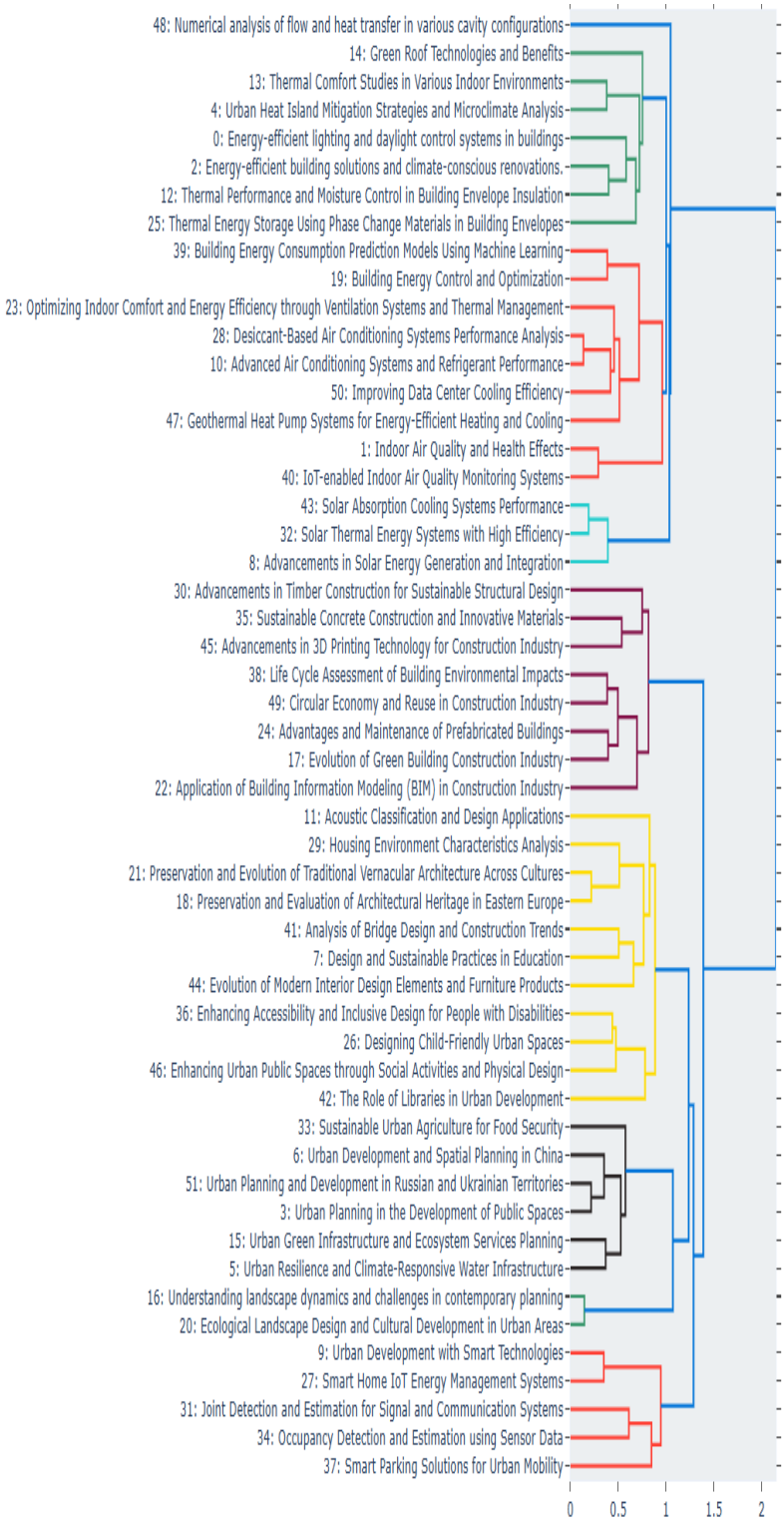


Figure 4: Dendrogram representation of the optimized topics' hierarchical clustering. The leaves of the tree represent the 52 clusters, the intermediate nodes represent merged clusters, and the height of the merging (distance from the leaves) indicate topic similarity as based on the cosine distance matrix between topic embeddings.

Table 3: The consolidated set of macro-topics resulting from topic merging, with their document counts, LLM-generated descriptions, and term based representation.

Topic	Count	Representation	Terms
0	39619	Energy Efficiency and Thermal Comfort in Building Environments	['energy', 'thermal', 'lighting', 'building', 'comfort', 'buildings', 'temperature', 'heat', 'study', 'performance']
1	22740	Indoor Air Quality and Sustainable Air Conditioning Systems	['air', 'indoor', 'ventilation', 'conditioning', 'cooling', 'quality', 'temperature', 'energy', 'concentrations', 'control']
2	14635	Urban Development Strategies and Sustainable City Planning	['urban', 'land', 'planning', 'development', 'city', 'spatial', 'cities', 'growth', 'public', 'expansion']
3	10650	Enhancing Child-Friendly Urban Spaces Through Design	['design', 'space', 'product', 'architectural', 'public', 'architecture', 'students', 'library', 'research', 'study']
4	10244	Smart city development and urban data management	['smart', 'city', 'parking', 'cities', 'data', 'occupancy', 'urban', 'information', 'development', 'based']
5	9518	Urban Resilience and Green Infrastructure in Climate Change Planning	['urban', 'resilience', 'infrastructure', 'water', 'climate', 'green', 'planning', 'cities', 'flood', 'change']
6	7452	Architectural Integration of Solar Photovoltaic Systems in Buildings	['solar', 'pv', 'energy', 'photovoltaic', 'bipv', 'power', 'systems', 'electricity', 'renewable', 'building']
7	5632	Preservation and Evolution of Traditional Architecture in Modern Contexts	['architectural', 'architecture', 'heritage', 'traditional', 'house', 'historical', 'houses', 'buildings', 'study', 'cultural']
8	5600	Sustainable Building Construction and Design with Environmental Assessment	['building', 'green', 'construction', 'buildings', 'assessment', 'life', 'environmental', 'cycle', 'industry', 'design']
9	4619	Landscape Planning and Design Theory	['landscape', 'landscapes', 'design', 'garden', 'architecture', 'cultural', 'ecological', 'planning', 'rural', 'research']
10	3368	Urban Sound Environment Research in Architectural Design	['noise', 'sound', 'acoustic', 'design', 'insulation', 'floor', 'building', 'level', 'environment', 'study']
11	3039	Sustainable Construction Materials and Technologies	['concrete', 'timber', 'construction', 'wood', '3d', 'materials', 'material', 'structural', 'structures', 'building']
12	1901	Utilizing BIM in Construction and Building Information Modeling Industry	['bim', 'construction', 'information', 'building', 'technology', 'design', 'industry', 'modeling', 'project', 'management']
13	1090	Urban Agriculture and Sustainable Food Systems	['food', 'urban', 'city', 'cities', 'planning', 'land', 'production', 'community', 'social', 'development']
14	767	Sustainable Bridge Design and Construction	['lt', 'gt', 'design', 'structural', 'construction', 'structures', 'structure', 'new', 'river', 'engineering']
15	615	Investigation of Cavity Dynamics and Heat Transfer in Various Flow Scenarios	['cavity', 'flow', 'wall', 'transfer', 'heat', 'number', 'pressure', 'walls', 'numerical', 'surface']

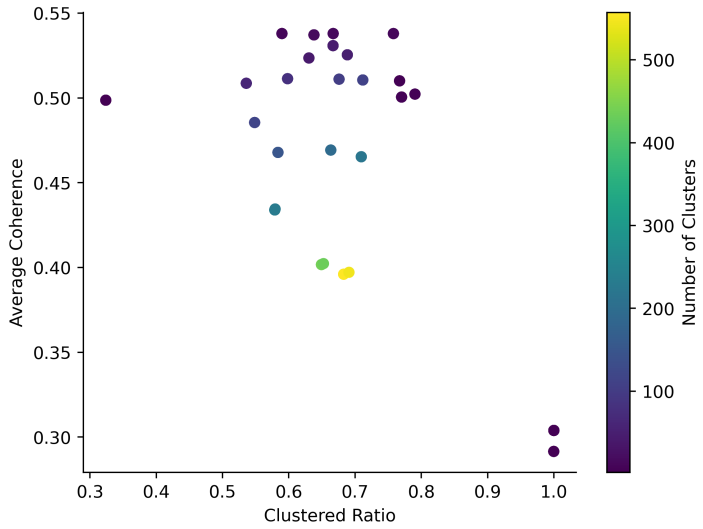


Figure 5: Average topic coherence values against the ratio of clustered datapoints for a subset of HDBSCAN hyperparameter settings. The color-coded values of the number of resulting clusters are also shown.

D CAUSALITY GRAPHS

Code Repositories: https://github.com/zavavan/CRE_LLM_Benchmark
Main Programming Languages: Python
External Libraries: Hugging Face's Transformers, PEFT, Datasets, Evaluate; SpaCy; Scikit-learn; Pandas

The full list of Prompt Variations of the Prompt Sensitivity Analysis is available at the URL: https://drive.google.com/file/d/1mGawv1h-o-0xT6DNca4IAzVavmgrq8Ps/view?usp=sharing

---

Tables 4 and 5 provide the hyperparameters used across all zero-/few-shot and all instruction fine-tuning experiments, respectively. For any other unspecified parameter, we retained the default values provided by Hugging Face's classes.

We train and run model inferences on Google Colab Pro using a single NVIDIA A100-SXM4 GPU with 80GB VRAM running Ubuntu 22.04.4 LTS with CUDA 12.4, with the exception of prompt-based methods using the *DeepSeek-Qwen-Distill* model, which were run through the HF Inference API (https://huggingface.co/docs/inference-providers/en/providers/hf-inference).

The software environment used throughout this study included the following library versions: Python 3.12.12, PyTorch 2.8.0, Transformers 4.57.1, PEFT 0.17.1, Accelerate 1.11.0, and BitsAndBytes 0.48.1.

The released artifacts for the models fine-tuned using the PEFT LoRA framework contain only the LoRA adapter weight matrices and configuration files (stored in the files adapter_model.bin and adapter_config.json, respectively). The base model weights are not redistributed; instead, users can reproduce the complete model by loading the corresponding base checkpoint from the Hugging Face Hub and attaching the adapter, using the code in Figure 6:

```
from transformers import AutoModelForCausalLM, AutoTokenizer
from peft import PeftModel
base_model_id = "johnsnowlabs/JSL-MedLlama-3-8B-v2.0"
adapter_path = "/path-to-fine-tuned-model"
bnb_config = BitsAndBytesConfig(
    load_in_4bit=True,
    bnb_4bit_quant_type="nf4",
    bnb_4bit_use_double_quant=True)
base_model = AutoModelForCausalLM.from_pretrained(
    base_model_id,
    quantization_config=bnb_config)
tokenizer = AutoTokenizer.from_pretrained(base_model_id)
model = PeftModel.from_pretrained(base_model, adapter_path)
```

Figure 6: Code snippet for merging the released LoRA adapter with the MedLlama base model.

Table 4: Model instantiation and inference parameters used across all zero-shot and few-shot experiments. *max_new_tokens* parameter for two-step prompt methods *SumAsk* and *2-Chain* is specified as a pair of values, one for each inference call. *max_new_tokens* is raised to 900 for inference with *DeepSeek-Qwen-Distill* in all prompting methods, in order to accommodate for the long reasoning chains of this model.

Parameter	**HF parameter**	**Prompt**	**Value**
Use sampling	`do_sample`	all	False
4 bit quantization	`load_in_4bit`	all	True
quantization scheme	`bnb_4bit_quant_type`	all	nf4
maximum num of new tokens	max_new_tokens	InstPrompt	256
		iCL	256
		CoT	512
		SumAsk	256,64
		2-Chain	256,64

Table 5: LoRA configuration and training parameters used across all fine-tuning experiments.

Parameter	**HF parameter**	**Value**
LoRA rank	`r`	8
LoRA alpha	`lora_alpha`	32
LoRA dropout	`lora_dropout`	0.05
LoRA target modules	`target_modules`	["q_proj", "v_proj"]
Validation Metric	`metric_for_best_model`	eval_loss
Train batch size	`per_device_train_batch_size`	8
Validation batch size	`per_device_eval_batch_size`	8
Gradient Accumulation Steps	`gradient_accumulation_steps`	4
Evaluation Strategy	`evaluation_strategy`	steps
Evaluation Steps	`eval_steps`	10
Early Stopping Patience	`early_stopping_patience`	3
Learning Rate	`learning_rate`	2e-4
Optimizer	`optim`	paged_adamw_8bit
16-bit Precision Training	`fp16`	True

Input

```
Text:Bacterial Meningitis/ Fever :Again, the patient presented
with obtundation and an inflammatory CSF after URI symptoms.She
was admitted to the[**Hospital Unit Name 153**] for a short course
and improved dramatically on Ceftriaxone, Vancomycin and Acyclovir.
Entities: E1:Fever , E2:Acyclovir
Gold Standard Label: 7
```

Model Response MistralOrca

```
The relation holding between Fever and Acyclovir is that Acyclovir
is a medication used to treat Fever.The closest relation that best
describes it is: Acyclovir prevents Fever.The label associated to
this closest relation is 5. The answer is: LABEL 5.
```

Model Response DeepSeek-Qwen-Distill

```
Response CoT: Okay, so I need to figure out the relationship between
`Fever' and `Acyclovir' in the given text. Let me read the text
carefully.The text says: [...] So, the patient had fever and other
symptoms, and was treated with three medications: Ceftriaxone,
Vancomycin, and Acyclovir. The improvement was dramatic. Now, the
entities are `Fever' and `Acyclovir'. I need to see how they relate.
Looking at the options: 0: Fever causes Acyclovir. That doesn't make
sense because Acyclovir is a medication, not a symptom or condition.
[...] 5: Acyclovir prevents Fever. That could be possible. If the
fever is a symptom of the underlying infection, then treating with
Acyclovir might prevent further fever episodes or reduce fever.
[...] 7: Acyclovir hinders Fever. This is similar to option 5. If
Acyclovir is treating the cause of the fever, then it would hinder or
reduce the fever. [...]. So, the patient had fever, and was treated
with Acyclovir, which likely helped reduce the fever. So, Acyclovir
hinders Fever. That would be option 7.Wait, but another way to look
at it: the fever was a symptom, and the treatment (Acyclovir) was
given, which helped the patient improve. So, the fever was a result
of the infection, and the medication helped, so the fever went down.
So, Acyclovir hinders Fever. Alternatively, maybe the fever was a
symptom that indicated the need for Acyclovir, but that's more about
enabling, which is option 2. But I'm not sure if that's the correct
interpretation. Wait, the question is about the relation between
Fever and Acyclovir. So, Fever is a symptom, and Acyclovir is a
treatment. So, does Acyclovir hinder Fever? Yes, because it's used
to treat the underlying cause, which would reduce the fever.So, the
correct label would be 7.</think>The patient had fever as a symptom
and was treated with Acyclovir, which likely helped reduce the fever.
Therefore, Acyclovir hinders Fever.LABEL: 7
```

Figure 7: Comparing *Mistral Orca* and *DeepSeek-Qwen-Distill* model responses for a sample zero-shot CoT prompt. The target label to be extracted is 7 (*E2 hinders E1*). Notice as *DeepSeek-Qwen-Distill* evaluates, among others (not shown for simplicity), Label 5 option, but eventually opts for Label 7.

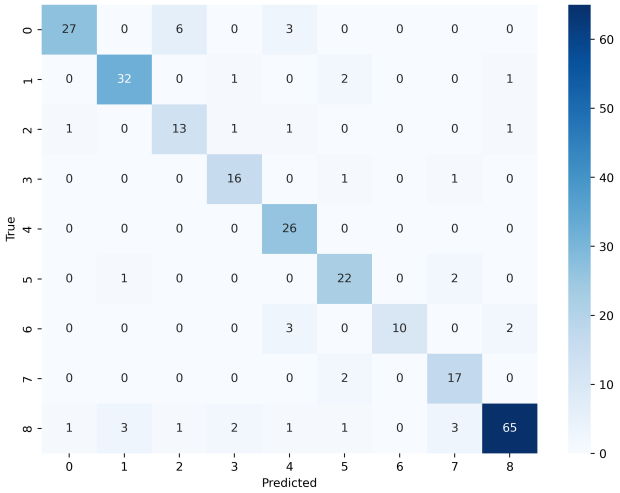


Figure 8: Confusion matrix across the nine causal relation classes for the best-performing model, the instruction fine-tuned *MedLlama*.