LLM-Assisted Multilingual Information Extraction for Knowledge Graph Population: The DBpedia Hindi Chapter

Sanju Tiwari[1,2,*,†], Debarghya Datta[4], Aditya Venkatesh Marada[3], Ronak Panchal[5], Ananya[6], Ronit Banerjee[7] and Tommaso Soru[8]

[1]DBpedia, Leipzig, Germany
[2]Sharda University, Delhi-NCR, India
[4]Microsoft Research, Bengaluru, India
[3]University of Amsterdam, The Netherlands
[5]Cognizant, Pune, India
[6]Stanford University, California, USA
[7]Keysight Technologies, Kolkata, India
[8]Liber AI Research London, United Kingdom

Abstract

DBpedia chapters are the community-driven initiative under DBpedia project to extract and enhance knowledge from various language editions of wikipedia. This paper presents the development of the DBpedia Hindi Chapter project in two GSoC editions, which were focused to develop a scalable framework to extract information in different languages and design a knowledge graph from Hindi Wikipedia. In 2024, we extended the DBpedia Information Extraction Framework to handle Devanagari-script data, added mappings for new infobox templates, and address the challenges posed by limited resources and natural language complexities. In 2025, we evaluated small language models (SLMs) for Hindi Open Information Extraction (OIE), built a hybrid pipeline combining rule-based and neural extraction, and developed a predicate linking module to map extracted relations to DBpedia ontology properties. The resulting open-source tool provides a blueprint to extend KG building to more under represented languages and hence facilitate digital equity and inclusiveness within the semantic web paradigm.

Keywords

Knowledge Graphs, Semantic Web, Information Extraction, Language Models, DBpedia

1. Introduction

Knowledge graphs are the basis of information systems in the present day, however, the linguistic diversity on which it is built is limited. Indic languages [1, 2] like Hindi which is spoken by over 600 million people are untapped in their potential for extracting structured knowledge from text. Multilingual knowledge extraction [3] is vital in developing knowledge systems that are comprehensive, all-encompassing, and universally useful. Most of the information in the world is written and read in languages other than English, and hence, there is a language bias and incompletion of information when only using monolingual sources. Multilingual knowledge extraction [4] enables structured information to be derived from diverse language sources, ensuring broader knowledge coverage and cultural representation.

DBpedia [5, 6] is a widely used knowledge graph that structures information extracted from Wikipedia and supports numerous semantic web and AI applications. While early DBpedia efforts focused mainly on the English Wikipedia, the rapid growth of Wikipedia in hundreds of languages highlights the

ESWC'26: 5th Workshop on LLM-Integrated Knowledge Graph Generation from Text (TEXT2KG), May 10-14, 2026 | Dubrovnik, Croatia

*Corresponding author.
tiwarisanju18@ieee.org (S. Tiwari); debarghya.datta89@gmail.com (D. Datta); adit.marada@student.uva.nl (A. V. Marada); ronakvtcbb@gmail.com (R. Panchal); anhooda@stanford.edu (Ananya); ronitbanerjee03@gmail.com (R. Banerjee); tom@liberai.org (T. Soru)
0000-0001-7197-0766 (S. Tiwari); 0009-0001-2554-6152 (D. Datta); 0009-0003-9003-3964 (A. V. Marada); 0000-0002-9260-9686 (R. Panchal); 0009-0002-2431-2511 (Ananya); 0009-0003-4939-4362 (R. Banerjee); 0000-0002-1276-2366 (T. Soru)

importance of multilingual knowledge extraction. Early editions of the DBpedia Information Extraction Framework (DIEF) focused on the English Wikipedia, resulting the loss of language-specific information while integrating the data. Greek DBpedia [7] has been proposed as a first language-specific DBpedia edition to generate, maintain the emerging Greek Linked Data Cloud.

Incorporating multiple language editions enables DBpedia to capture culturally diverse, region-specific, and domain-specific knowledge that is often missing or under-represented in English sources. Multilingual knowledge extraction significantly improves the completeness and coverage of DBpedia by reducing language bias and enriching entity representations. It also helps bridge the digital divide since most people are only interested in information in their native language. DBpedia provides backing to semantic apps in native languages and thus makes information easily accessible to those with limited language resources. Multilingual knowledge extraction using DBpedia chapters increases the scope of coverage of the knowledge graph, which becomes more extensive, diverse, and comprehensive. The local Wikipedia sites provide more in-depth information on regional entities, personalities, schools and organizations, places, and traditions. The extraction and publication of this data in multilingual DBpedia chapters can help remove language bias and provide a more equitable and global view of things.

Furthermore, multilingual DBpedia facilitates cross-lingual interoperability by aligning equivalent entities across languages, which is essential for multilingual question answering and cross-language information retrieval. Extracting knowledge from multiple languages also enhances data quality, as facts can be cross-validated across sources. Overall, multilingual knowledge extraction is crucial for building a globally inclusive, reliable, and semantically rich knowledge graph. Semantic Web and Natural Language Processing techniques are playing a significant role in knowledge extraction from multilingual text for question answering [3]. Currently, several advanced approaches are concentrating to construct knowledge graphs from text. Singh et. al. [2] proposed the task of Cross Lingual Fact Extraction(CLFE) from low resource Indian Languages text.

In this paper we have proposed a new Chapter for Hindi Language in DBpedia as an outcome of GSoC 2024-2025 project. Section 2 has discussed the related work about the existing chapters of DBpedia community. Section 3 aimed to build the foundational architecture of Hindi Chapter. This section focuses on the creation and evaluation of Hindi language mappings, along with the configuration of extractors for the Hindi DBpedia edition. It also covers the finalization and improvement of the extraction process, followed by the introduction of neural extraction methods. Additionally, progress includes refining coreference resolution and entity linking, leveraging tools such as Stanza and IndicNER for improved Hindi named entity recognition, and exploring techniques to extract structured knowledge from unstructured text. Section 4 focuses on setting up and streamlining the existing extraction pipeline and evaluating large language models (LLMs) for Hindi relation extraction. The work was further extended to enhance extraction capabilities through neural methods, followed by link prediction and improvements to the IndIE extraction approach for the Hindi DBpedia knowledge graph. Additional efforts included synthetic data generation, performance evaluation, and demonstration of a SPARQL endpoint for the Hindi chapter. Finally, predicate linking was proposed to improve the semantic quality and ontology alignment of the extracted triples.

2. Related Work

DBpedia [5] includes about 20 language chapters dedicated to the extraction and enhancement of structured data from language-specific Wikipedia editions. These chapters are coordinated by the DBpedia community and playing a significant role to advancing the DBpedia infrastructure. The primary objective of the DBpedia project [5] is to extract structured and multilingual information from Wikipedia and publish it as openly accessible knowledge on the Web using Semantic Web and Linked Data technologies. DBpedia in Dutch represents the first initiative that formally established an official DBpedia Chapter Consortium.

The DBpedia community is organized into multiple DBpedia Chapters, each serving a specific purpose.

Language chapters are responsible for maintaining DBpedia datasets in particular languages (such as English as the core language, German, Dutch, French and Czech), while regional chapters focus on collecting and managing knowledge related to specific regions or cities [6].

The QAKiS (Question Answering wiKiFramework-based System)[8] is a relation-driven question answering system that retrieves its answers from structured knowledge sources like DBpedia with the help of linguistic patterns learned from unstructured sources like Wikipedia. Its main aim is to "align" the English and French chapters of DBpedia in order to support cross-lingual and consistent question answering. It works by interpreting questions in a human language and matching them with relation patterns stored in the WikiFramework repository and then translates them into SPARQL queries. Its architecture is modular and consists of a query generator, pattern matcher, SPARQL query handler, and named entity recognizer that map questions to ontology relations and fetch answers accordingly.

A tutorial [6] shows how DBpedia has grown into a powerful and flexible knowledge graph platform with support for multilingual knowledge representation and how it has enabled the "automated extraction and publication of data" with its rich set of services like Databus, Spotlight, and Archivo that help build and use Linked Open Data in an efficient manner.

3. Building the Foundational Architecture of Hindi Chapter in GSoC 2024

In this part, we outline our approach for enhancing the Hindi Knowledge Graph through the integration of conventional structured extraction and neural techniques. The entity coverage with the current DBpedia Information Extraction Framework (DIEF) stands at 50%, prompting us to enhance the extractors and mappings tailored for Hindi. This required improving the interpretation of indic data within the central ontology and revising the ontology mappings to more effectively accommodate the distinct formatting of the Devanagari script. By enhancing the range of supported infoboxes and refining these data extractors, we boosted the quantity of Hindi entities included in the graph by 10%. These enhancements were incorporated into the official DBpedia release and are now available through a public SPARQL endpoint for public access. To enhance the framework, we have additionally introduced a neural extraction system designed to retrieve data from unstructured, raw text. This integrated approach not only strengthens the graph but also supports better search tools and language processing applications, ultimately enhancing the availability of Hindi digital content for educational and cultural purposes.

3.1. DBpedia Information Extraction

The DBpedia Information Extraction Framework (DIEF) acts as the main tool for transforming Wikipedia's semi-structured information from different lanuage into DBpedia KG. The system begins by handling Wikipedia XML data dumps with standardized mappings provided by maintainers that match source entities to the DBpedia ontology, enhanced by using a set of extractors that can interpret different data types (e.g., temporal, numerical) into RDF triples. Although the DBpedia community has traditionally provided significant support for various major global languages such as English, French, Japanese, and Greek, the Hindi language faced a serious lack of coverage in the past. The lack of representation hindered the ability to conduct important searches in the native Devanagari script. To resolve this inconsistency, targeted measures were executed throughout the GSoC 2024 and GSoC 2025 initiatives. These efforts successfully advanced the mappings for Hindi, greatly enhancing entity coverage to about 60% (as of March 2026).

3.2. Impact of Hindi DBpedia

The creation of an extensive Hindi Knowledge Graph (KG) enables a significant transition towards semantic interoperability in Indic computational linguistics. The KG facilitates machines to execute intricate relational reasoning over entities in the Devanagari script by organizing unstructured data into

a formal Resource Description Framework (RDF). This organized framework is crucial for Knowledge-Based Information Retrieval (KBIR) and detailed Named Entity Recognition (NER), advancing from keyword-centric matching to context-sensitive semantic search [5]. Additionally, by offering a native-script database of organized information, we reduce the ”translation loss” typical in cross-lingual processes, thus improving the efficiency of subsequent NLP tasks in specialized areas such as healthcare and governance.

3.3. Enhancing Hindi Coverage in DBpedia

To address the gap in the initial Hindi DBpedia release, we performed a systematic optimization of the ontological mappings and extraction framework. Originally, the Hindi chapter supported a limited set of mappings, specifically restricted to two templates: **'ज्ञानसन्दूक भारत के षेत्र'** (*Infobox Regions in India*) and **'ज्ञानसन्दूक फ़िल्म'** (*Infobox Film*). We expanded the support to 22 distinct infobox templates, significantly broadening the entity coverage as reported above, across the Hindi Wikipedia corpus. Moreover, we refined the extraction heuristics to account for the morphological and syntax-level nuances of the Hindi language, ensuring the generation of high-fidelity RDF triples. The primary updates were implemented in the following extraction components:

- **Disambiguation Extractor**
- **Homepage Extractor**
- **Abstract Extractor**
- **Date-interval Extractor**
- **Date-time Parser**

These enhancements are fundamental for the effective integration of structured data written in the Devanagari script into the DBpedia Knowledge Graph.

3.4. Release and Accessibility of the Hindi DBpedia

The Hindi Knowledge Graph (KG) has been effectively incorporated into the standardized DBpedia release framework, which depends on the **DBpedia Databus** for overseeing the extraction process and distributing versioned data [6]. The obtained triples were imported into a **Virtuoso triple store** in accordance with the established pipeline. This deployment approach guarantees that the Hindi KG is readily available to researchers and end-users. Access to query the knowledge graph will be available via a publicly accessible **SPARQL endpoint**, found at https://hi.dbpedia.org/sparql.

3.5. Neural Pipeline for Knowledge Graph Enrichment

While Wikipedia infoboxes provide high-precision structured data, they typically suffer from low recall, omitting significant relational data embedded in unstructured article text. To augment the Hindi KG, we developed a neural pipeline for Open Information Extraction (OpenIE) and Relation Extraction (RE) tailored to the linguistic constraints of the Devanagari script. This pipeline transforms unstructured Hindi corpora into formal RDF triples through the modular architecture shown in Figure 1:

1. **Source Preparation:** The process starts with gathering articles from Hindi Wikipedia dumps, which are then transformed into a set of plain text files to ease further linguistic processing.
2. **Syntactic and Entity Identification:** Tokenization and **Part-of-Speech (POS) tagging** are executed on every document with the aid of the **Stanza library** [9]. The tokens are subsequently sent to the **IndicNER** model for the detection and classification of named entity references(e.g., ORG, PER, LOC) found in the text.
3. **Textual Coherence:** Mentions are detected using a rule-based method that combines POS tags with token-level patterns. To maintain entity consistency across the document, we apply the TransMuCoRes model [10] for coreference resolution, which links different textual references that refer to the same real-world entity.

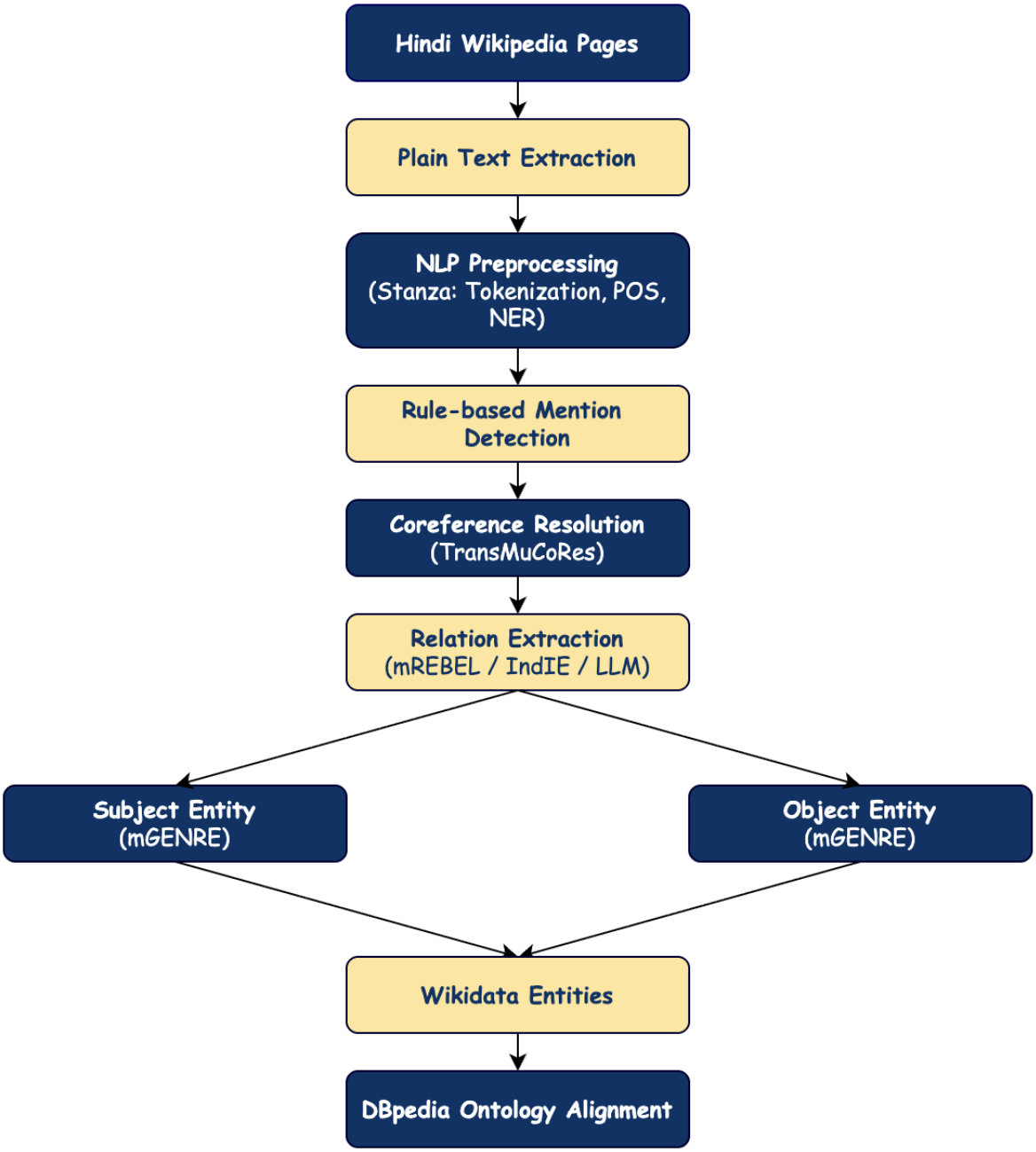


Figure 1: Pipeline efficiently converts unstructured text from Hindi Wikipedia into structured knowledge

4. **Relational Extraction:** Following coreference resolution, the sentences obtained are utilized to derive structured triples in the format (Subject, Relation, Object). This phase utilizes cutting-edge relation extraction models, such as **mREBEL** [11], **IndIE** [1], along with tailored **LLM-enhanced extraction models**.
5. **Ontological Alignment:** In the concluding phase, the identified entities and relationships are matched with current knowledge graph schemas to guarantee compatibility. Subjects and objects are connected to their standard Wikidata identifiers through the **mGENRE** model [12] for **Entity Linking**. The relationships obtained are then aligned with the hierarchical framework of the DBpedia ontology, allowing for smooth incorporation of the newly created triples into the knowledge graph.

4. Evaluation and Enhancement Strategies

The focus in 2025 was evaluating what current SLMs can actually do on Hindi OIE, and developing a pipeline for enriching the knowledge graph from Hindi text.

4.1. Evaluating SLMs for Hindi Open Information Extraction

We systematically evaluated multiple open source models like Mistral, Llama3.2, and the Gemma 3 series on the Hindi-BenchIE[1] dataset to get a baseline for SLM performance on this task. We selected these models, as they were the state-of-the-art locally-deployable models at the time of evaluation.

We tested six prompting strategies, from zero-shot to few-shot examples to structured reasoning chains. As per our experiments, the most effective strategies were:

- **Chain-of-Thought with Evidence Reasoning (CoT-ER):** The model was required to identify the subject, relation, and object, and to cite exact text from the source sentence as evidence for each before producing the final triple which led to a significant reduction in hallucinations. [13]
- **ReAct (Reason and Act):** Building on CoT-ER, and borrowing from [14], the model first worked through the extraction as a reasoning step, then produced a structured JSON output with the final triples.

Gemma-3-12B was the strongest model. With the ReAct strategy it reached 34.4% F1, compared to 33.7% for CoT-ER alone (Table 1). It also outperformed GPT-4.1-nano, which scored 29.2% F1 on the same benchmark.

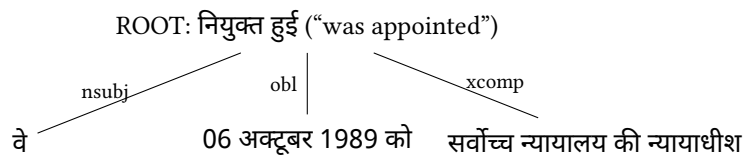
Table 1
Performance of Gemma-3-12B on Hindi-BenchIE with Advanced Prompting

Prompting Strategy	Precision	Recall	F1-Score
Chain-of-Thought ER	36.3%	31.4%	33.7%
ReAct + CoT-ER	**41.1%**	29.6%	**34.4%**

4.2. A Hybrid Approach: Augmenting Rule-Based Systems

SLM-only performance (34.4% F1) falls well short of the existing state-of-the-art **IndIE**, which scores 51% F1. IndIE uses a 3-stage pipeline which consists of chunking, merged dependency tree (MDT) creation, and a rule based complex engine of over 100 handwritten rules for extraction.

To illustrate how IndIE extracts triples from text and what we do differently consider this sentence from the gold annotated dataset: "06 अक्टूबर 1989 को वे सर्वोच्च न्यायालय की न्यायाधीश नियुक्त हुई" ("On 6 October 1989, she was appointed judge of the Supreme Court"). In Stage 1, the chunker groups the tokens into phrases: '06 अक्टूबर 1989 को' ("on 6 October 1989"), 'वे' ("she"), 'सर्वोच्च न्यायालय की' ("of the Supreme Court"), 'न्यायाधीश' ("judge"), and the auxiliary 'हुई' ("was") with the main verb 'नियुक्त' ("appointed"). In Stage 2, 'हुई' is merged into 'नियुक्त' via the aux rule, and the resulting VP becomes the root of the MDT:



In Stage 3, the rules walk the tree in dependency-priority order (`subj` → `obj` → `obl` → `nmod`) and emit three triples:

(a) (वे, नियुक्त हुई, न्यायाधीश) — "(she, was appointed, judge)". The `nsubj` child is taken as the subject and the head of the `xcomp` subtree as the object.
(b) (06 अक्टूबर 1989 को, नियुक्त हुई, वे) — "(on 6 October 1989, was appointed, she)". A second pass lifts the `obl` child into the subject slot to surface the temporal argument.
(c) (न्यायाधीश, property, सर्वोच्च न्यायालय की) — a property triple emitted by the augmentation step for the court modifier.

Here triples (a) and (b) match the gold annotation. Triple (c), however, is emitted in the wrong direction: the gold annotation expects (सर्वोच्च न्यायालय की, property, न्यायाधीश) (judge of the supreme court), but the rule that fires on `xcomp`-internal modifiers inverts subject and object and produces the reversed triple. More importantly, IndIE never emits the semantically obvious relation (वे, नियुक्त हुई, सर्वोच्च न्यायालय की) ("she was appointed to the Supreme Court"), because no single MDT edge connects 'वे' to 'सर्वोच्च न्यायालय की'—the `xcomp` node sits between them, so the rules cannot bridge the two phrases. This is a repeated pattern we observe where the rules work well on canonical word order and explicit dependency markers, but fail on implicit relations that do not surface as a single edge.

To combine the recall of neural models with the precision of structured rule based systems, we replaced IndIE's rule engine with our best SLM, feeding it IndIE's intermediate dependency tree and asking it to generate the final triples directly using this rich information dense data structure.

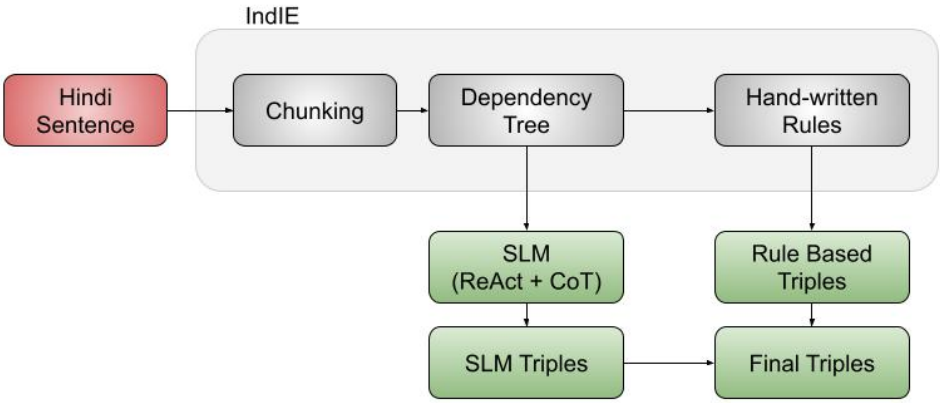


Figure 2: Comparison of the original rule-based IndIE pipeline and our hybrid SLM-augmented approach. Both pipelines share the initial Chunking and Merged Dependency Tree (MDT) creation stages. Our contribution replaces the brittle, hand-written rule engine with a flexible SLM, increasing recall.

On the same sentence, the SLM reads the MDT and the three IndIE triples, and generates five additional triples. The most useful one is exactly the relation IndIE could not reach: (वे, नियुक्त हुई, सर्वोच्च न्यायालय की) ("she, was appointed, to the Supreme Court"). The model also emits the chained form (06 अक्टूबर 1989 को, नियुक्त हुई वे, सर्वोच्च न्यायालय की) ("on 6 October 1989, she was appointed, to

the Supreme Court"), which ties the date, subject, verb and court into one structured tuple. Together these two additions recover the gold triples that the rule engine missed on this sentence.

The trade-off is that the SLM also produces noise. On the same sentence it emits (06 अक्टूबर 1989 को, हुआ, नियुक्त हुई) ("on 6 October 1989, happened, was appointed")—a content-free "happened" predicate—and two redundant permutations that recycle the same chunks into lower-quality tuples. This pattern of genuine recall gains mixed with low-precision noise is consistent across the dataset.

Overall, the hybrid approach led to a substantial improvement in recall (66%) over both the standalone SLM and the original IndIE system which shows the model's ability to generalize beyond the brittle handwritten rules and will only get better with model capabilities. However, we do note that the gain in recall is accompanied by a drop in precision, as the model introduces more noise. While we tried adding a second SLM as a filtering layer, it did not help as SLMs appear to be better at generating candidates than rejecting them.

Table 2
Comparison of Hindi OIE Approaches

Method	Precision	Recall	F1-Score
IndIE	49.0%	53.0%	51.0%
SLM Only (Gemma-3 + ReAct)	41.1%	29.6%	34.4%
Hybrid (IndIE + SLM)	26.7%	**66.0%**	38.0%

4.3. Predicate Linking to the DBpedia Ontology

Triples extracted from text use natural language predicates and are not yet mapped to a formal ontology. To integrate them into DBpedia, we built a module that maps a relation like 'के पिता हैं' to `dbo:parent`.

The linker uses three signals. First, **lexical similarity**: the relation is translated to English and fuzzy-matched against DBpedia property labels and descriptions. Second, **embedding similarity**: cosine distance between relation and property embeddings catches matches that surface-form comparison misses. Third, **graph-context validation**: candidate properties are checked against the domain and range constraints implied by the linked subject and object, filtering out type-inconsistent mappings.

Figure 3: Predicate linking pipeline.

The predicate linker is itself a neurosymbolic module as it combines symbolic graph-context constraints with embedding similarty making sure that linked predicates are both semantically and ontologically consistent.

4.4. Link Prediction as a Diagnostic

To understand whether knowledge graph completion could address the sparsity of the Hindi DBpedia infobox-derived graph, we evaluated link prediction on the derived graph. We treat this experiment like a diagnostic that clarifies the conditions required before completion methods can be effectively implemented.

The initial extracted graph contained a total 2,039,012 triples. We then trim the graph by restricting to DBpedia ontology relations removing disambiguation style predicates. This led to cutting the graph down to 544,161 triples over 304,395 unique entities. However, this graph remained highly sparse and imbalanced: the mean entity degree was only 3.57, the median degree was 1, and the degree distribution showed a pronounced long tail with more than 200,000 entities having degree 1. In addition, the graph was dominated by high-frequency but weakly informative relations such as `dbo:language` (108,087

instances), `dbo:thumbnail` (78,098), and `dbo:wikiPageRedirects` (77,046), along with a large number of relations occurring only a handful of times. These statistics indicate that a graph is too sparse and imbalanced for a link prediction model to generalize reliably.

Building on this analysis, we implemented two-stage pruning strategy including: (i) semantic filtering to remove administrative and noisy relations, and (ii) frequency-based pruning to discard low-support relations and poorly connected entities.

Graph stage	Triples	Entities
Raw extracted graph	2,039,012	–
Core ontology graph	544,161	304,395
Strict pruned graph $(100, 10)$	10,183	1,091
Relaxed pruned graph $(50, 6)$	51,607	7,584

Table 3
Progressive graph pruning on Hindi DBpedia.Even the relaxed setting retains less than 10% of the core ontology graph, underscoring the severity of the sparsity problem

Using the PyKEEN library, we evaluated TransE [15] and ConvE [16] on the pruned graph. Under the stricter pruning setting—relations with frequency at least 100 and entities with degree at least 10—the graph contained 10,183 triples over 1,091 entities. As shown in Table 4, initializing entity embeddings with MuRIL [17], a multilingual transformer trained on Indian languages, improved performance over random initialization for both models. The best result was TransE + MuRIL, with Hits@10 of 13.15% and MRR of 0.0646. MuRIL initialization helped ConvE most, lifting its Hits@10 from 2.85% to 9.03%.

We also tested a less aggressive pruning setting (frequency $\geq$50, degree $\geq$6), which expanded the graph to 51,607 triples and 7,584 entities. The larger graph did not consistently improve results and we observed untuned models degraded even with MuRIL initialization and hyperparameter tuning.

Model	Hits@1	Hits@3	Hits@10	MRR
TransE (default init)	0.0098	0.0353	0.1129	0.0487
ConvE (default init)	0.0049	0.0088	0.0285	0.0185
TransE (+ MuRIL)	**0.0226**	**0.0667**	**0.1315**	**0.0646**
ConvE (+ MuRIL)	0.0147	0.0402	0.0903	0.0446

Table 4
Link prediction results on the stricter pruned Hindi DBpedia graph. MuRIL initialization improves both TransE and ConvE.

The modest scores can be explained by the extreme sparsity of the graph. For comparison, the standard FB15k-237 [18] dataset for knowledge graph completion has 310,116 triples with around 14,000 triples whereas our strict pruned graph has only 10,183 triples across 1091 entities. The fact that MuRIL initialization improves metrics provides some useful inductive bias but still cannot compensate for the lack of training data. For that reason, we treat OIE as the more immediate priority for enrichment.

4.5. Deployment and Performance Evaluation

The Hindi KG was deployed in a Dockerized environment with an exposed SPARQL endpoint. To check deployment readiness, we ran a 60-second load test with 10 concurrent threads. The endpoint sustained over 390 QPS, with single-query response time of 12 ms, rising to 64 ms at 50 concurrent users. Peak memory was 1.3 GB. The setup is straightforward to replicate using the provided Docker image.

5. Challenges and Limitations

Throughout the project, there were several challenges.

Metric	Value
Peak throughput	>390 QPS
Load test duration	60 s
Concurrent threads	10
Single-query response time	12 ms
Response time at 50 users	64 ms
Peak memory usage	1.3 GB RAM

Table 5
Deployment performance of the Hindi KG SPARQL endpoint.

5.1. Data Scarcity

A major limitation has been the lack of a large, high-quality annotated Hindi OIE corpus. Synthetic data helped expand training resources, but there is a gap in diversity. In addition, alignment with evaluation benchmarks remained difficult limiting both model development and reliable comparison.

5.2. Precision-Recall Trade-off

The precision-recall trade-off in the hybrid pipeline was a second unresolved issue. SLMs generate more candidate triples than rule-based systems and improve recall, but they also introduce more noise. They are not reliable at filtering their own outputs, which limited how much precision could be recovered post-extraction. Possible mitigation strategies could include (i) training a lightweight binary classifier on (sentence, triple) pairs to filter low-confidence extractions, (ii) cross-validating triples against a second extraction method and retaining only the intersection.

5.3. Knowledge Graph Sparsity

KG sparsity was also a related bottleneck. The infobox-derived graph was too sparse for link prediction to work well, which means stronger OIE is needed before completion methods can be fully effective.

6. Impact and Future Work

With this project, we first provide a detailed analysis of various prompting strategies across various Small Language Models. We also introduce a hybrid OIE architecture that achieves 66% recall on the Hindi-BenchIE dataset and deliver an open-source pipeline for knowledge graph enrichment from Hindi text extraction to predicate linking.

Several directions remain for future work. Finetuning on LLM-judged synthetic data is the most promising path to improving both precision and recall simultaneously. Precision could also be improved with a trained classifier in place of a rule-based or generative filter. Finally, scaling the pipeline and running extraction on the full Hindi Wikipedia corpus would help build a denser knowledge graph and make downstream link prediction more effective.

7. Conclusion

Over two years and two GSoC cycles, we built a working extraction pipeline for Hindi DBpedia, evaluated SLMs on the OIE task, and measured what the hybrid approach actually gains and costs. This constructs a clear path from unstructured Hindi text to a structured, linked, and queryable knowledge graph. With this project we enhance the DBpedia ecosystem and also serve as a blueprint for similar efforts in other under-resourced languages, contributing to a more linguistically diverse and equitable Semantic Web.

Declaration on Generative AI

During the preparation of this work, the author(s) used ChatGPT for Grammar and spelling check. After using this, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication's content.

References

[1] R. Mishra, S. Singh, R. Shah, P. Kumaraguru, P. Bhattacharyya, IndIE: A Multilingual Open Information Extraction Tool For Indic Languages, in: Findings of the Association for Computational Linguistics: IJCNLP 2023, 2023, pp. 410–424. URL: https://aclanthology.org/2023.findings-ijcnlp.28/.

[2] B. Singh, S. V. P. K. Kandru, A. Sharma, V. Varma, Massively multilingual language models for cross lingual fact extraction from low resource indian languages, in: Proceedings of the 19th International Conference on Natural Language Processing (ICON), 2022, pp. 11–18.

[3] A. Graciotti, Knowledge extraction from multilingual and historical texts for advanced question answering., in: DC@ ISWC, 2023.

[4] P.-L. HUGUET CABOT, et al., From text to knowledge: multilingual information extraction for knowledge graph construction (2025).

[5] J. Lehmann, R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P. N. Mendes, S. Hellmann, M. Morsey, P. Van Kleef, S. Auer, et al., Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia, Semantic web 6 (2015) 167–195.

[6] M. Dojchinovski, J. Forberg, J. Frey, M. Hofer, D. Streitmatter, K. Yankov, The dbpedia technology tutorial, in: 2021 Workshops and Tutorials-Language Data and Knowledge, LDK, 2021.

[7] D. Kontokostas, C. Bratsas, S. Auer, S. Hellmann, I. Antoniou, G. Metakides, Internationalization of linked data: The case of the greek dbpedia edition, Journal of Web Semantics 15 (2012) 51–61.

[8] J. Cojan, E. Cabrio, F. Gandon, Filling the gaps among dbpedia multilingual chapters for question answering, in: Proceedings of the 5th Annual ACM Web Science Conference, 2013, pp. 33–42.

[9] P. Qi, Y. Zhang, Y. Zhang, J. Bolton, C. D. Manning, Stanza: A python natural language processing toolkit for many human languages, in: Proceedings of the 58th annual meeting of the association for computational linguistics: system demonstrations, 2020, pp. 101–108.

[10] R. Mishra, P. Desur, R. Shah, P. Kumaraguru, Multilingual coreference resolution in low-resource south asian languages, in: Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024), 2024, pp. 11813–11826.

[11] P.-L. H. Cabot, R. Navigli, Rebel: Relation extraction by end-to-end language generation, in: Findings of the association for computational linguistics: EMNLP 2021, 2021, pp. 2370–2381.

[12] N. De Cao, L. Wu, K. Popat, M. Artetxe, N. Goyal, M. Plekhanov, L. Zettlemoyer, N. Cancedda, S. Riedel, F. Petroni, Multilingual autoregressive entity linking, Transactions of the Association for Computational Linguistics 10 (2022) 274–290.

[13] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. Chi, Q. Le, D. Zhou, Chain-of-thought prompting elicits reasoning in large language models (2022). URL: https://arxiv.org/abs/2201.11903.

[14] S. Yao, J. Zhao, D. Yu, N. Du, E. Durmus, T. Lin, Y. Chen, H. Chen, J. Gu, H.-T. Cheng, et al., React: Synergizing reasoning and acting in language models (2022). URL: https://arxiv.org/pdf/2210.03629.

[15] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, O. Yakhnenko, Translating embeddings for modeling multi-relational data, Advances in neural information processing systems 26 (2013). URL: https://papers.nips.cc/paper_files/paper/2013/file/1cecc7a77928ca8133fa24680a88d2f9-Paper.pdf.

[16] T. Dettmers, P. Minervini, P. Stenetorp, S. Riedel, Convolutional 2d knowledge graph embeddings, in: Proceedings of the AAAI conference on artificial intelligence, volume 32, 2018. URL: https://arxiv.org/abs/1707.01476.

[17] S. Khanuja, D. Bansal, S. Mehtani, S. Khosla, A. Dey, B. Gopalan, D. K. Margam, P. Aggarwal, R. T. Nagipogu, S. Dave, S. Ramoji, V. P. Namboodiri, M. M. Khapra, P. Kumar, MuRIL: Multilingual

Representations for Indian Languages, in: Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, 2021, pp. 1745–1751. URL: https://aclanthology.org/2021.eacl-main.150/.

[18] K. Toutanova, D. Chen, Observed versus latent features for knowledge base and text inference, in: Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality, 2015, pp. 57–66.