Knowledge Graphs can play together: Addressing knowledge graph alignment from ontologies in the biomedical domain

Hanna Abi Akl, Dominique Mariko, Yann-Alan Pilatte, Stéphane Durfort, Nesrine Yahiaoui, Anubhav Gupta

Knowledge Graphs can play together: Addressing knowledge graph alignment from ontologies in the biomedical domain

Hanna Abi Akl[1,2][a], Dominique Mariko[3], Yann-Alan Pilatte[3], Stéphane Durfort[3], Nesrine Yahiaoui[3] and Anubhav Gupta[3]

[1]*Data ScienceTech Institute (DSTI), 4 Rue de la Collégiale, 75005 Paris, France*

[2]*Université Côte d'Azur, Inria, CNRS, I3S*

[3]*Yseop, 4 Rue de Penthièvre, 75008 Paris, France*

hanna.abi-akl@dsti.institute, {dmariko, ypilatte, sdurfort, nyahiaoui, agupta}@yseop.com

Keywords: Knowledge Graphs, Ontologies, Natural Language Processing, Information Extraction

Abstract: We introduce DomainKnowledge, a system that leverages a pipeline for triple extraction from natural text and domain-specific ontologies leading to knowledge graph construction. We also address the challenge of aligning text-extracted and ontology-based knowledge graphs using the biomedical domain as use case. Finally, we derive graph metrics to evaluate the effectiveness of our system compared to a human baseline.

1 INTRODUCTION

In the era of Large Language Models (LLMs), Knowledge Graphs (KGs) have resurfaced to play an important role, whether as complements to LLM-based technology to enhance predictions in Retrieval Augmented Generation (RAG) models, or as standalone systems that more faithfully capture factual information (Pan et al., 2023b), (Peng et al., 2023), (Vogt et al., 2022). The ongoing problem of hallucinations in LLMs draws a line on their reliability and questions the interpretability and explainability of their outputs. The inability to trust the responses of these deep learning models leads to much hesitation in implementing and deploying them in production, especially in sensitive domains such as healthcare (Pan et al., 2023a).

KGs, on the other hand, have demonstrated their staying power by circumventing the black-box mechanism of LLMs and offering open and traceable representations of domain information (Pan et al., 2023a). Their staying power is also strengthened by their integration with both deep learning solutions and more classical frameworks like ontologies that provide formal representations of knowledge (Pan et al., 2023b), (Vogt et al., 2022). A major weakness they exhibit however is their difficulty in integrating and aligning new knowledge. Unlike LLMs, which benefit from fine-tuning to add new knowledge, ontologies, and KGs by extension, require a lot of work in order to enrich their representations in a single domain or extend to a new one (Van Tong et al., 2021). This weakness makes these technologies less transferable on their own which is why they are often utilized as components in larger systems that can benefit from their advantages (Peng et al., 2023).

In this work, we present DomainKnowledge, a system comprised of a workflow of information extraction (IE) from unstructured text leading to the construction of a consolidated domain KG. We showcase strategies in our implementation to combine domain knowledge from ontological sources and amount to a generalized domain-specific KG mapping input text entities to higher-order concepts. We also introduce metrics inspired from graph theory to evaluate our system. The rest of the work is structured as follows. Section 2 presents related work in the literature. Section 3 describes our methodology. In section 4 we present our experimental setup. Section 5 discusses our findings with an analysis of our results. Finally, we conclude with future directions of work in section 6.

2 RELATED WORK

This section covers the literature pertaining to text-to-graph extraction techniques as well as KG alignment methods.

[a] https://orcid.org/0000-0001-9829-7401

2.1 Knowledge graph construction from text

Research in IE shows different methods to construct KGs from text. In their work, (Liu et al., 2022) surveyed different methods for text information extraction from relation triples. They explored and compared systems that capture relations in the form of triples, spans and clusters using symbolic and deep learning techniques at the syntactic and semantic levels. (Kamp et al., 2023) compared rule-based open IE engines to machine learning extraction systems and found a trade-off between implementation and precision. While rule-based systems exhibited better overall performance in identifying and extracting relations, they were much harder and more exhaustive to implement than off-the-shelf machine learning models.

Natural language processing (NLP) methods like sentence chunking, domain entity classification, relation classification and sentence-to-graph techniques to manipulate text directly through graph properties have achieved promising results that exploit syntactic and semantic text attributes through models built on robust rule engines. These techniques, while capable of controlling the type of information to extract, have shown limitations when it comes to extending them to cover more exhaustive knowledge (Chouham et al., 2023), (Dong et al., 2023), (Motger and Franch, 2024), (Yu et al., 2022). Other research geared toward machine and deep learning technology combines these methods with classical NLP techniques for better results. In their work, (Qian et al., 2023) proposed an IE pipeline that combines pattern-based, machine learning and LLM extractions that undergo rule-based and machine learning scoring to decide on keeping or discarding extracted information. Transformer-based approaches have also been applied to leverage embeddings information and transform them to node properties in graphs constructed from text (Friedman et al., 2022), (Melnyk et al., 2022). These methods have showcased a better ability at capturing text properties as node representations. Novel hybrid systems making use of the availability of LLM technology leveraged their prompting abilities to provide domain annotations for better information extraction (Dunn et al., 2022), combine them with other sources of knowledge like ontologies (Mihindukulasooriya et al., 2023), (Wadhwa et al., 2023) for better coverage, and even use text generation techniques as a comparative benchmark to identify viable relation candidates for extraction (Hong et al., 2024).

2.2 Knowledge graph alignment

Several research avenues explore graph-based techniques for KG alignment. (Zeng et al., 2021) survey distance-based and semantic matching scores for effective entity alignment in KGs. In their work, (Zhang et al., 2021) propose systems based on stacked graph embeddings of different graph components like neighboring entities and predicates to improve entity alignment. Other methods focus on integrating deep learning models to better express graph component properties and answer the graph alignment problem. (Chaurasiya et al., 2022), (Dao et al., 2023) and (Fanourakis et al., 2023) show that graph neural networks performed well in aligning different graph entities when paired with distance-based graph features and embeddings. In their work, (Yang et al., 2024) show that LLMs could be leveraged to decompose the alignment problem into multiple choice questions referring to sub-tasks to approximate the alignment of entities with respect to neighboring nodes. (Trisedya et al., 2023) propose a system composed of an attribute aggregator and a node aggregator to combine both node and relation properties and get better alignment predictions. (Zhang et al., 2023) showcase a similar method aggregating property, relationship and attribute triples to get a more complete representation of entities and aid the entity alignment process.

Finally, neuro-symbolic systems aiming to combine both classical rule-based techniques with sub-symbolic architectures have also been proposed to tackle the graph alignment problem. (Cotovio et al., 2023) survey neural network architectures and reinforcement learning methods for better entity alignment predictions. In their work, (Xie et al., 2023) convert different KGs into vector space embeddings and combine them with graph neural networks to create transitions and better delimit the best alignment for a node entity. (Abi Akl, 2023) show the benefits of using logic neural networks as reasoners with a ruleset derived from upper ontologies in a hybrid system to align entities from different KGs.

3 METHODOLOGY

The DomainKnowledge system proposes a data acquisition and transformation pipeline that leverages NLP and graph techniques to extract meaningful relationships from raw text and store them in graph structures to create a domain vocabulary. It consists of the following components:

- An IE pipeline which handles the relationship extraction. The IE component depends on the docu-

ment or text extraction process that precedes it, which should be capable of extracting raw text and transforming it into a list of sentences, since the IE pipeline identifies relationships at sentence level.

- A knowledge storage system which references the graph database storage and KG construction.

The system workflow can be summarized in the following steps:

- Initiate a generic pipeline to identify and extract relations from raw text
- Define a ruleset for meaningful relations
- Prune relations to conserve only meaningful ones
- Export relations into semantic graph structures
- Generate the domain vocabulary from graph relationships

3.1 System overview

The user provides a number of documents from the same domain (e.g., Pharmaceutical). The documents are processed one by one as raw texts. The DomainKnowledge pipeline analyzes the texts as sentences and extracts relationships as triples of the form *(subject,relation,object)*. Relationships are identified with the help of a domain ontology that emphasizes important domain words to look out for, e.g., MedDRA for Pharmaceutical. Once relationships are extracted, a set of rules is applied to prune the bad ones. These rules can vary from simple, e.g., eliminating relationships with missing elements in the triples, to more complex, e.g., evaluating the nature of the relation like verbal versus non-verbal. The ruleset can also be aided by the reference ontology to drop relationships that contain no relevant terms in the subject and/or object entities of the triple. The relationship matrices are then concatenated into one matrix containing all the relevant relationships from all the documents. The matrix is then formatted into several files and exported in a way to preserve the following information:

- Each relationship is unique and is assigned a unique identifier
- Each relationship triple has a clear subject, predicate and object
- Each relationship clearly references the sentence it is extracted from
- Repeated triples are kept
- Each relationship clearly references the document it is extracted from
- Each document is unique and is assigned a unique identifier

The exported information is then ingested into a graph database that conserves the above-mentioned information in a graph network. The graph network is modeled as a subject/object node KG where nodes are subject and object entities and edges are the relation of the triple. Each node has properties associated with it like its unique identifier, the sentence it is extracted from, the name of the document it is extracted from, the unique identifier of the document, the type of the document (e.g., Clinical Study Report, Protocol) and the domain of the document (e.g., Pharmaceutical). Figure 1 shows the high-level architecture of our system.

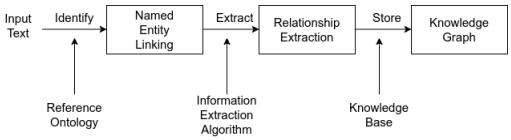

Figure 1: DomainKnowledge pipeline.

3.2 System modules

3.2.1 Extractor

A plain text extractor keeping document layout based on MuPDF[1] in Python.

3.2.2 Annotator

The Annotator's output is based on Stanza's[2] dependency parser which provides a standardized way of representing syntactic dependencies between words in a sentence. Our system produces relations from texts of documents using specific dependencies appearing in Stanza's output. Two main relation types were considered for extraction:

1. Verbal relations: canonical verbal relations take a verb as a cornerstone to build a triple (entity, verb, entity) which can be transformed to (subject, relation, object). The relations are described as follows:
   - *root*: the root of the sentence should usually be the verb that is the main predicate. The root usually has subject(s) and object(s), unless it is intransitive or another verbal dependency interferes.
   - *acl*: behave like roots, but their subject already has a dependency link to another verb (typically, as an object of the root, but not only).

[1] https://shorturl.at/WXmwU

[2] https://stanfordnlp.github.io/stanza/

- *acl:relcl*: adnominal relative clause introduced by relative pronouns, which can either be their subject or object, and reference another subject or object in the context.
- *advcl*: adverbial clauses can have their own subjects and objects, in which case they behave like roots. If they modify nouns and have no subject, they are linked to the verb they modify and its subject.

2. Prepositional relations: we use OpenIE[3], an open-source relation extraction tool, to build prepositional relations from adpositions (e.g., 'as', 'with', 'for', etc.). Considering prepositions as the pivot of the relation, subjects and objects of verbal relations are split into smaller pieces and can match better with ontology terms. We use the dependency tag of the object entities of these relations to identify them with the *nmod* tag.

The Annotator module is also in charge of constructing triples. Each sentence in the original text is decomposed into entity-relation triples and stored with metadata attributes such as document ID, section ID (from the document layout), sentence ID, tokens positions and tokens POS tags. The triples are sets of nodes and relations to be compared with the value of the string data type available in the UMLS metathesaurus[4]. Figure 2 shows the annotation logic.

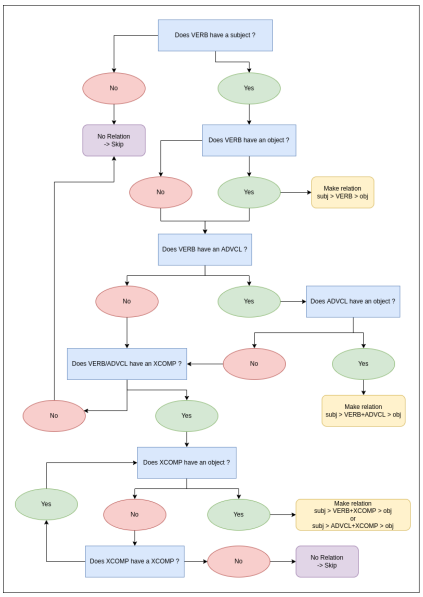

Figure 2: Annotation flowchart.

[3]https://shorturl.at/2VNh0
[4]https://shorturl.at/5F0P8

3.2.3 Aggregator

The Aggregator relies on the National Library of Medicine Unified Medical Language System (UMLS® 2022AA) release. We consider the MRCONSO, MRSAB, MRSTY and MRREL data tables and reorganize their content into a graph data model. We follow the UMLS data types as described in the UMLS Metathesaurus Rich Release Format[5] and keep the data objects as nodes in the graph data model. The data model consists of the following nodes:

- AUI: atom
- CUI: concept
- LUI: term
- SUI: unique string
- TUI: semantic type

We preserve the relationships attributes as defined in the original UMLS Metathesaurus[6]. We turn the incoming relationships into direct links to find paths between text NER nodes and UMLS nodes:

- CUI node has an atom node: $CUI \xrightarrow{HAS\_AUI} AUI$
- SUI node has an atom node: $SUI \xrightarrow{HAS\_AUI} AUI$
- SUI node has concept node: $SUI \xrightarrow{HAS\_CUI} CUI$
- CUI node has semantic type node: $CUI \xrightarrow{HAS\_STY} TUI$

The resulting data model is available in Figure 3.

3.2.4 Merger

Outputs from the Annotator, i.e., entity-relation triples, and the Aggregator, i.e., SUI objects, are mapped with measures of semantic similarity using the following algorithm:

- An exact matching measure using the Levenshtein distance to compute a first similarity score.
- A semantic matching algorithm using cosine similarity to compute a more refined evaluation of entities that do not score highly on the exact matching: each entity is mapped to a 512 dimensional dense vector space, so the semantic matching algorithm can draw similarities from the generated vectors to find associations between two entities.

An additional Named Entity Recognition tagger, i2b2[7], is used to map long triples entities and SUI objects to augment the text-to-ontology mapping. Subject and object entities declared in extracted triples

[5]https://shorturl.at/2HYBj
[6]https://shorturl.at/iV97e
[7]https://shorturl.at/aMN80

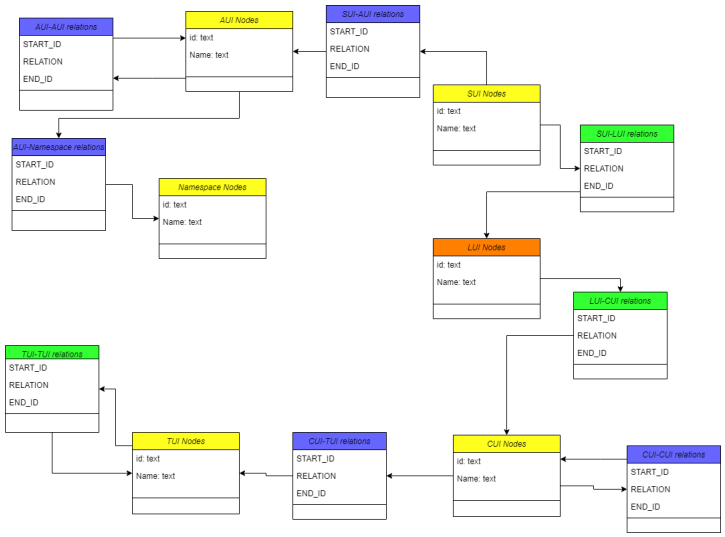

Figure 3: Graph data model.

from sentences are declared as NER nodes in the constructed KG. The Merger outputs a KG construction from a set of pre-configured semantic graphs, consistent with the graph data model, adding the following nodes and edges to the graph:

- Text node linked to another text node: $NER \xrightarrow{TEXT\_LINK} NER$
- Text node matched to SUI node: $NER \xrightarrow{HAS\_LEXICAL} SUI$

An example graph is presented in Figure 4.

3.3 Metrics

We define the following evaluation metrics:

- Coverage (CVRG): Let DT be the set of domain tokens, i.e., any extracted entity from a given text that is also linked, i.e., sharing a direct relation in our KG, to an ontological concept from the domain. Let TT be the set of text tokens, i.e., any extracted entity from the same text. The Coverage is defined as

$$\frac{|DT|}{|TT|} \times 100 \tag{1}$$

- Mapping (MAPG): Let CT be the set of concept tokens, i.e., any extracted entity from a given text sharing the same syntactic (and semantic) name as an ontological concept from the domain. The Mapping is defined as

$$\frac{|CT|}{|DT|} \times 100 \tag{2}$$

- Alignment (ALGT): Let $r_{NER \rightarrow TUI}$ be a direct link from any extracted entity (NER) from a given text to an ontological semantic type (TUI). Let r_{TUI} be a link from any source node to a TUI node, i.e., $r_{TUI} = r_{NER \rightarrow TUI} + r_{CUI \rightarrow TUI}$. The Alignment is defined as

$$\frac{count(r_{NER \rightarrow TUI})}{count(r_{TUI})} \times 100 \tag{3}$$

4 EXPERIMENTS

Experiments were performed on 52 Clinical Study Reports (CSR) with the objective of finding direct relationships between text entities, i.e., subject or object of a triple (NER nodes), and ontological concepts (CUI nodes) and semantic types (TUI nodes). We perform two experiments, each testing an algorithmic approach using the DomainKnowledge pipeline to obtain an alignment from NER nodes to TUI nodes. All experiments were hosted on an instance of Neo4j AuraDB[8]. The first experiment focuses on building sentence clusters based on a sentence similarity score calculated from the triples forming the sentences. The intuition is that similar sentences will very likely be paraphrases or rewording and will trace back to the same higher-order ontological concepts. Grounding these concepts makes the task of aligning NER and TUI nodes easier. The experiment can be broken down to the following steps:

1. The node2vec[9] embeddings is calculated for every NER node.

[8] https://tinyurl.com/yzxneyy5
[9] https://tinyurl.com/55jc525f

Figure 4: Sample output graph.

2. A K-Nearest Neighbors (KNN)[10] clustering algorithm is used to create pairwise clusters of NER nodes using the node2vec embeddings as property.

3. The resulting KNN similarity score *knn_score* for each pair of NER nodes is appended to the relation in their triple if and only if they share a triple.

4. We define the sentence score for a sentence as

$$sentence\_score = \sum_{i=1}^{n} knn\_score_i \qquad (4)$$

where n is the number of relations of the triples extracted from the sentence. All sentences are compared and grouped based on the sentence_score. Sentences with equal sentence_scores underline similar sub-graphs from NER to TUI nodes.

The final step is extracting the relevant NER and TUI nodes from the different sentence group sub-graphs. While this experiment shows promising results on a small batch of sentences, we lacked the resources to handle the computational complexity of the procedure on our ensemble of documents. We therefore did not report results for this method. The second experiment targets ontology alignment directly using the paths between NER, CUI and TUI nodes. The experimental setup is as follows:

1. The degree centrality DC[11] measure is calculated for every NER, SUI, CUI and TUI node. Relations between SUI and AUI nodes are also considered for the SUI degree centrality calculation as they are considered additional information on the representation of a concept. The calculations are based on the following directed graph orientations:
   - $NER \longrightarrow SUI \longrightarrow AUI$ (a)
   - $NER \longrightarrow SUI \longrightarrow CUI \longrightarrow TUI$ (b)

   The aim of this measure is to identify popular nodes.

2. we define the weight w of a relation between 2 nodes A and B as the sum of their degree centralities DC_A and DC_B respectively. Formally, $w_{AB} = w_{BA} = w = DC_A + DC_B$.

3. For each NER node, we traverse the closed sub-graphs respecting the path in (b) while opting for the maximum total weight

$$W = \sum_{i=1}^{n} w_i \qquad (5)$$

where n is the total number of relations between a NER node and a CUI node in a closed sub-graph.

4. We apply the same traversal algorithm to identify the best direct relation between a CUI node and a TUI node.

[10] https://tinyurl.com/yzx7dxv9

[11] https://tinyurl.com/bddhh9e7

5. We finally use the results from the previous two steps to find the best direct relation between a NER node and a TUI node.

We evaluate our DomainKnowledge pipeline against a human baseline consisting of clinical analysts from the biomedical domain who manually perform the alignment on the same dataset and report our results.

5 RESULTS

Of the 7407 extracted sentences over all documents, a total of 172836 tokens were identified. From these tokens, 131625 were relevant domain tokens covered in triples, representing 76.16% of text coverage into triples. To ensure these triples are viable, the relations binding subject and object tokens had to be either verbal or prepositional to rule out unusable triples. 16051 verbal or prepositional relations were extracted over the text, which resulted in 13821 unique triples representing approximately 10.50% of the total set of extracted triples. This figure signifies that domain concepts make up roughly 10% of a CSR, whereas the remaining 90% are the different context windows in which the domain vocabulary is used. A summary of the triple extraction process from our pipeline is detailed in Table 1. From the extracted triples, 4002 NER objects are domain vocabulary that can be mapped to UMLS concepts. An additional 3417 objects tagger by the NER tagger means a total of 53,67% of the extracted triple objects can be mapped to UMLS concepts. The final KG yields 7151 indirect links from NER to TUI nodes. Indirect links encompass any direct link from NER to CUI ot NER to TUI directly. The calculations from our graph traversal algorithm identify 1533 direct links from NER to TUI, resulting in an alignment of 21,40%. Table 2 shows the details of the NER node alignment to the domain ontology. Table 3 shows the performance of our pipeline with respect to the human baseline. The results show promise for our pipeline: it beats the human baseline on all metrics while retaining a good domain coverage of the text. The mapping score indicates the over half the extracted triples contains pertinent nodes that can be traced back to the domain ontology, showcasing the effectiveness of our annotation and extraction methods. The alignment score, while relatively low, is encouraging when it comes to finding higher-level concepts linked to the initial document text. This opens the possibility to a wider integration between domain ontologies and domain texts, with potential possibilities to enhance the latter with the former using the links between NER and TUI to semi-automatically generate in-context text templates and enrich the document. It is worth noting that the noticeable discrepancy in scores between the metrics suggests issues that need to be addressed at annotation and extraction level. Our pipeline still performs poorly on adjectival relationships, identifying acronyms (e.g., *human arm* versus *ARM*) and specific wordings (e.g., *6 cycle* versus *cycle 6*) which explains the drops in scores between metrics.

Object	Count
Sentences	7407
Tokens in sentences	172836
Tokens covered in triples	131625
Verbal or prepositional relations	16051
Unique triple objects	13821

Table 1: Triple extraction summary.

Object	Count
Unique NER objects linked to UMLS	4002
I2b2 NER objects linked to UMLS	3417
NER to CUI/TUI indirect links	7151
NER to TUI direct links	1533

Table 2: Alignment Summary.

Method	CVRG	MAPG	ALGT
Baseline	68.00	40.00	10.00
Our Pipeline	**76.16**	**53.67**	**21.40**

Table 3: Comparative results of our methodology.

6 CONCLUSION

We introduce a system for domain information abstraction from text and ontology alignment for a more effective KG creation. Our method has the advantage of providing good text-to-triple coverage while maintaining strict semantic consistency for overlapping tokens, which allows better mapping and alignment to higher-order domain ontologies. Our experiments show the need to expand the annotation and extraction processes of our system in order to handle edge cases in unstructured text and capture triples more faithfully. In future work, we will target enhancing the triple extraction process from text by making the annotator more flexible with handling edge cases like acronyms or sentence rewordings. We will integrate features like coreference resolution to capture more fine-grained triples and improve KG construction. we will also aim to evaluate our system against other architectures like LLMs and widen the scope of our experimentation to include other types of biomed-

ical documents (e.g., Protocols) as well as extend it to other domains like finance.

REFERENCES

Abi Akl, H. (2023). The path to autonomous learners. In *Science and Information Conference*, pages 808–830. Springer.

Chaurasiya, D., Surisetty, A., Kumar, N., Singh, A., Dey, V., Malhotra, A., Dhama, G., and Arora, A. (2022). Entity alignment for knowledge graphs: progress, challenges, and empirical studies. *arXiv preprint arXiv:2205.08777*.

Chouham, E. M., Espejel, J. L., Alassan, M. S. Y., Dahhane, W., and Ettifouri, E. H. (2023). Entity identifier: A natural text parsing-based framework for entity relation extraction. *arXiv preprint arXiv:2307.04892*.

Cotovio, P. G., Jimenez-Ruiz, E., and Pesquita, C. (2023). What can knowledge graph alignment gain with neuro-symbolic learning approaches? *arXiv preprint arXiv:2310.07417*.

Dao, N.-M., Hoang, T. V., and Zhang, Z. (2023). A benchmarking study of matching algorithms for knowledge graph entity alignment. *arXiv preprint arXiv:2308.03961*.

Dong, K., Sun, A., Kim, J.-J., and Li, X. (2023). Open information extraction via chunks. *arXiv preprint arXiv:2305.03299*.

Dunn, A., Dagdelen, J., Walker, N., Lee, S., Rosen, A. S., Ceder, G., Persson, K., and Jain, A. (2022). Structured information extraction from complex scientific text with fine-tuned large language models. *arXiv preprint arXiv:2212.05238*.

Fanourakis, N., Efthymiou, V., Kotzinos, D., and Christophides, V. (2023). Knowledge graph embedding methods for entity alignment: experimental review. *Data Mining and Knowledge Discovery*, 37(5):2070–2137.

Friedman, S., Magnusson, I., Sarathy, V., and Schmer-Galunder, S. (2022). From unstructured text to causal knowledge graphs: A transformer-based approach. *arXiv preprint arXiv:2202.11768*.

Hong, Z., Chard, K., and Foster, I. (2024). Combining language and graph models for semi-structured information extraction on the web. *arXiv preprint arXiv:2402.14129*.

Kamp, S., Fayazi, M., Benameur-El, Z., Yu, S., and Dreslinski, R. (2023). Open information extraction: A review of baseline techniques, approaches, and applications. *arXiv preprint arXiv:2310.11644*.

Liu, P., Gao, W., Dong, W., Huang, S., and Zhang, Y. (2022). Open information extraction from 2007 to 2022–a survey. *arXiv preprint arXiv:2208.08690*.

Melnyk, I., Dognin, P., and Das, P. (2022). Knowledge graph generation from text. *arXiv preprint arXiv:2211.10511*.

Mihindukulasooriya, N., Tiwari, S., Enguix, C. F., and Lata, K. (2023). Text2kgbench: A benchmark for ontology-driven knowledge graph generation from text. In *International Semantic Web Conference*, pages 247–265. Springer.

Motger, Q. and Franch, X. (2024). Nlp-based relation extraction methods in re. *arXiv preprint arXiv:2401.12075*.

Pan, J. Z., Razniewski, S., Kalo, J.-C., Singhania, S., Chen, J., Dietze, S., Jabeen, H., Omeliyanenko, J., Zhang, W., Lissandrini, M., et al. (2023a). Large language models and knowledge graphs: Opportunities and challenges, 2023. *arXiv preprint arXiv:2308.06374*.

Pan, S., Luo, L., Wang, Y., Chen, C., Wang, J., and Wu, X. (2023b). Unifying large language models and knowledge graphs: A roadmap, 2023. *arXiv preprint arXiv:2306.08302*.

Peng, C., Xia, F., Naseriparsa, M., and Osborne, F. (2023). Knowledge graphs: Opportunities and challenges. *Artificial Intelligence Review*.

Qian, K., Belyi, A., Wu, F., Khorshidi, S., Nikfarjam, A., Khot, R., Sang, Y., Luna, K., Chu, X., Choi, E., et al. (2023). Open domain knowledge extraction for knowledge graphs. *arXiv preprint arXiv:2312.09424*.

Trisedya, B. D., Salim, F. D., Chan, J., Spina, D., Scholer, F., and Sanderson, M. (2023). i-align: an interpretable knowledge graph alignment model. *Data Mining and Knowledge Discovery*, 37(6):2494–2516.

Van Tong, V., Huynh, T. T., Nguyen, T. T., Yin, H., Nguyen, Q. V. H., and Huynh, Q. T. (2021). Incomplete knowledge graph alignment. *arXiv preprint arXiv:2112.09266*.

Vogt, L., Kuhn, T., and Hoehndorf, R. (2022). Semantic units: Organizing knowledge graphs into semantically meaningful units of representation [internet].

Wadhwa, S., Amir, S., and Wallace, B. C. (2023). Revisiting relation extraction in the era of large language models. *arXiv preprint arXiv:2305.05003*.

Xie, F., Zeng, X., Zhou, B., and Tan, Y. (2023). Improving knowledge graph entity alignment with graph augmentation. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 3–14. Springer.

Yang, L., Chen, H., Wang, X., Yang, J., Wang, F.-Y., and Liu, H. (2024). Two heads are better than one: Integrating knowledge from knowledge graphs and large language models for entity alignment. *arXiv preprint arXiv:2401.16960*.

Yu, B., Zhang, Z., Li, J., Yu, H., Liu, T., Sun, J., Li, Y., and Wang, B. (2022). Towards generalized open information extraction. *arXiv preprint arXiv:2211.15987*.

Zeng, K., Li, C., Hou, L., Li, J., and Feng, L. (2021). A comprehensive survey of entity alignment for knowledge graphs. *AI Open*, 2:1–13.

Zhang, R., Su, Y., Trisedya, B. D., Zhao, X., Yang, M., Cheng, H., and Qi, J. (2023). Autoalign: fully automatic and effective knowledge graph alignment enabled by large language models. *IEEE Transactions on Knowledge and Data Engineering*.

Zhang, R., Trisedy, B. D., Li, M., Jiang, Y., and Qi, J. (2021). A benchmark and comprehensive survey on knowledge graph entity alignment via representation learning. *arXiv preprint arXiv:2103.15059*.