A Guided and Flexible LLM-based Approach for Knowledge Extraction from Text

Carmelle Meli Songuon
Orange Research
Belfort, France
carmelle.melisonguon@orange.com

Yoan Chabot
Orange Research
Belfort, France
yoan.chabot@orange.com

Lucas Jarnac
Orange Research, Université de Lorraine, CNRS, LORIA
Belfort, Nancy, France
lucas.jarnac@orange.com

Viet-Phi Huynh
Orange Research
Paris, France
vietphi.huynh@orange.com

Abstract

Extracting structured knowledge from text in the form of (subject, predicate, object) triples is a key task for many artificial intelligence applications, in particular for knowledge graphs (KGs) construction. In closed information extraction (cIE) where the extracted triples are constrained by a predefined KG schema, most existing approaches rely on Wikidata for entities and relation extraction. As a result, they often lack the flexibility to adapt to other KGs without prior retraining and costly data annotation. In this paper, to address these limitations, we propose FlexCIE, an approach for cIE which leverages Large Language Models (LLMs) and a KG completeness analysis tool. Given an input text, it identifies a list of entity mentions, links them to entities in the target KG using embedding techniques combined with LLMs, and constructs triples from these extracted entities using relevant properties retrieved from the KG. This enables the use of LLMs for cIE, while ensuring that the generated triples are accurate and compliant with the KG schema. Therefore, our work contributes to making cIE more practical and flexible, particularly for domain specific or enterprise KGs.

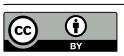
CCS Concepts

• **Computing methodologies** → **Information extraction**; • **Information systems** → *Graph-based database models.*

Keywords

closed information extraction, knowledge graphs, large language models

ACM Reference Format:
Carmelle Meli Songuon, Yoan Chabot, Lucas Jarnac, and Viet-Phi Huynh. 2025. A Guided and Flexible LLM-based Approach for Knowledge Extraction from Text. In *Knowledge Capture Conference 2025 (K-CAP '25), December 10–12, 2025, Dayton, OH, USA.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3731443.3771373



K-CAP '25, Dayton, OH, USA

ACM ISBN 979-8-4007-1867-0/25/12
https://doi.org/10.1145/3731443.3771373

1 Introduction

Knowledge Graphs (KGs) have become a central tool for structuring and leveraging data in many industrial fields. In an environment where data are continuously generated, they offer a convenient way to organize information so that it can be easily used for downstream tasks such as information retrieval, question-answering or data analysis [1, 12]. To build and maintain KGs, some approaches rely on manual efforts [1, 3] which, although accurate, often require domain expertise knowledge and can be resource-intensive. Alternative approaches using automatic knowledge extraction techniques have been developed to address scalability issues, in particular those leveraging unstructured data such as texts. However, automatically transforming unstructured text into structured triples remains a challenging task, as it involves several complex subtasks, such as Entity Linking (EL) and Relation Extraction (RE). In this work, we focus on closed information extraction (cIE), the task of extracting from a text a set of triples that conform to a predefined Knowledge Base (KB) schema [6]. Recent advances in machine learning have significantly improved performance in this area. Some state-of-the-art approaches address the cIE task as a sequence-to-sequence generation problem, usually by leveraging transformer-based Pre-trained Language Models (PLMs) [4, 6, 11]. While these approaches proved to be powerful, they required a large dataset for training on a specific KG, and since KGs are huge and dynamic, these approaches result in annotation and retraining costs [2].

Inspired by recent research on the use of LLMs for knowledge extraction [8], we propose FlexCIE, an approach to cIE that leverages LLMs for their natural language understanding capabilities, and Recoin [10], a KG completeness analysis tool. The designed extraction process consists of two main phases. First, entities are extracted from the text and then aligned with the target KG using embedding techniques and LLMs. In the second phase, triples are constructed from the entities previously extracted. A particularity of our approach is to guide the extraction of triples in the second phase by providing the LLM a list of relevant properties to be used. This is done using the reference KG and Recoin to identify desirable properties for an entity. FlexCIE enables to finely control the entities and relations used in the extraction process, so that they conform to the KG schema. In addition, it supports the extraction of emerging entities [5] and literal values (*e.g.,* date, quantities). In summary, the main contributions of this work are as follows :

- We propose a novel approach to closed Information Extraction that leverages LLMs in a modular architecture.
- Our solution does not require model retraining or costly data annotation to adapt to enterprise or domain-specific KGs.

The remainder of the paper is organized as follows. Section 2 reviews related work in the field of cIE and Section 3 describes our proposed approach. In Section 4, we present the experimental setup and Section 5 discusses the results. Finally, Section 6 concludes the paper and outlines directions for future work.

2 Related Work

As mentioned in the introduction, cIE refers to the task of extracting a set of (subject, predicate, object) triples from a text, where the entities and relations must correspond to those defined in a target KG. This constraint makes the task particularly challenging compared to *open Information Extraction (oIE)*, also referred to in the literature as text-to-graph [6, 7], where the extracted entities and relations are in a free form. There exist two directions of research to address cIE: pipeline-based approaches and end-to-end models.

Pipeline-based methods break the cIE into sequential subtasks: Named Entity Recognition (NER) to identify entity mentions, EL to associate mentions with KG entities, and Relation Classification (RC) to connect entities with predicates to form triples. While this modular design allows for reuse and independent optimization of each component [2], it suffers from error propagation, where mistakes in earlier stages affect later results [6].

End-to-end systems have been proposed to mitigate the error propagation by jointly performing the extraction and the KG alignment of entities and relations in a unified model [6, 7]. Among them, generative approaches have gained particular attention. These methods frame cIE as a machine translation task, where the model takes a raw text as input and generates a set of structured triples as output. In this line of work, [6] introduces GenIE, the first generative approach to cIE. The GenIE model autoregressively generates the linearized representation of the set of triples expressed in a text, outputting the Wikipedia title of each entity and relation labels from Wikidata. KnowGL [4] extends the GenIE model by including in the generated output the entity surface form and the corresponding entity type and label, which can be directly mapped to Wikidata IDs. Another recent system addressing cIE is ReLiK [9], which adopts a Retriever-Reader architecture in which the retriever module identifies candidate entities and relations, and the reader module aligns them to the relevant spans in the source text.

These models have significantly advanced the state-of-the-art in cIE by improving automation and making it possible to process a wider range of texts. However, they remain limited in flexibility as they are trained on public resources such as Wikidata [3], making adaptation to new KGs difficult without retraining or additional data annotation. In the next section, we present FlexCIE, our approach to cIE that addresses these limitations.

3 FlexCIE Approach

FlexCIE is our approach for cIE, that leverages LLMs to perform the extraction and Recoin [10], a KG analysis tool that allows the identification of a list of desirable properties for an entity. It is a pipeline consisting of four main modules. The first module undertakes the extraction of entity mention that may appear in the input text. The second module handles the linking of these mentions to the entities of the reference KG. The third module consists of identifying a list of properties that can be used to form triples, and the last module is tasked to construct triples from the extracted entities and these properties. Figure 1 illustrates FlexCIE knowledge extraction process.

1) Mention Extraction. As shown in Figure 1, the first step consists of extracting the list of entity mentions present in the text by querying an LLM. To better prepare for the next alignment step with the entities from the reference KG, the LLM is also tasked with providing a type and a brief description for each extracted mention. This additional information will help in the next step for disambiguation purposes.

2) Entity Linking. Once the list of mentions has been extracted from the text, the next step is to link these mentions to the KG entities. To achieve this, we perform the following steps:

(1) The entity mention embedding is computed from the concatenation of its mention label, the entity type, and its description returned by the LLM in the first step.
(2) Using the pre-computed embeddings of the KG entities, we perform a similarity calculation to obtain the top k entities in the KG closest to the mention.
(3) The candidates returned by the similarity calculation are then passed to the LLM along with the initial mention to return the best candidate. Each candidate from the KG is provided with its QID, label, and description if available.

This module returns a list of entities, each consisting of its label in the text, its type, and the identifier of the entity in the KG that corresponds to the mention, or *"null"* if the entity is not present in the KG. The information on the type of the mention has been retained for disambiguation purposes in the triple construction phase.

3) Relevant Property Retrieval. In order to constrain the LLM to extract valid and correctly formatted relations, a list of desirable properties that may be missing in the KG for an entity is identified using a KG completeness analysis tool such as Recoin [10] in our case. Recoin (Relative completeness indicator) is a tool initially developed to assess the degree of completeness of knowledge in Wikidata. Recoin is based on the relative completeness approach *i.e.,* that compares the triples associated with a given entity with those of similar entities to suggest missing properties. For example, if most countries have a *"president"* property but a given country lacks it, Recoin will detect this property as missing for that country. Based on this tool, we consider two approaches. A first approach consists of retrieving from the KG the list of properties instantiated for the entity identified by the EL module, and complementing them with the missing properties using Recoin. This will allow the extraction of potentially existing triples in the KG, with the aim of either reinforcing the confidence in the fact (since it is confirmed by a new textual source), updating the fact if it is outdated, or completing the fact (*e.g.,* for multivalued properties). The second approach relies on Recoin to retrieve the most frequent properties that are instantiated in the class(es) to which the extracted entities

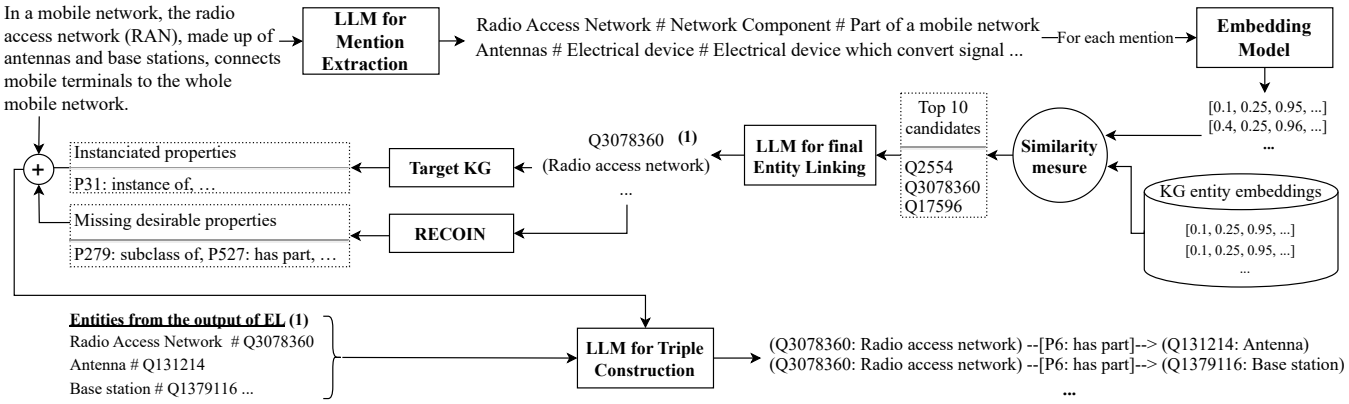


Figure 1: Description of FlexCIE. Given an input text, entity mentions are extracted. Each extracted mention is linked to an entity in the KG, then a list of properties that can be used to form triples from that entity is identified. Once all entities and properties have been identified, the set is provided to the LLM along with the initial text for the generation of triples.

belong. These approaches allow us to gather a set of properties that are likely to appear in the text.

4) Triple Construction. The final step of the IE pipeline is to construct the relevant triples entailed by the text, based on the entities and properties retrieved in the previous modules. To this end, an LLM is queried with the list of entities and properties to be used, given the input text. In addition, the LLM can also be tasked with extracting literals (*e.g.,* dates, numbers, URLs) necessary for completing triples. To improve the quality of the generated output, self-evaluation guidelines and refinement steps are added to the prompt. The output format of this module has the following structure: `(QID: label) –[PID: relation_label]–> (QID: label)`. *QID* refers to the unique identifier of the entity in the KG or *null* in case of literals or emerging entities. *PID* is for the identifier in the KG of the property linking the two entities.

4 Experiments

Data: For evaluation purposes, we implement FlexCIE using Wikidata[1] as target KB for EL and RE (*i.e., Relevant Property Retrieval* and *Triple Construction*). However, FlexCIE can be adapted to different KGs, requiring only a supporting KG for EL and a property recommendation mechanism that guides the LLM to focus on relevant properties. For the dataset, we choose REBEL [7], a distantly supervised benchmark built from Wikipedia abstract and Wikidata triples. REBEL is commonly used by recent cIE models for training and evaluation. In our experiments, we use the extended version from [6]. Given the large size of Wikidata, we restrict the scope of the entity embeddings computation for EL to a subset of around 2.5 million entities, which correspond to all the entities appearing in the REBEL dataset. Since our approach does not require training, we used only the test set which contains approximately 175,000 sentences. However, evaluating the entire test set would have been time-consuming, as the evaluation had to be performed at the sentence level, although our method supports full texts as input. For this reason, we randomly sampled 1,000 inputs from the test set to report the results. This evaluation subset contains 2,296 unique entities and 2,623 annotated triples.

Evaluation metrics: We measure the performance in terms of precision, recall, and F1-score. A triple is considered correct when the relation and the two corresponding entities (subject and object) are properly identified. In addition, we perform an independent evaluation of the EL module based on the set of entities present in the annotated triples. We considered an entity correctly predicted if its predicted span overlaps the gold span and is linked to the correct Wikidata identifier (QID) or Wikipedia title. Predicted entities that are not in the gold annotations are ignored. An entity not predicted by the system but present in gold annotations is counted as false negative, while an incorrectly disambiguated entity or one with non-overlapping span is counted as false positive. We adopt weak span matching rather than strong matching since entity mention can be expressed in different valid forms. For instance in the sentence: *"Austins Bridge is an American Christian country band originally formed in Austin, Texas."*, the entity *Austin* (referring to the city, with the identifier *Q16559* in Wikidata) can be identified with the span [74, 80] (corresponding to *"Austin"*) or [74, 87] (corresponding to *"Austin, Texas"*), both being valid mentions of the same entity.

Baselines: We compare FlexCIE to two recent state-of-the-art systems: KnowGL and ReLiK, described in section 2. For ReLiK, we evaluated performance on EL and RE on the test subset. Unfortunately, we were unable to properly evaluate EL on KnowGL, as the publicly available model output does not provide spans for extracted entities.

FlexCIE Setup: We use *Qwen* as the language model for the generation tasks. For the mention extraction and triple construction steps, we used *Qwen3-32B-AWQ*[2], a 32 billion-parameter model, and for the linking step, we used *Qwen2.5-14B-Instruct-AWQ*[3], a 14 billion-parameter model. We can notice that these models are smaller than other LLMs such as GPT-3 with 175 billion-parameter. For the evaluation on the REBEL dataset, we did not use the KG completeness analysis tool to retrieve entity missing properties

[1]Dump from June 2025

[2]https://huggingface.co/Qwen/Qwen3-32B-AWQ

[3]https://huggingface.co/Qwen/Qwen2.5-14B-Instruct-AWQ

Table 1: Entity Linking (EL) and Relation Extraction (RE) evaluation results on full REBEL and a subset of REBEL. P stands for precision, R stands for recall, and F1 stands for F1-score. The best results are in bold.

Dataset	Model	EL			RE		
		P	R	F1	P	R	F1
Full	KnowGL	–	–	82.73	73.88	67.85	70.74
	$\text{ReLiK}_{\text{large}}$	–	–	85.1	–	–	75.6
Subset	KnowGL	–	–	–	**55.04**	36.47	**43.87**
	$\text{ReLiK}_{\text{large}}$	**99.28**	63.78	77.67	22.23	28.52	24.98
	FlexCIE	97.95	**75.87**	**85.51**	27.73	**37.03**	31.71

in the KG as it would have reduced performance since REBEL is built only from Wikidata entities and their instantiated properties. Furthermore, to ensure a fair comparison with existing systems, we instructed the LLM not to extract literals or emerging entities. However, we still observed that sometimes the model generated triples containing literals or fabricated QIDs. To address this, we applied a final post-processing step to filter out any such triples.

5 Results

As shown in Table 1 our system achieves remarkable performance on EL, outperforming ReLiK by more than 10 points in recall. Besides, the performance on RE is lower across all systems, including ours. KnowGL achieves the best precision and F1 score for RE while FlexCIE obtains the highest recall. This performance drop can be explained by the fact that the REBEL dataset was built using distant supervision. As a result, many valid triples in the text are not annotated in the gold set and are therefore counted as false positives when predicted by FlexCIE or other models. This limitation significantly affects the precision across all systems. Furthermore, the same triple can often be expressed using different but semantically equivalent relations (*e.g., developer* vs *discover or inventor*).

In some cases, predicted relations appear in inverse form compared to their gold annotation (*e.g., has part* vs *part of*). This complicates the evaluation and penalizes systems even when their outputs are correct. However, it is important to note that FlexCIE achieves these results without having been trained, unlike KnowGL and ReLiK, which were both trained on the REBEL dataset.

6 Conclusion and Future Work

In this work, we presented FlexCIE, a flexible pipeline-based approach for closed information extraction that leverages an LLM and a KG completeness analysis tool to extract structured knowledge from text. FlexCIE allows precise control and guidance of the LLM through the extraction process and supports the extraction of literal values and emerging entities. In addition, it can be easily adapted to different KGs without requiring retraining or data annotation. However, since the approach relies on an LLM, the outputs are not fully deterministic and may vary between executions. This variability is particularly notable during the final phase of triple construction, where the system may hallucinate. In future work, it would be interesting to perform a manual evaluation on a sample output to measure the quality of the predictions and overcome the limitations of the REBEL dataset. Another limitation is the execution time, which is longer than in other approaches due to multiple sequential LLM calls and processing steps. Finally, our approach needs further testing on domain-specific datasets.

References

[1] Aidan Hogan et al. 2022. Knowledge Graphs. *ACM Comput. Surv.* 54, 4 (2022), 71:1–71:37. doi:10.1145/3447772

[2] Cedric Möller et al. 2024. DISCIE-Discriminative Closed Information Extraction. In *The Semantic Web - ISWC 2024 - 23rd International Semantic Web Conference, Baltimore, MD, USA, November 11-15, 2024, Proceedings, Part II (Lecture Notes in Computer Science, Vol. 15232)*, Gianluca Demartini, Katja Hose, Maribel Acosta, Matteo Palmonari, Gong Cheng, Hala Skaf-Molli, Nicolas Ferranti, Daniel Hernández, and Aidan Hogan (Eds.). Springer, 23–40. doi:10.1007/978-3-031-77850-6_2

[3] Denny Vrandečić et al. 2014. Wikidata: a free collaborative knowledgebase. *Commun. ACM* 57, 10 (2014), 78–85. doi:10.1145/2629489

[4] Gaetano Rossiello et al. 2023. KnowGL: Knowledge Generation and Linking from Text. In *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023*, Brian Williams, Yiling Chen, and Jennifer Neville (Eds.). AAAI Press, 16476–16478. doi:10.1609/AAAI.V37I13.27084

[5] Johannes Hoffart et al. 2014. Discovering emerging entities with ambiguous names. In *23rd International World Wide Web Conference, WWW '14, Seoul, Republic of Korea, April 7-11, 2014*, Chin-Wan Chung, Andrei Z. Broder, Kyuseok Shim, and Torsten Suel (Eds.). ACM, 385–396. doi:10.1145/2566486.2568003

[6] Martin Josifoski et al. 2022. GenIE: Generative Information Extraction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, Marine Carpuat, Marie-Catherine de Marneffe, and Iván Vladimir Meza Ruíz (Eds.). Association for Computational Linguistics, 4626–4643. doi:10.18653/V1/2022.NAACL-MAIN.342

[7] Pere-Lluís Huguet Cabot et al. 2021. REBEL: Relation Extraction By End-to-end Language generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 16-20 November, 2021*, Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (Eds.). Association for Computational Linguistics, 2370–2381. doi:10.18653/V1/2021.FINDINGS-EMNLP.204

[8] Roos M. Bakker et al. 2024. From Text to Knowledge Graph: Comparing Relation Extraction Methods in a Practical Context. In *Joint Proceedings of the ESWC 2024 Workshops and Tutorials co-located with 21th European Semantic Web Conference (ESWC 2024), Hersonissos, Greece, May 26-27, 2024 (CEUR Workshop Proceedings, Vol. 3749)*, Bruno Sartini, Joe Raad, Pasquale Lisena, Albert Meroño-Peñuela, Michael Beetz, Inès Blin, Philipp Cimiano, Jacopo de Berardinis, Simon Gottschalk, Filip Ilievski, Nitisha Jain, Jongmo Kim, Michaela Kümpel, Enrico Motta, Ilaria Tiddi, and Jan-Philipp Töberg (Eds.). CEUR-WS.org.

[9] Riccardo Orlando et al. 2024. ReLiK: Retrieve and LinK, Fast and Accurate Entity Linking and Relation Extraction on an Academic Budget. In *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, Lun-Wei Ku, Andre Martins, and Vivek Srikumar (Eds.). Association for Computational Linguistics, 14114–14132. doi:10.18653/V1/2024.FINDINGS-ACL.839

[10] Vevake Balaraman et al. 2018. Recoin: Relative Completeness in Wikidata. In *Companion of the The Web Conference 2018 on The Web Conference 2018, WWW 2018, Lyon , France, April 23-27, 2018*, Pierre-Antoine Champin, Fabien Gandon, Mounia Lalmas, and Panagiotis G. Ipeirotis (Eds.). ACM, 1787–1792. doi:10.1145/3184558.3191641

[11] Zikang Zhang et al. 2025. A Survey of Generative Information Extraction. In *Proceedings of the 31st International Conference on Computational Linguistics, COLING 2025, Abu Dhabi, UAE, January 19-24, 2025*, Owen Rambow, Leo Wanner, Marianna Apidianaki, Hend Al-Khalifa, Barbara Di Eugenio, and Steven Schockaert (Eds.). Association for Computational Linguistics, 4840–4870.

[12] Ralph Grishman. 2012. Information extraction: capabilities and challenges. *International Winter School in Language and Speech Technologies WSLST* 41 (2012).