# Automated Knowledge Graph Construction using Large Language Models and Sentence Complexity Modelling

**Sydney Anuyah**[1] **Mehedi Mahmud Kaushik**[1] **Krishna Dwarampudi**[2]
**Rakesh Shiradkar**[3] **Arjan Durresi**[1] **Sunandan Chakraborty**[1]
Luddy School of Informatics, Indiana University, Indianapolis, IN[1]
School of Science Purdue University, Indianapolis, IN[2]
Department of Biomedical Engineering and Informatics, Indiana University, Indianapolis, IN[3]
{sanuyah, mekaush, rshirad, adurresi, sunchak}@iu.edu; sdwaramp@purdue.edu

## Abstract

We introduce CoDe-KG, an open-source, end-to-end pipeline for extracting sentence-level knowledge graphs by combining robust coreference resolution with syntactic sentence decomposition. Using our model, we contribute a dataset of over 150 000 knowledge triples, which is open source. We also contribute a training corpus of 7248 rows for sentence complexity, 190 rows of gold human annotations for co-reference resolution using open source lung-cancer abstracts from PubMed, 900 rows of gold human annotations for sentence conversion policies, and 398 triples of gold human annotations. We systematically select optimal prompt-model pairs across five complexity categories, showing that hybrid chain-of-thought and few-shot prompting yields up to 99.8% exact-match accuracy on sentence simplification. On relation extraction (RE), our pipeline achieves 65.8% macro-F1 on REBEL, an 8-point gain over the prior state of the art, and 75.7% micro-F1 on WebNLG2, while matching or exceeding performance on Wiki-NRE and CaRB. Ablation studies demonstrate that integrating coreference and decomposition increases recall on rare relations by over 20%. Code and dataset are available at https://github.com/KaushikMahmud/CoDe-KG_EMNLP_2025.

## 1 Introduction and Background

One way to represent data is through knowledge graphs (KGs) (Hogan et al., 2021). KGs have transformed the way data is organized and by leveraging complex network chains, we have been able to explore complex fields like causality in different domains (Friedman et al., 2022; MacLean, 2021; Naser, 2022).

With the recent advancements in Natural Language Processing (NLP) using Large Language Models (LLMs), KGs have become instrumental, both as knowledge bases and in finetuning these large models (Pan et al., 2024). One of such advantages is in the creation of domain-specific ontologies (Karim et al., 2023; Chandak et al., 2023) closely associated with creating new reasoning and inference methods (Kau et al., 2024; Zhang et al., 2024).

Previous research has built the foundational concepts of KGs, which include the models used in creating these graphs and their representation (Hogan et al., 2021). Automated KG constructions (Zhong et al., 2023) and representation learning (Ji et al., 2021) have defined major stages in building a KG: from knowledge acquisition and semantic table interpretation (Liu et al., 2023) to entity extraction–covering Named Entity Recognition (NER), Named Entity Disambiguation (NED), and Named Entity Linking (NEL) (Al-Moslmi et al., 2020). These studies and many more provide a system in which unstructured text can be transformed into an organized corpus of interlinked entities. Secondly, domain techniques such as graph knowledge distillation (Tian et al., 2023) and embedding schemes (Cao et al., 2024) have helped reinforce the ability to compress, optimize, and represent KGs which are then utilized in downstream applications. The goal of event KGs (Guan et al., 2022) and explainable artificial intelligence (AI) on KGs (Schramm et al., 2023) is to empathically ensure that models not only have to be efficient but also interpretable (Kaur et al., 2022).

The challenges we are tackling is two fold: (1) We have a large volume of unstructured text data, and because it varies in structure, writing style, and vocabulary across different domains, it has become harder to parse, and (2) Many automated pipelines for KG creation are not really automated, as some are heavily prompt reliant on the end user (Buehler, 2024), others have issues of handling noisy datasets (Zhang et al., 2023). This is why we are researching a one-size-fits-all open-source model framework that could help in knowledge extraction irrespec-



*Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 15515–15539
November 4-9, 2025 

tive of our data. Through the typical structure of the English Language, it is possible to extract relationships through verb usage and clauses. This leads us to the first research question. **RQ1:** Can sentence modelling be used to effectively create KGs that rival other methods? We also compare our method to the popular closed-source AI model: GPT 4 series which is renowned for parsing academic literature, and we design evaluation prompts to benchmark their performance against ours. This can be summarized as **RQ2:** can an open-source, LLM model using the sentence semantics approach reliably construct KGs from raw texts? Our contributions are as follows:

- We introduce a novel sentence-semantic framework for relation extraction (RE) and KG construction, borrowing from linguistic theory and semantic parsing. This idea though common, to the best of our knowledge has been under-explored in mainstream NLP information extraction pipelines. The novelty of our work lies in the integration of multiple frameworks rather than just one task. Our method explicitly models semantic sentence types (e.g., complex (CX), compound (CD), and compound-complex (CC) forms) as the foundation for extracting knowledge triples. Each triples is a simple, three-part structure $(\text{entity}_1, \text{relationship}, \text{entity}_2)$ used to represent a single fact in a KG.

- We also explore diverse prompting strategies across our pipeline, including Chain-of-Thought (CoT) reasoning, Few-Shot In-Context Learning (FICL), and Zero-Shot General Instruction prompting (GIP), and empirically demonstrate their varying contributions to structural decomposition. To support this architecture, we release a suite of open-source resources:

  1. A 7248-row dataset that categorizes and maps diverse sentence semantics aligned to our model's decomposition strategy (complex, compound, compound-complex, simple and incomplete sentence).
  2. A gold-standard co-reference resolution corpus comprising 190 PubMed lung-cancer abstracts annotated by four domain experts.
  3. A 900-sample sentence transformation dataset, consisting of 300 annotated examples each for converting complex, compound, and compound-complex sentences into simple, extractable forms.
  4. A machine-generated KG corpus of over 150,000 structured triples, created using our full end-to-end pipeline.

## 2 Background

### 2.1 Sentence Semantic Modelling for Knowledge Extraction

Sentence semantic modelling involves organizing sentences into various types, which structure how ideas can relate to one another. Let us define a grammar structure as $G = (N, \Sigma, P, S)$ where $N$ is a finite set of non-terminal symbols, $\Sigma$ is a finite set of terminal symbols (the actual words or tokens in the language), $P$ is a finite set of **production rules** that describe how non-terminals can be expanded into sequences of non-terminals and terminals and $S \in N$ is the **start symbol**, which we conventionally call `Sentence`. Appendix A discusses the interplay of sentence and clauses in more detail.

To understand the interplay of clauses and how they make up a sentence, we need to consider the types of sentences in English Language (Das et al., 2018), which are:

- Simple Sentences: Having only one independent clause and no dependent clause

$$S_{\text{simple}} = \{\, (NP,\, VP) \mid NP \in \mathcal{N},\ VP \in \mathcal{V} \}$$

- Complex Sentences: Having one independent clause and at least one dependent clause

$$S_{\text{complex}} = S_{\text{main}} \cup DC$$

- Compound Sentences: Having two or more independent clauses joined by a conjunction and no dependent clause

$$S_{\text{compound}} = S_1 \oplus S_2 \text{ where } \oplus \text{ is a conjunction}$$

- Compound-Complex Sentences: Having two or more independent clauses joined by a conjunction and have at least one dependent clause

$$S_{\text{comp-comp}} = (S_1 \oplus S_2) \cup DC$$

| Method | Sentence Decomp. | Coref Res. | Open-Source | Domain-Agnostic | Eval Scripts |
|---|---|---|---|---|---|
| GraphRAG (Han et al., 2024) | × | × | × | ✓ | × |
| EDC (Zhang and Soh, 2024) | × | × | ✓ | ✓ | ✓ |
| GKG-LLM (Zhang et al., 2025) | × | ✓ | ✓ | ✓ | ✓ |
| Neo4j LLM-KG (Bharti et al., 2024) | × | × | × | ✓ | × |
| KGGen (2025) (Mo et al., 2025) | × | × | ✓ | ✓ | ✓ |
| Our Pipeline | ✓ | ✓ | ✓ | ✓ | ✓ |

Table 1: Current LLM-induced KG Methods Comparison

The core motivation behind this work stems from the assumption that LLMs emulate human reasoning (Wu et al., 2024). Additionally, as shown by Nurmalan (Hendrawati, 2018), human comprehension of sentence structure is far from uniform. Undergraduate students fail to accurately interpret CC sentences in 44.54% of cases, followed by CD (23.2%) and CX (22.13%) sentences. In contrast, error rates drop significantly to 10.13% for simple sentences. Notably, academic and scientific writing rarely employs simple sentences, favouring more elaborate constructions aligned with formal and jargon-heavy discourse. We posit that modelling and converting these complex sentence types into simpler forms enables more effective interpretation by LLMs, particularly for structured information extraction.

A common misconception is that a simple sentence means a simplified or short sentence. However, as Phil (Atteberry, 2016) illustrates, even syntactically rich sentences, such as "Being an English teacher with a penchant for syntactical complexity, I love simple sentences upon getting up and before going to bed"—qualify as simple if they contain only one independent clause. Despite structural simplicity, such sentences may encode multiple relationships, contradicting the assumption that simple sentences yield only one extractable relation. Importantly, a simple sentence can feature compound subjects ("John and Mary run…"), compound predicates ("runs and jumps…"), or compound objects ("an apple and a banana…"). Our framework explicitly models these variations, ensuring RE remains robust across all syntactic permutations of the simple sentence form.

### 2.2 Prompting Strategies

We explore four prompting strategies within our pipeline to evaluate their effectiveness in sentence restructuring and RE: GIP, FICL, CoT, and Hybrid CoT + FICL. GIP relies on general instructions without examples, offering baseline performance and broad generalizability (Ouyang et al., 2022; Kojima et al., 2022). FICL incorporates a small set of in-context examples to guide the model, improving structure-sensitive tasks like sentence decomposition (Brown et al., 2020; Li et al., 2023). COT prompting, which encourages step-by-step reasoning, has proven especially effective in multi-step reasoning and relation-rich generation (Wei et al., 2022; Li et al., 2025). Our hybrid CoT + FICL strategy combines the benefits of example-guided prompting with intermediate reasoning steps, significantly improving accuracy in sentence decomposition and RE. We benchmark each strategy across multiple subtasks in our pipeline and find that hybrid prompting consistently yields the most precise and coherent results, particularly in complex biomedical sentences, which is in tune with recent advances in prompt engineering that emphasize structure-aware and compositional prompting for complex NLP tasks (Kojima et al., 2022).

## 3 Data

### 3.1 PubMed Lung Cancer Abstracts

PubMed is an open biomedical literature repository. From PubMed, we randomly parsed 7,500 abstracts related to the lung cancer keyword, published between 2020 and 2025, to create our primary evaluation corpus. This dataset supports our co-reference resolution, sentence decomposition, and triple extraction tasks. The inclusion principle was any abstract that mentioned lung cancer, was open source and free to use, and we did not particularly exclude any research apart from those that fell outside the random sample. The data was sampled on March 18, 2025. The motivation behind using the lung cancer abstract was linked to a 20-year international study in Radiology (Henschke et al., 2023). We believed that the dataset is well-versed and well-researched.

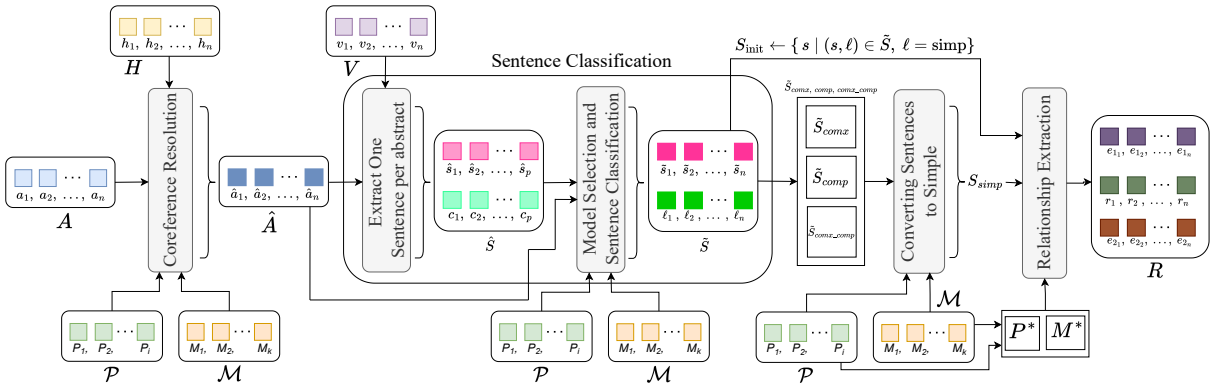


Figure 1: Overview of ***CoDe-KG***, the automated KG creation pipeline. First, the input set of abstracts is given to the **Coreference Resolution stage**. In this phase, a team of annotators, a collection of prompt strategies, and models are jointly applied to produce the coreference-resolved abstract set, which is given as input in the **Sentence Classification stage**. With the help of verifiers, prompting strategies and models, a list of correctly classified sentences with labels is generated in this stage. Then, in the **Converting Sentences to Simple** stage, $\tilde{S}_{comx,\ comp,\ comx\_comp}$, prompt strategies, and models are given as input and converted into simple sentences $\tilde{S}_{simp}$. In **Relationship Extraction** stage, $\tilde{S}_{simp}$, $S_{\text{init}}$ and best model–prompt pair $(P^*, M^*)$ from previous stage are given as input and relationships (entity$_1$, relationship, entity$_2$) are extracted for constructing KG.

### 3.2 REBEL (Cabot and Navigli, 2021)

We adopt the same 1,000-sample subset used in the EDC model (Zhang and Soh, 2024) for evaluation, originally drawn from the REBEL test partition of 105,516 entries, also published in EMNLP.

### 3.3 WebNLG+2020 (Ferreira et al., 2020)

WebNLG+2020 (v3.0) is a semantic parsing benchmark containing text-triple pairs. We use its full test split of 1,165 samples covering 159 unique relation types.

### 3.4 Wiki-NRE (Distiawan et al., 2019)

Wiki-NRE is a distant supervision dataset for RE. We also used the same sample of 1,000 pairs used in the EDC model (Zhang and Soh, 2024). The dataset contains 29,619 entries encompassing 45 distinct relation types .

### 3.5 CaRB (Bhardwaj et al., 2019)

The CaRB dataset is a benchmark for Open Information Extraction (OpenIE), created by re-annotating the original OIE2016 dataset with improved human judgments. The exact number of the final dataset is not known; what was reported in the paper was the devset from Amazon Mechanical Turk, which was 1,282 sentences. However, on the GitHub page, we found 50 unique sentences spanning through 172 lines.

## 4 Methodology

In this research, we propose an automated KG creation pipeline, ***CoDe-KG***, for creating a KG from abstracts. Our approach, as shown in Figure 1, consists of four key stages: doing coreference resolution, sentence classification, sentence conversion, and RE. In this section, we give a detailed overview of our pipeline implementation.

### 4.1 Problem Setup

Let the set of input abstracts be denoted by

$$A = \{a_1, a_2, \ldots, a_n\}, \tag{1}$$

And let the set of valid relation triples extracted from $A$ be denoted by

$$R = \{(e_1, r, e_2) \mid e_1, e_2 \in \mathcal{E}, \; r \in \mathcal{R}, \; (e_1, r, e_2) \text{ is valid}\}. \tag{2}$$

Where $\mathcal{E}$ is the set of all unique entities appearing in those triples and $\mathcal{R}$ is the relation vocabulary. Let the resulting KG be denoted by

$$\mathcal{G} = (\mathcal{E}, \mathcal{R}), \tag{3}$$

Our goal is to construct $\mathcal{G}$ so that it faithfully represents all extracted factual relations across $A$.

### 4.2 Coreference Resolution

In our proposed pipeline, the coreference-resolution stage (as shown in Appendix: Algorithm 1) proceeds by creating a **gold-standard** through expert annotation, and then selecting the optimal **prompt-model combination** for creating coreference-resolved abstracts. First, we draw a random subset of size $s$:

$$A' = \text{UniformSample}(A,\, s).$$

For four expert annotators–two with biological expertise and two with linguistic expertise–working in pairs to resolve coreference on each $a \in A'$, producing

$$R_j(a) = f_{\text{ann}}(h_j,\, a), \quad j = 1, 2.$$

Here, $f_{\text{ann}}$ apply annotator $h_j$'s annotation procedure to abstract $a$, yielding the resolution $R_j(a)$.

We then define the gold set of abstracts $A$, which can be unanimously annotated:

$$G = \big\{\, a \in A' \;\big|\; R_j(a) = R_k(a)\ \forall\, j, k \big\}.$$

And extract the corresponding annotated gold standard

$$G' = \big\{ R_j(a) \;\big|\; a \in G \big\},$$

where any $R_j(a)$ may be used since all agree.

Next, we exhaustively evaluate each prompt-model pair $(P, M) \in \mathcal{P} \times \mathcal{M}$ by generating predicted annotations on the gold inputs

$$\hat{R}_{P,M} = f_{\text{prompt}}\big(P,\, M,\, G\big),$$

and computing a score (e.g., $\text{F}_1$) against the gold-standard pairs:

$$S_{P,M} = \text{score}\big(\hat{R}_{P,M},\, G'\big).$$

We then select the best pair by

$$(P^*, M^*) = \text{argmax}_{P \in \mathcal{P},\, M \in \mathcal{M}} S_{P,M}.$$

Finally, this optimal configuration is applied to the full collection:

$$\hat{A} = f_{\text{prompt}}\big(P^*,\, M^*,\, A\big),$$

yielding fully resolved co-reference annotations $\hat{A}$.

This design ensures that (i) human expertise defines a robust gold standard through unanimous agreement, (ii) prompt-model selection is systematic and exhaustive, and (iii) large-scale annotation inherits the reliability established in the gold-standard phase.

### 4.3 Sentence Classification

The first step of the sentence classification stage (as shown in Appendix: Algorithm 2) processes the resolved abstracts $\hat{A}$ by sampling per category, extracting one representative sentence from each sampled abstract, and keeping only those sentences on which two expert verifiers agree. For the five complexity categories

$$\mathcal{C} = \{\text{simp}, \text{comx}, \text{comp}, \text{comx\_comp}, \text{incomp}\},$$

we draw a random subset

$$A_c = \text{UniformSample}(\hat{A},\, p_c), \quad c \in \mathcal{C},$$

so that $|A_c| = p_c$. From each $a \in A_c$ we then extract exactly one representative sentence:

$$s = \begin{cases} f_{\text{create}}(\text{annotator},\, a), & c \in \{\text{simp}, \text{incomp}\}, \\ f_{\text{choose}}(a), & \text{otherwise.} \end{cases}$$

We aggregate all candidates into

$$S_{\text{all}} = \bigcup_{c \in \mathcal{C}} S_c.$$

Next, two expert verifiers $v_1, v_2$ independently review every $s \in S_{\text{all}}$, and we keep only those sentence-category pairs on which they agree:

$$\hat{S} = \left\{ (s, c) \;\middle|\; \begin{array}{l} c \in \mathcal{C},\ s \in S_c, \\ f_{\text{ver}}(v_1, s) = f_{\text{ver}}(v_2, s) \end{array} \right\}.$$

The output $\hat{S}$ is thus a high-agreement, category-labeled sentence set, ready to serve as the input for Step 2.

The second step of the sentence classification stage (as shown in Appendix: Algorithm 3) takes the verified sentence-category set $\hat{S}$ along with the set of prompting strategies $\mathcal{P}$, and models $\mathcal{M}$ as input to produce a fully labeled sentence corpus $\tilde{S}$. First, we formed the training dataset

$$D = \{(s_i, y_i) \mid (s_i, y_i) \in \hat{S}\}.$$

For each $m \in \mathcal{M}$, train on $D_{\text{train}}$ and we computed its validation score

$$\text{score}_m = \text{Evaluate}(m,\, D_{\text{val}}).$$

Then we selected the best model

$$m^* = \text{argmax}_{m \in \mathcal{M}} \text{score}_m.$$

Finally, we applied $m^*$ to every sentence in the fully coreference-resolved abstract set $\hat{A}$:

$$\tilde{S} = \big\{(s, \ell_s) \mid s \in \text{Sentences}(\hat{A}),\ \ell_s = m^*(s)\big\}.$$

The resulting $\tilde{S}$ is the complete collection of sentence-label pairs for downstream tasks.

### 4.4 Converting Sentences to Simple

The approach of this stage (as shown in Appendix: Algorithm 4) consists of prompt-model selection for each category, and large-scale sentence simplification using the selected configurations.

For each category $c$, we hold out the set $S_c$ of complex (`comx`), compound (`comp`), and complex-compound (`comx_comp`) sentences, and exhaustively evaluate every prompt-model combination $(P, M) \in \mathcal{P} \times \mathcal{M}$. We compute a performance score via

$$\text{score}_{P,M}(c) = \text{EvaluatePromptModel}(P, M, S_c).$$

and choose

$$(P_c^*,\ M_c^*) = \text{argmax}_{(P,M)}\text{score}_{P,M}(c).$$

With $(P_c^*, M_c^*)$ fixed for each category, we process every sentence $s \in S_c$ by invoking:

$$\hat{s} = f_{\text{prompt}}\big(P_c^*,\ M_c^*,\ s\big), \quad S_{\text{simp}} \leftarrow S_{\text{simp}} \cup \{\hat{s}\}.$$

Once all categories are processed, $S_{\text{simp}}$ constitutes the collection of simplified sentences from complex, compound, and complex-compound sentences.

### 4.5 Relationship Extraction

In this stage, we implement RE (as shown in Appendix: Algorithm 5) through **sentence consolidation** and **triple generation**. First, we form the working sentence set by combining:

$$S_{\text{init}} = \{\, s \mid (s, \ell) \in \tilde{S},\ \ell = \text{simp}\},$$
$$S = S_{\text{simp}} \ \cup\ S_{\text{init}}.$$

Here, $S_{\text{simp}}$ is the set of all sentences produced by the simplification stage, while $S_{\text{init}}$ contains those initially classified as simple. Next, for each $s \in S$, we extract a candidate relation triple via the $f_{\text{rel}}(s)$ function, where the best prompt and model combination $(P^*,\ M^*)$ from the previous stage was used. Here,

$$(e_1,\ r,\ e_2)\ =\ f_{\text{rel}}(P^*, M^*, s).$$

We collect only non-empty outputs:

$$R\ \leftarrow\ R \cup \{(e_1, r, e_2)\} \quad \text{if} \quad (e_1, r, e_2) \neq \varnothing.$$

Upon completion, $R$ holds all valid $(\text{entity}_1, \text{relation}, \text{entity}_2)$ triples. Finally, we assemble the extracted triples into our knowledge graph. Let

$$\mathcal{E}\ =\ \bigcup_{(e_1,r,e_2)\in R} \{e_1, e_2\}, \quad \mathcal{R}\ =\ \bigcup_{(e_1,r,e_2)\in R} \{r\},$$

and define the graph as

$$\mathcal{G}\ =\ (\mathcal{E},\ \mathcal{R}).$$

Each triple $(e_1, r, e_2) \in R$ becomes a directed, labeled edge from node $e_1$ to node $e_2$. This knowledge graph $\mathcal{G}$ now encodes all valid factual relations extracted across the corpus and can be used for downstream querying and inference.

## 5 Experiments

### 5.1 Experiment 1: Results of Co-reference Resolution

Biomedical text is harder to understand and therefore, RE is perceived to be harder (Johnson and Bernstam, 2023). Therefore, we construct our benchmark dataset of 190 coreference abstracts in biomedical literature on Lung Cancer to evaluate the performance of LLM. Therefore, we crafted the SOTA prompts discussed in chapter 2.2. The prompts we finally used were the COT+FICL prompt after experimenting on the different prompts, which are in Appendix E.1. We then randomly sampled 190 abstracts from the full set of 7,500. Each abstract was independently annotated by two domain experts and two language experts. After the initial pass, the experts exchanged annotations and discussed any discrepancies. Full annotation information available in Appendix E.2.

We evaluated several LLMs on our benchmark. Results were poor for most models. Predictions were scored using MUC, $B^3$, $CEAF_4$, and the CoNLL $F_1$ aggregate (Pradhan et al., 2012). Deepseek-distill-Qwen-7B, Qwen-Chat-7B, and Qwen-7B scored 0% $F_1$. Deepseek-7B, Deepseek-6.7B, and Deepseek-Prover-7B scored below 2% $F_1$. Deepseek-distill-Llama-8B scored below 10% $F_1$ in all categories. Table 2 lists $F_1$ scores for the models doing co-reference resolution and was benchmarked with ChatGPT o4-mini-high and ChatGPT-4.5 responses as a baseline. ChatGPT o4-mini-high did the best overall with an $F_1$ of approximately 63% using the FICL prompt. We bolded the best open-source model, which was comparable to the closed source models for this task.

The evaluation of the co-reference was done using a cosine similarity score of 0.9, because we

| Model | Prompt | MUC (%) | $B^3$ (%) | CEAF (%) | CoNLL (%) |
|---|---|---|---|---|---|
| **Mixtral-8x7B-Instruct-v0.1** | FICL | 32.42 | 70.61 | 70.61 | **57.88** |
| Llama-3.1-8B-Instruct | FICL | 27.16 | 69.57 | 69.57 | 55.43 |
| Llama-3.2-3B-Instruct | COT_FICL | 16.98 | 70.7 | 70.7 | 52.79 |
| Llama-3.3-70B-Instruct | FICL | 31.25 | 70.94 | 70.94 | 57.71 |
| Mistral-7B-Instruct-v0.3 | COT_FICL | 18.58 | 70.74 | 70.74 | 53.35 |

Table 2: Comparison of $F_1$ Scores by Models.

noticed that 99% of all values that were marked at 0.9 correctly but not exact match were actually correct, just differing in preposition. For instance, the gold standard says "a house", and the model says "house". At 0.8% the values were not significant to be considered, therefore, we stuck to using a cosine similarity score of 0.9 for the co-reference evaluation.

### 5.2 Experiment 2: Syntactic Sentence Classification

We created a dataset for classification, and all details are given in Appendix E.3. We fine-tuned six transformer-based models and two smaller LLMs on the training set and evaluated them on the test set. Table 3 reports test accuracy and macro-averaged $F_1$ for each model. We evaluated the entire test set on a GPT-4o model.

Table 3: Sentence-type classification results (train set: 2,00- sentences test set: 5,269 sentences)

| Model | Accuracy | $F_{1\text{macro}}$ |
|---|---|---|
| BERT | 87.25% | 86.14% |
| BERT-Large | **87.68%** | **86.69%** |
| BioBERT | 87.19% | 86.21% |
| BioBERT-Large | 85.97% | 85.16% |
| ClinicalBERT | 86.92% | 85.87% |
| RoBERTa | 87.13% | 85.83% |
| Gemma3 1-B | 9.24% | 4.15 % |
| LLama3.2 1-B | 17.71 % | 0.27 % |
| **GPT 4-0** | 80.30 % | 76.14% |
| Random Guessing Lowest | 8.83 % | 3.24% |
| Random Guessing Highest | 32.61 % | 9.84 % |

### 5.3 Experiment 3: Evaluating the Prompting Strategies and Semantic Conversion

We created a systematic structure (see Appendices F.1, F.2, and F.3 for the systemic conversion process) of evaluating how a model would convert a cx, cd or cc sentence to a simple one, thereby, evaluating the four prompting strategies–GIP, FICL, COT, and COT+FICL prompts (see Appendix F.4). From the 65,175 sentences extracted from 7,500 abstracts, our classifier labelled 42,282 as complex, 4,942 as compound, 2,366 as compound-complex, 13,465 as simple, and 2,120 as incomplete. We then randomly sampled 300 sentences each from the complex, compound, and compound-complex classes and translated them into simple sentences. These sample sizes at 300 achieve 95% confidence for their respective populations with margins of error of $\pm 5.62\%$, $\pm 5.57\%$, and $\pm 5.26\%$, respectively. We tested on the top performing models and evaluated their performance on the different prompting strategies. The results are shown in Table 10, with the hybrid prompt performing the best in all cases.

Table 4: Model performance on the conversion of *Compound to Simple Sentences*

| Model | Macro Avg. | Exact-Match | RMSE |
|---|---|---|---|
| DeepSeek-LLM-67B | 15.56% | 14.67% | 1.5891 |
| DeepSeek-LLM-7B | 55.83% | 50.00% | 1.1506 |
| DeepSeek-R1-Distill-Llama-8B | 68.04% | 65.00% | 1.0571 |
| DeepSeek-Prover-V1.5-7B | 81.38% | 76.33% | 0.8591 |
| Llama-3.3-70B | 95.30% | 81.00% | 0.3213 |
| **Llama-3-8B** | **99.78%** | **98.00%** | **0.1078** |
| Mistral-7B-Instruct-v0.3 | 96.64% | 90.33% | 0.2356 |
| Mixtral-8x7B-Instruct-v0.1 | 96.14% | 91.67% | 0.2323 |
| Qwen-7B | 56.17% | 54.00% | 1.2339 |
| Qwen-7B-Chat | 1.33% | 1.33% | 1.7193 |
| *GPT 4 o* | *91.68%* | *77.67%* | *0.4204* |
| *GPT 4 o-3* | *87.69%* | *68.67%* | *0.5924* |

Table 5: Model performance on the conversion of *Complex to Simple Sentences*

| Model | Macro Avg. | Exact-Match | RMSE |
|---|---|---|---|
| DeepSeek-LLM-67B | 26.23% | 22.00% | 1.7562 |
| DeepSeek-LLM-7B | 63.30% | 62.33% | 1.1866 |
| DeepSeek-R1-Distill-Llama-8B | 43.78% | 33.00% | 1.5540 |
| DeepSeek-Prover-V1.5-7B | 91.33% | 65.67% | 0.8841 |
| **Llama-3.3-70B** | **99.59%** | **98.67%** | **0.1364** |
| Llama-3-8B | 97.17% | 92.67% | 0.2866 |
| Mistral-7B-Instruct-v0.3 | 93.73% | 81.00% | 0.3114 |
| Mixtral-8x7B-Instruct-v0.1 | 98.48% | 94.67% | 0.1533 |
| Qwen 7B | 64.61% | 64.00% | 1.1987 |
| Qwen-7B-Chat | 9.24% | 7.67% | 1.8699 |
| *GPT 4 o* | *96.72%* | *91.33%* | *0.3050* |
| *GPT 4 o-3* | *99.05%* | *97.00%* | *0.1714* |

Table 6: Model performance on the conversion of *Compound-Complex to Simple Sentences*

| Model | Macro Avg. | Exact-Match | RMSE |
|---|---|---|---|
| DeepSeek-LLM-67B | 28.42% | 11.00% | 1.6178 |
| DeepSeek-LLM-7B | 49.45% | 32.33% | 1.2848 |
| DeepSeek-R1-Distill-Llama-8B | 37.25% | 21.67% | 1.5183 |
| DeepSeek-Prover-V1.5-7B | 71.12% | 51.00% | 0.8411 |
| Llama-3.3-70B | 89.57% | 72.67% | 0.4836 |
| Llama-3-8B | 91.71% | 78.00% | 0.4544 |
| Mistral-7B-Instruct-v0.3 | 91.19% | 80.00% | 0.3273 |
| **Mixtral-8x7B-Instruct-v0.1** | **92.16%** | 76.67% | **0.2727** |
| Qwen 7B | 56.68% | 40.33% | 1.1977 |
| Qwen-7B-Chat | 15.81% | 4.33% | 1.7556 |
| *GPT 4 o* | *82.75%* | *68.33%* | *0.5354* |
| *GPT 4 o-3* | *94.10%* | ***81.67%*** | *0.2320* |

### 5.4 Extracting Relationship Pairs from Simple Sentences

With the total number of simple sentences exceeding 177,000, we used a carefully crafted COT + FICL prompt as it has shown from data to perform the best. The annotators AB and CD studied 100 sentences generated from each model that were tested below and came to an agreement that the Mixtral-8x7B-Instruct-v0.1 model performed well with a 99% accuracy in parsing relationships from simple sentences, which also involved capturing multiple relationships in text. It was a Boolean task. The table for the task is Table 9 located in Appendix D.

## 6 Discussion

We evaluated the results on a subset of 200 samples each from Rebel, Web-NLG, and Wiki-NRE obtained from the GitHub page of (Zhang and Soh, 2024). We also evaluated the 50 unique sentences from the CaRB dataset (Bhardwaj et al., 2019).

Table 7: Evaluation metrics across benchmarks

| Metrics | ReBEL | Web-NLG2 | Wiki-NRE | CaRB |
|---|---|---|---|---|
| Exact-Match | 14.50% | 43.00% | 7.00% | 43.14% |
| Prec Macro | 72.97% | 80.70% | 60.26% | 66.96% |
| Rec Macro | 59.88% | 71.64% | 60.13% | 68.43% |
| F1-Score Macro | 65.78% | 75.90% | 60.20% | 62.61% |
| Prec Micro | 66.34% | 79.35% | 59.02% | 63.69% |
| Rec Micro | 59.88% | 72.32% | 58.67% | 59.52% |
| F1-Score Micro | 62.94% | 75.67% | 58.84% | 61.54% |
| RMSE | 0.8813 | 0.5785 | 0.9648 | 0.3633 |

The results show that even with the extra aid, LLMs still have difficulty parsing relationships the way humans can. Comparing to current SOTA techniques, for the ReBEL benchmark, our model achieves a macro-F1 of 65.78% and a micro-F1 of 62.94%, surpassing the published macro-F1 of 51.0% (Cabot and Navigli, 2021) but falling short of the SOTA micro-F1 of 74.0% (Cabot and Navigli, 2021), which tells us that our pipeline can handle diverse, low-frequency relations reasonably well (macro), yet would not perform the best on the most common triplets that dominate micro averaging. The very low exact-match rate (14.5%) are due to the sentence decomposition. In Web-NLG2 and Wiki-NRE RE, we record a micro-F1 of 75.67% and 58.84% respectively, which falls largely behind the SOTA at 93.6% (Ferreira et al., 2020) and GenIE's 91.48% (Distiawan et al., 2019) respectively. However, knowing that this generative task without any fine-tuning shows opportunities for great improvement. In the CaRB OpenIE benchmark, we achieve micro-F1 61.54% and macro-F1 62.61%, compared to DetIE's SOTA 67.7% (Bhardwaj et al., 2019).

Though the results were not as high as we expected, they look promising for an unsupervised approach. Future work would look into fine-tuning LLMs for specific tasks like these. To justify the need for our pipeline, we performed an ablation study on 23 articles already in our coreferenced set. Table 11, Appendix D, shows that removing coreference resolution or sentence-level decomposition hurts our performance sharply, and without these decomposition, our recall falls below half, which explains to us that biomedical sentences have a lot of nuanced co-referent names that must be split before extraction. Therefore, having this pipeline is a reliable way to use LLMs for generated content. We have baselines like ChatGPT-4.5, which do have near-perfect precision. They often summarize relationships and do not give the user the ability to pick and choose, as it makes the decision on the user's behalf. The DeepSeek-405B model performs well too, as it has a higher recall, but introduces a lot of errors. It is interesting to see how our pipeline maintained high precision and recall, but extracted some relationships beyond human reference. The results were hand-graded by humans using the gold standard.

### 6.1 Error Analysis

From the analysis, we classified the errors into 3 types, namely: "missing", "spurious", and "relation-mismatch". Missing error happens when a relationship type is not considered. Spurious is non-existent but invented relationships and relation mismatch occurs when there is a switch between Entity 1 and Entity 2 or the relationship actually exists with another Entity (Entity3), but the model misclassified it. Furthermore, in the Rebel dataset,

the relation-extraction is consistent, as it only contains four variables across the dataset. We noticed that the pipeline parses some "less important" or "missed relationships", which we categorized as spurious in this evaluation. During the evaluation, if a model outputs something not in the gold, but is a valid relationship from the text, the model is not given a score (neither penalized nor awarded).

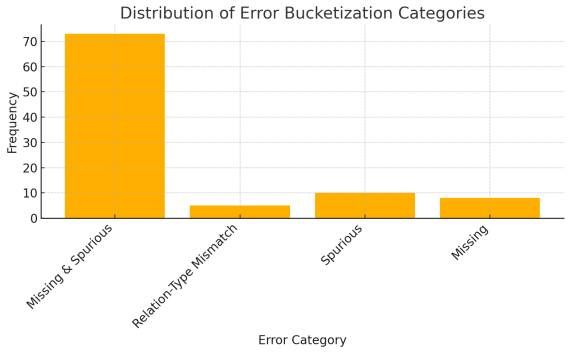


Figure 2: Distribution of error bucketization categories.

However, if it outputs something not in the gold, and cannot be inferred from the text, then we penalize the model. We also noticed a few relation-type mismatches, where the model defaults to a more simplistic relationship type, like "is", "was", rather than focusing on the actual relationship. For example, the model used a generic "is" for party membership instead of "member of political party", in the sentence "Anju Dhillon (born 1979) is a Canadian Liberal politician, who was elected to represent the riding of Dorval-Lachine-LaSalle in the House of Commons of Canada in the 2015 federal election.". Missing and Spurious relationships amongst the errors were profound, as the model typically highlights 3 out of 4 of the relationship types in REBEL and typically omits one. A combination of spurious and missing relationships characterized 73% of the errors, while spurious relationships alone scored 11%, and missing relationships 9% and relation-type is 7%. Interestingly, more often than not, it highlights newer relationships that we were not considering.

## 7 Conclusion

In this work, we created a pipeline and verified that using sentence decomposition on open-source models actually helps the model think through each problem uniquely. By releasing our implementation, annotated datasets, and evaluation scripts, we aim to promote reproducibility and accelerate future work in information extraction and co reference resolution. The annotated datasets from humans include 190 abstract texts, 7248 rows for sentence classification, and 900 rows for sentence decomposition.

## Limitations

The limitations of our paper are as follows:

- Before, our pipeline would be of industry-standard, it is imperative that we find a solution to coreference resolution and its weaknesses. Future work would look into how to improve coreference resolution in thick abstract texts.
- Using just open-source models, though comparative with ChatGPT tends to fail at times, and thereby, average the RE score. Future iterations of the work would employ agent systems that can scan for missing or inconsistent data through a feedback loop and judge the quality to improve the output.
- We only focused on prompting this time; future iterations would look at fine-tuning strategies like Parameter-Efficient Fine-tuning (PEFT) that could help the LLM perform better in weak tasks.
- We had a lot of human annotations for quality. Future work would engage more code scripts, so we can evaluate a wider range of output.
- We acknowledge that our pipeline is costly computationally, and as such, it might not be possible to run with low-level resources, and though the CoT+FICL prompting did well, it might incur a non-trivial inference time and token consumption. However, since this is a one-off knowledge graph, it is still deployable.
- Our evaluation is confined to scientific and benchmark datasets. We have not tested the pipeline on domains such as newswire, legal text, or social media. Thus, its generalizability to noisier or less formal text remains unverified.

## Acknowledgements

We thank the expert annotators for their work. The biologist: Thelma Emeji and Olaniyi Glory; and the linguist: Oluwafaratunmi Olakanmi, and Godman

Adebayo. We also thank the rest of the other annotators who did not want their names mentioned. All instructions given to the annotators were the same as the prompt. Each annotator was paid weekly at a rate of 40,000 Nigerian Naira, over a period of 8 weeks, and this project was self-funded.

## References


Tareq Al-Moslmi, Marc Gallofré Ocaña, Andreas L Opdahl, and Csaba Veres. 2020. Named entity extraction for knowledge graphs: A literature overview. *IEEE Access*, 8:32862–32881.

Phil. Atteberry. 2016. Sentence types: Simple, compound, complex, and compound-complex sentences. https://sites.pitt.edu/~atteberr/comp/0150/grammar/sentencetypes.html. Accessed: March 16, 2025.

Sangnie Bhardwaj, Samarth Aggarwal, and Mausam. 2019. CaRB: A crowdsourced benchmark for open IE. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6262–6267, Hong Kong, China. Association for Computational Linguistics.

Suman Bharti, Dan Chia-Tien Lo, and Yong Shi. 2024. Enhancing contextual understanding in knowledge graphs: Integration of quantum natural language processing with neo4j llm knowledge graph. In *2024 IEEE International Conference on Big Data (BigData)*, pages 8628–8630. IEEE.

Stephen H Bradley, Nathaniel Luke Fielding Hatton, Rehima Aslam, Bobby Bhartia, Matthew EJ Callister, Martyn PT Kennedy, Luke TA Mounce, Bethany Shinkins, William T Hamilton, and Richard D Neal. 2021. Estimating lung cancer risk from chest x-ray and symptoms: a prospective cohort study. *British Journal of General Practice*, 71(705):e280–e286.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Markus J Buehler. 2024. Accelerating scientific discovery with generative knowledge extraction, graph-based representation, and multimodal intelligent graph reasoning. *Machine Learning: Science and Technology*, 5(3):035083.

Pere-Lluís Huguet Cabot and Roberto Navigli. 2021. Rebel: Relation extraction by end-to-end language generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2370–2381.

Jiahang Cao, Jinyuan Fang, Zaiqiao Meng, and Shangsong Liang. 2024. Knowledge graph embedding: A survey from the perspective of representation spaces. *ACM Computing Surveys*, 56(6):1–42.

Payal Chandak, Kexin Huang, and Marinka Zitnik. 2023. Building a knowledge graph to enable precision medicine. *Scientific Data*, 10(1):67.

Bidyut Das, Mukta Majumder, and Santanu Phadikar. 2018. A novel system for generating simple sentences from complex and compound sentences. *International Journal of Modern Education and Computer Science*, 11(1):57–64.

Bayu Distiawan, Gerhard Weikum, Jianzhong Qi, and Rui Zhang. 2019. Neural relation extraction for knowledge base enrichment. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 229–240.

Thiago Castro Ferreira, Claire Gardent, Nikolai Ilinykh, Chris Van Der Lee, Simon Mille, Diego Moussallem, and Anastasia Shimorina. 2020. The 2020 bilingual, bi-directional webnlg+ shared task overview and evaluation results (webnlg+ 2020). In *Proceedings of the 3rd International Workshop on Natural Language Generation from the Semantic Web (WebNLG+)*.

Scott Friedman, Ian Magnusson, Vasanth Sarathy, and Sonja Schmer-Galunder. 2022. From unstructured text to causal knowledge graphs: A transformer-based approach. *arXiv preprint arXiv:2202.11768*.

Saiping Guan, Xueqi Cheng, Long Bai, Fujun Zhang, Zixuan Li, Yutao Zeng, Xiaolong Jin, and Jiafeng Guo. 2022. What is event knowledge graph: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 35(7):7569–7589.

Haoyu Han, Yu Wang, Harry Shomer, Kai Guo, Jiayuan Ding, Yongjia Lei, Mahantesh Halappanavar, Ryan A Rossi, Subhabrata Mukherjee, Xianfeng Tang, et al. 2024. Retrieval-augmented generation with graphs (graphrag). *arXiv preprint arXiv:2501.00309*.

Nurmala Hendrawati. 2018. An analysis on students' errors in writing sentence patterns. *Loquen: English Studies Journal*, 11(1):63–85.

Claudia I Henschke, Rowena Yip, Dorith Shaham, Steven Markowitz, José Cervera Deval, Javier J Zulueta, Luis M Seijo, Cheryl Aylesworth, Karl Klingler, Shahriyour Andaz, et al. 2023. A 20-year follow-up of the international early lung cancer action program (i-elcap). *Radiology*, 309(2):e231988.

Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia d'Amato, Gerard De Melo, Claudio Gutierrez, Sabrina Kirrane, José Emilio Labra Gayo, Roberto Navigli, Sebastian Neumaier, et al. 2021. Knowledge graphs. *ACM Computing Surveys (Csur)*, 54(4):1–37.

Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and S Yu Philip. 2021. A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE transactions on neural networks and learning systems*, 33(2):494–514.

Todd R Johnson and Elmer V Bernstam. 2023. Why is biomedical informatics hard? a fundamental framework. *Journal of Biomedical Informatics*, 140:104327.

Md Rezaul Karim, Lina Molinas Comet, Md Shajalal, Oya Deniz Beyan, Dietrich Rebholz-Schuhmann, and Stefan Decker. 2023. From large language models to knowledge graphs for biomarker discovery in cancer. *arXiv preprint arXiv:2310.08365*.

Amanda Kau, Xuzeng He, Aishwarya Nambissan, Aland Astudillo, Hui Yin, and Amir Aryani. 2024. Combining knowledge graphs and large language models. *arXiv preprint arXiv:2407.06564*.

Davinder Kaur, Suleyman Uslu, Kaley J Rittichier, and Arjan Durresi. 2022. Trustworthy artificial intelligence: a review. *ACM computing surveys (CSUR)*, 55(2):1–38.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

Jia Li, Ge Li, Yongmin Li, and Zhi Jin. 2025. Structured chain-of-thought prompting for code generation. *ACM Transactions on Software Engineering and Methodology*, 34(2):1–23.

Tianle Li, Xueguang Ma, Alex Zhuang, Yu Gu, Yu Su, and Wenhu Chen. 2023. Few-shot in-context learning for knowledge base question answering. *arXiv preprint arXiv:2305.01750*.

Jixiong Liu, Yoan Chabot, Raphaël Troncy, Viet-Phi Huynh, Thomas Labbé, and Pierre Monnin. 2023. From tabular data to knowledge graphs: A survey of semantic table interpretation tasks and methods. *Journal of Web Semantics*, 76:100761.

Finlay MacLean. 2021. Knowledge graphs and their applications in drug discovery. *Expert opinion on drug discovery*, 16(9):1057–1069.

Belinda Mo, Kyssen Yu, Joshua Kazdan, Proud Mpala, Lisa Yu, Chris Cundy, Charilaos Kanatsoulis, and Sanmi Koyejo. 2025. Kggen: Extracting knowledge graphs from plain text with language models. *arXiv preprint arXiv:2502.09956*.

MZ Naser. 2022. Causality, causal discovery, and causal inference in structural engineering. *arXiv preprint arXiv:2204.01543*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Shirui Pan, Linhao Luo, Yufei Wang, Chen Chen, Jiapu Wang, and Xindong Wu. 2024. Unifying large language models and knowledge graphs: A roadmap. *IEEE Transactions on Knowledge and Data Engineering*, 36(7):3580–3599.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang. 2012. Conll-2012 shared task: Modeling multilingual unrestricted coreference in ontonotes. In *Joint conference on EMNLP and CoNLL-shared task*, pages 1–40.

Simon Schramm, Christoph Wehner, and Ute Schmid. 2023. Comprehensible artificial intelligence on knowledge graphs: A survey. *Journal of Web Semantics*, 79:100806.

Yijun Tian, Shichao Pei, Xiangliang Zhang, Chuxu Zhang, and Nitesh Chawla. 2023. Knowledge distillation on graphs: A survey. *ACM Computing Surveys*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Tianhao Wu, Janice Lan, Weizhe Yuan, Jiantao Jiao, Jason Weston, and Sainbayar Sukhbaatar. 2024. Thinking llms: General instruction following with thought generation. *arXiv preprint arXiv:2410.10630*.

Bowen Zhang and Harold Soh. 2024. Extract, define, canonicalize: An llm-based framework for knowledge graph construction. *arXiv preprint arXiv:2404.03868*.

Hanrong Zhang, Xinyue Wang, Jiabao Pan, and Hongwei Wang. 2023. Saka: an intelligent platform for semi-automated knowledge graph construction and application. *Service Oriented Computing and Applications*, 17(3):201–212.

Jian Zhang, Bifan Wei, Shihao Qi, Jun Liu, Qika Lin, et al. 2025. Gkg-llm: A unified framework for generalized knowledge graph construction. *arXiv preprint arXiv:2503.11227*.

Wen Zhang, Jiaoyan Chen, Juan Li, Zezhong Xu, Jeff Z Pan, and Huajun Chen. 2024. Knowledge graph reasoning with logics and embeddings: Survey and perspective. In *2024 IEEE International Conference on Knowledge Graph (ICKG)*, pages 492–499. IEEE.

Lingfeng Zhong, Jia Wu, Qian Li, Hao Peng, and Xindong Wu. 2023. A comprehensive survey on automatic knowledge graph construction. *ACM Computing Surveys*, 56(4):1–62.

## A Sentence Steps

Sentences are made up of clauses, phrases, words, and punctuation. Phrases are a group of words that act as a single unit but do not have both a subject and a predicate. The common types are noun phrase $NP$, verb phrase $VP$, and prepositional phrases $PP$. A sentence in English is defined as: $Sentence \rightarrow NP, VP$. This means every sentence must have a noun phrase, typically including the subject and the verb. Not all sentences have a predicate or an object; however, this is very uncommon in academic writing. Noun phrases are defined as: $NP \rightarrow$ |Det $N$ | Det Adj $N$ | Pronouns |Proper Nouns|, while verb phrases are defined as: $VP \rightarrow |V|VNP|VNPPP|$, where $Det$ is a determiner: for example, "a", "the", "an", "those", etc., $Adj$ is an adjective, and $V$ is a verb. Clauses are typically of two types: independent clauses (IC) and dependent clauses (DC). An IC can stand alone as a complete sentence, while the DC relies on an IC. Dependent clauses start with subordinating conjunctions like “because”, “although”, “when”, “if”, etc., and are then followed by an IC to make a complete sentence. $Clause \rightarrow NP, VP$. Mathematically, it is represented by $DC \rightarrow Subconj\ IC$.

## B Algorithms

**Algorithm 1** Coreference Resolution

**Input**: abstracts $A$, subset size $s$, annotators $H$, prompt strategies $\mathcal{P}$, models $\mathcal{M}$
**Output**: resolved abstracts $\hat{A}$

$A' \leftarrow \mathrm{UniformSample}(A,\ s)$
▷ Select $s$ abstracts at random
**for all** $a \in A'$ **do**
  **for all** $h_j \in H$ **do**
    $R_j(a) \leftarrow f_{\mathrm{ann}}(h_j,\ a)$
▷ Annotator $h_j$ resolves coreference on $a$
$G \leftarrow \{\, a \in A' \mid R_j(a) = R_k(a)\ \forall\, j, k\}$
▷ Gold set of unanimous abstracts
$G' \leftarrow \{R_j(a) \mid a \in G\}$
▷ Gold-standard annotated abstracts
$(P^*, M^*) \leftarrow (\varnothing, \varnothing)$; $\mathrm{bestScore} \leftarrow -\infty$
**for all** $P \in \mathcal{P}$ **do**
  **for all** $M \in \mathcal{M}$ **do**
    $\hat{R}_{P,M} \leftarrow f_{\mathrm{prompt}}(P,\ M,\ G)$
▷ Predict annotations on $G$
    $S_{P,M} \leftarrow \mathrm{score}(\hat{R}_{P,M},\ G')$
▷ Evaluate predictions against $G'$
    **if** $S_{P,M} > \mathrm{bestScore}$ **then**
      $(P^*, M^*) \leftarrow (P, M)$
      $\mathrm{bestScore} \leftarrow S_{P,M}$
▷ Update best prompt-model pair
$\hat{A} \leftarrow f_{\mathrm{prompt}}(P^*,\ M^*,\ A)$
▷ Resolve coreference on full collection
**return** $\hat{A}$

**Algorithm 2** Step 1: Sample Abstracts, Extract One Sentence, and Verify

**Input:** resolved abstracts $\hat{A}$, sample sizes $p_{\text{simp}}, p_{\text{comx}}, p_{\text{comp}}, p_{\text{comx\_comp}}, p_{\text{incomp}}$, expert verifiers $V = \{v_1, v_2\}$
**Output:** verified sentences with category labels $\hat{S}$
**for all** category $c$ in {`simp`, `comx`, `comp`, `comx_comp`, `incomp`} **do**
    $A_c \leftarrow \text{UniformSample}(\hat{A}, p_c)$ ▷ Select $p_c$ abstracts for $c$
    $S_c \leftarrow \emptyset$
    **for all** $a \in A_c$ **do**
        **if** $c \in \{\text{simp}, \text{incomp}\}$ **then**
            $s \leftarrow f_{\text{create}}(\text{annotator}, a)$ ▷ Create sentence for simple/incomplete
        **else**
            $s \leftarrow f_{\text{choose}}(a)$ ▷ Choose sentence for other categories
        $S_c \leftarrow S_c \cup \{s\}$ ▷ Collect one sentence per abstract
**Ensure** $\bigcup_c A_c = \hat{A}$ ▷ Coverage of all abstracts
$S_{\text{all}} \leftarrow \bigcup_c S_c$ ▷ All candidate sentences
$\hat{S} = \{(s, c) \mid c \in \mathcal{C},\ s \in S_c,\ f_{\text{ver}}(v_1, s) = f_{\text{ver}}(v_2, s)\}$ ▷ Keep only unanimously verified sentences with their category; where $\mathcal{C} = \{\text{simp}, \text{comx}, \text{comp}, \text{comx\_comp}, \text{incomp}\}$
**return** $\hat{S}$

**Algorithm 3** Step 2: Model Selection and Full Classification

**Input:** verified dataset $D = \{(s_i, y_i) \mid s_i \in \hat{S}\}$, candidate models $\mathcal{M}$, resolved abstracts $\hat{A}$
**Output:** full classification $\tilde{S} = \{(s, \ell) \mid s \in \text{Sentences}(\hat{A})\}$
$\text{bestScore} \leftarrow -\infty, \quad m^* \leftarrow \varnothing$
**for all** $m \in \mathcal{M}$ **do**
    Train $m$ on training split of $D$
    $\text{score} \leftarrow \text{Evaluate}(m, \text{val split of } D)$
    **if** $\text{score} > \text{bestScore}$ **then**
        $\text{bestScore} \leftarrow \text{score}$
        $m^* \leftarrow m$
$\tilde{S} \leftarrow \emptyset$
**for all** abstract $a \in \hat{A}$ **do**
    **for all** sentence $s \in \text{Sentences}(a)$ **do**
        $\ell \leftarrow m^*.\text{classify}(s)$ ▷ Classify each sentence in the main abstracts
        $\tilde{S} \leftarrow \tilde{S} \cup \{(s, \ell)\}$
**return** $\tilde{S}$

**Algorithm 4** Unified Sentence Simplification

**Input:** sentence sets $\{S_c\}_{c \in \{\text{comx,comp,comx\_comp}\}}$, prompt strategies $\mathcal{P}$, models $\mathcal{M}$
**Output:** simplified sentences $S_{\text{simp}}$
$S_{\text{simp}} \leftarrow \emptyset$
▷ Initialize output set
**for all** category $c \in \{\text{comx}, \text{comp}, \text{comx\_comp}\}$ **do**
▷ Iterate over sentence categories
    $\text{bestScore} \leftarrow -\infty$
▷ Reset best score
    **for all** $P \in \mathcal{P}$ **do**
▷ For each prompting strategy
        **for all** $M \in \mathcal{M}$ **do**
▷ For each model
            $\text{score} \leftarrow \text{EvaluatePromptModel}(P, M, S_c)$
▷ Evaluate on $S_c$
            **if** $\text{score} > \text{bestScore}$ **then**
                $\text{bestScore} \leftarrow \text{score}$; $(P^*, M^*) \leftarrow (P, M)$
▷ Update best pair
    **for all** $s \in S_c$ **do**
▷ Simplify each sentence in category $c$
        $\hat{s} \leftarrow f_{\text{prompt}}(P^*, M^*, s)$
▷ Generate simplified sentence
        $S_{\text{simp}} \leftarrow S_{\text{simp}} \cup \{\hat{s}\}$
▷ Collect simplified sentence
**return** $S_{\text{simp}}$
▷ Return all simplified sentences

**Algorithm 5** Relationship Extraction from Simplified Sentences

**Input:** simplified sentences $S_{\text{simp}}$, classified sentences $\tilde{S} = \{(s, \ell)\}$ best prompting strategy-model pair $(P^*, M^*)$
**Output:** relation triples $R = \{(e_1, r, e_2)\}$
$S_{\text{init}} \leftarrow \{\, s \mid (s, \ell) \in \tilde{S},\ \ell = \text{simp}\}$ ▷ Select only initially classified simple sentences
$S \leftarrow S_{\text{simp}} \cup S_{\text{init}}$ ▷ Combine with previously simplified sentences
$R \leftarrow \emptyset$ ▷ Initialize relation set
**for all** $s \in S$ **do**
    $(e_1, r, e_2) \leftarrow f_{\text{rel}}(P^*, M^*, s)$ ▷ Extract (entity$_1$, relationship, entity$_2$)
    **if** $(e_1, r, e_2) \neq \varnothing$ **then**
        $R \leftarrow R \cup \{(e_1, r, e_2)\}$ ▷ Keep valid triples
**return** $R$ ▷ All extracted relation triples

# C Tables

## C.1 Coreference Resolution

Table 8: Coreference resolution performance using cosine similarity across models and prompting styles.

| Model | Prompt | MUC (%) | B3 (%) | CEAF (%) | CoNLL (%) |
|---|---|---|---|---|---|
| **Mixtral-8x7B-Instruct-v0.1** | GIP | 13.34 | 70.96 | 70.96 | 51.75 |
| | COT | 14.50 | 70.87 | 70.87 | 52.08 |
| | **FICL** | **32.42** | **70.61** | **70.61** | **57.88** |
| | COT+FICL | 27.75 | 70.73 | 70.73 | 56.40 |
| Llama-3.1-8B-Instruct | GIP | 9.74 | 70.44 | 70.44 | 50.20 |
| | COT | 10.19 | 70.47 | 70.47 | 50.37 |
| | FICL | 27.16 | 69.57 | 69.57 | 55.43 |
| | COT+FICL | 26.00 | 69.58 | 69.58 | 55.06 |
| Llama-3.2-3B-Instruct | GIP | 7.13 | 70.56 | 70.56 | 49.41 |
| | COT | 6.14 | 71.32 | 71.32 | 49.60 |
| | FICL | 16.07 | 70.21 | 70.21 | 52.16 |
| | COT+FICL | 16.98 | 70.70 | 70.70 | 52.79 |
| Llama-3.3-70B-Instruct | GIP | 7.79 | 71.28 | 71.28 | 50.12 |
| | COT | 8.75 | 71.34 | 71.34 | 50.48 |
| | FICL | 31.25 | 70.94 | 70.94 | 57.71 |
| | COT+FICL | 28.97 | 70.95 | 70.95 | 56.95 |
| Mistral-7B-Instruct-v0.3 | GIP | 6.99 | 71.07 | 71.07 | 49.71 |
| | COT | 7.26 | 70.87 | 70.87 | 49.67 |
| | FICL | 16.96 | 70.99 | 70.99 | 52.98 |
| | COT+FICL | 18.58 | 70.74 | 70.74 | 53.35 |

# D Extracting Relationship Pairs from Simple Sentences

We went back to our coreference annotators and asked them if they could look at a small sample of the dataset and see if the model was successfully able to parse all relationships

Table 9: Model Accuracy on Boolean Relationship Extraction from Simple Sentences

| Model Name | # Params (B) | Accuracy | $F_1$-Score |
|---|---|---|---|
| LLaMA-3-8B | 8 | 98.00% | 98.00% |
| Mistral-7B | 7 | 87.00% | 93.55% |
| DeepSeek-Distilled-LLaMA-8B | 8 | 62.00% | 77.02% |
| Qwen-7B | 7 | 52.00% | 68.42% |
| LLaMA-2-7B | 7 | 35.00% | 51.85% |
| QwenChat-7B | 7 | 23.00% | 37.40% |
| DeepSeek-7B | 7 | 21.00% | 34.71% |
| DeepSeek-Prover-7B | 7 | 20.00% | 33.61% |
| DeepSeek-Distilled-Qwen-7B | 7 | 11.00% | 19.82% |
| **Mistral-MoE-8×7B** | **8×7** | **99.00%** | **99.50%** |
| DeepSeek-67B | 67 | 43.00% | 60.14% |
| LLaMA-2-13B | 13 | 13.00% | 23.01% |
| GPT-NeoX-20B | 20 | 0.00% | 0.00% |
| LLaMA-2-70B | 70 | 41.00% | 58.57% |

Table 10: Performance comparison across models and prompting styles

| Model | Prompting Style | Macro Average | Exact-Match | RMSE |
|---|---|---|---|---|
| LLAMA 3 8B | **COT+FICL** | 99.78% | 98.00% | 0.1078 |
| | COT | 82.86% | 64.00% | 0.4265 |
| | FICL | 68.89% | 28.33% | 0.5376 |
| | GIP | 45.81% | 27.33% | 0.9319 |
| MISTRAL 8 BY 7 | **COT+FICL** | 96.14% | 91.67% | 0.2323 |
| | COT | 92.42% | 83.00% | 0.3146 |
| | FICL | 90.23% | 78.00% | 0.3585 |
| | GIP | 82.26% | 57.67% | 0.4563 |
| MISTRAL 7 B | **COT+FICL** | 96.64% | 90.33% | 0.2356 |
| | COT | 84.72% | 64.33% | 0.4280 |
| | FICL | 85.78% | 67.00% | 0.4016 |
| | GIP | 78.88% | 53.00% | 0.5055 |
| LLAMA 3 70B | **COT+FICL** | 95.30% | 81.00% | 0.3213 |
| | COT | 86.39% | 68.33% | 0.4064 |
| | FICL | 71.98% | 45.33% | 0.6064 |
| | GIP | 62.01% | 35.00% | 0.7829 |

Table 11: Triple-extraction performance on the evaluation set

| Configuration | Triples | Precision | Recall | F1 Score |
|---|---|---|---|---|
| Human Standard | 398 | 100.00% | 100.00% | 100.00% |
| Full Model (Ours) | 422 | 92.00% | 92.90% | 92.40% |
| – Remove Coref Resolution | 376 | 80.60% | 74.60% | 77.50% |
| – Remove Sentence Decomposition | 208 | 74.60% | 46.20% | 57.20% |
| – Remove Coref + Sentence Decomposition | 220 | 76.80% | 42.70% | 54.80% |
| DeepSeek R1 | 323 | 93.50% | 78.60% | 85.40% |
| ChatGPT 4o | 215 | 98.10% | 52.80% | 68.50% |
| NotebookLM | 67 | 100.00% | 16.83% | 28.82% |
| ChatGPT 4.5 | 238 | 99.58% | 59.55% | 74.53% |

Table 12: Co-reference Group AB and Group CD, where each group's "link" set is the intersection of its two annotators.

| Statistic | Value |
|---|---|
| *Group definitions: A and B = 2,041; C and D = 1,929* | |
| Number of "link" assignments (Group AB) | 2,041 |
| Number of "link" assignments (Group CD) | 1,929 |
| Intersection | 1,847 |
| Union | 2,123 |
| Observed agreement $P_o$ | $\approx 0.87$ |
| Expected agreement estimated $P_e$ | 0.50 |
| Cohen's $\kappa$ | 0.74 |

# E Prompting Strategy

## E.1 Prompt Templates for Co-reference Resolution

**COT+FICL Co-reference Resolution Prompt**

You are a coreference resolution agent. Below is a biomedical abstract presented as tokenized text with indices. Your task is to identify and annotate coreference expressions within the text. For each co-referent expression:

- Record the surface form under "Expression".
- Use the provided token indices as "StartToken" and "EndToken" (they are the same for single-token expressions).
- Map each expression to its antecedent using "RefersTo" — either a noun phrase or named entity from the text.
- Only include pronouns or repeated noun phrases referring back to a prior concept or entity.

Use this format:

```
{
"Expression": "string",
"StartToken": int,
"EndToken": int,
"RefersTo": "string"
}
```

**Example:**
Given this tokenized abstract:

```
("BACKGROUND:", 0), ("There", 1),
("are", 2), ("few", 3), ("cases", 4),
("of", 5), ("pulmonary", 6),
("granulomatous", 7), ("changes", 8),
("secondary", 9), ("to", 10),
("primary", 11), ("biliary", 12),
("cirrhosis", 13), ("(PBC).", 14),
("No", 15), ("case", 16), ("of", 17),
("granulomatous", 18), ("lung", 19),
("disease", 20), ("secondary", 21),
("to", 22), ("PBC", 23),
("misdiagnosed", 24), ("as", 25),
("lung", 26), ("cancer", 27),
("had", 28), ("been", 29),
("reported.", 30), ("CASE", 31),
("SUMMARY:", 32), ("A", 33),
("middle-aged", 34), ("woman", 35),
("presented", 36), ("with", 37),
("lung", 38), ("nodules", 39),
("and", 40), ("was", 41),
("misdiagnosed", 42), ("with", 43),
("lung", 44), ("cancer", 45),
("by", 46), ("positron", 47),
("emission", 48),
("tomography/computed", 49),
("tomography.", 50), ("She", 51),
("underwent", 52), ("left", 53),
("lobectomy,", 54), ("and", 55),
("the", 56), ("pathology", 57),
("of", 58), ("the", 59),
("nodules", 60), ("showed", 61),
("granulomatous", 62),
("inflammation,", 63), ("which", 64),
("was", 65), ("then", 66),
("treated", 67), ("with", 68),
("antibiotics.", 69),
("However,", 70), ("a", 71),
("new", 72), ("nodule", 73),
("appeared.", 74), ("Further", 75),
("investigation", 76), ("with", 77),
("lung", 78), ("biopsy", 79),
("and", 80), ("liver", 81),
("serology", 82), ("led", 83),
("to", 84), ("the", 85),
("diagnosis", 86), ("of", 87),
("PBC,", 88), ("and", 89),
("chest", 90), ("computed", 91),
("tomography", 92),
("indicated", 93),
("significant", 94),
("reduction", 95), ("in", 96),
("the", 97), ("pulmonary", 98),
("nodule", 99), ("by", 100),
("treatment", 101), ("with", 102),
("methylprednisolone", 103),
("and", 104),
("ursodeoxycholic", 105),
("acid.", 106),
("CONCLUSION:", 107),
("Diagnosis", 108), ("of", 109),
("pulmonary", 110), ("nodules", 111),
("requires", 112),
("integrating", 113),
("various", 114), ("clinical", 115),
("data", 116), ("to", 117),
("avoid", 118), ("unnecessary", 119),
("pulmonary", 120),
("lobectomy.", 121)
```

```
[
{
"Expression": "PBC",
"StartToken": 14,
"EndToken": 14,
"RefersTo": "Primary
biliary cirrhosis"
},
{
"Expression": "PBC",
"StartToken": 23,
"EndToken": 23,
"RefersTo": "Primary biliary
cirrhosis"
},
{
"Expression": "She",
"StartToken": 51,
"EndToken": 51,
"RefersTo": "A middle-aged
woman"
},
{
"Expression": "PBC",
"StartToken": 88,
"EndToken": 88,
"RefersTo": "Primary biliary
cirrhosis"
}
]
```

Now process this tokenized abstract:
`{tokenized_text}`

**COT Co-reference Resolution Prompt**

You are a coreference resolution agent. Below is a biomedical abstract presented as tokenized text with indices. Your task is to identify and annotate coreference expressions within the text. For each co-referent expression:

- Record the surface form under "Expression".
- Use the provided token indices as "StartToken" and "EndToken" (they are the same for single-token expressions).
- Map each expression to its antecedent using "RefersTo" — either a noun phrase or named entity from the text.
- Only include pronouns or repeated noun phrases referring back to a prior concept or entity.

Use this format:

```
{
"Expression": "string",
"StartToken": int,
"EndToken": int,
"RefersTo": "string"
}
```

Now process this tokenized abstract:
`{tokenized_text}`

**FICL Co-reference Resolution Prompt**

You are a coreference resolution agent. Below is a biomedical abstract presented as tokenized text with indices. Your task is to identify and annotate coreference expressions within the text.
Use this format:

```
{
"Expression": "string",
"StartToken": int,
"EndToken": int,
"RefersTo": "string"
}
```

**Example:**
Given this tokenized abstract:

```
("BACKGROUND:", 0), ("There", 1),
("are", 2), ("few", 3), ("cases", 4),
("of", 5), ("pulmonary", 6),
("granulomatous", 7), ("changes", 8),
("secondary", 9), ("to", 10),
("primary", 11), ("biliary", 12),
("cirrhosis", 13), ("(PBC).", 14),
("No", 15), ("case", 16), ("of", 17),
("granulomatous", 18), ("lung", 19),
("disease", 20), ("secondary", 21),
("to", 22), ("PBC", 23),
("misdiagnosed", 24), ("as", 25),
("lung", 26), ("cancer", 27),
("had", 28), ("been", 29),
("reported.", 30), ("CASE", 31),
("SUMMARY:", 32), ("A", 33),
("middle-aged", 34), ("woman", 35),
("presented", 36), ("with", 37),
("lung", 38), ("nodules", 39),
("and", 40), ("was", 41),
("misdiagnosed", 42), ("with", 43),
("lung", 44), ("cancer", 45),
("by", 46), ("positron", 47),
("emission", 48),
("tomography/computed", 49),
("tomography.", 50), ("She", 51),
("underwent", 52), ("left", 53),
("lobectomy,", 54), ("and", 55),
("the", 56), ("pathology", 57),
("of", 58), ("the", 59),
("nodules", 60), ("showed", 61),
("granulomatous", 62),
("inflammation,", 63), ("which", 64),
("was", 65), ("then", 66),
("treated", 67), ("with", 68),
("antibiotics.", 69),
("However,", 70), ("a", 71),
```

```
("new", 72), ("nodule", 73),
("appeared.", 74), ("Further", 75),
("investigation", 76), ("with", 77),
("lung", 78), ("biopsy", 79),
("and", 80), ("liver", 81),
("serology", 82), ("led", 83),
("to", 84), ("the", 85),
("diagnosis", 86), ("of", 87),
("PBC,", 88), ("and", 89),
("chest", 90), ("computed", 91),
("tomography", 92),
("indicated", 93),
("significant", 94),
("reduction", 95), ("in", 96),
("the", 97), ("pulmonary", 98),
("nodule", 99), ("by", 100),
("treatment", 101), ("with", 102),
("methylprednisolone", 103),
("and", 104),
("ursodeoxycholic", 105),
("acid.", 106),
("CONCLUSION:", 107),
("Diagnosis", 108), ("of", 109),
("pulmonary", 110), ("nodules", 111),
("requires", 112),
("integrating", 113),
("various", 114), ("clinical", 115),
("data", 116), ("to", 117),
("avoid", 118), ("unnecessary", 119),
("pulmonary", 120),
("lobectomy.", 121)

[
{
"Expression": "PBC",
"StartToken": 14,
"EndToken": 14,
"RefersTo": "Primary biliary
cirrhosis"
},
{
"Expression": "PBC",
"StartToken": 23,
"EndToken": 23,
"RefersTo": "Primary biliary
cirrhosis"
},
{
"Expression": "She",
"StartToken": 51,
"EndToken": 51,
"RefersTo": "A middle-aged
woman"
},
{
"Expression": "PBC",
"StartToken": 88,
"EndToken": 88,
"RefersTo": "Primary biliary
cirrhosis"
}
]
```

Now process this tokenized abstract:
`{tokenized_text}`

**GIP Co-reference Resolution Prompt**

You are a coreference resolution agent. Below is a biomedical abstract presented as tokenized text with indices. Your task is to identify and annotate coreference expressions within the text. Use this format:
Use this format:

```
{
"Expression": "string",
"StartToken": int,
"EndToken": int,
"RefersTo": "string"
}
```

Now process this tokenized abstract:
`{tokenized_text}`

### E.2 Annotators Details

Annotator A is working on a medical degree Annotator B is a linguistic Annotator C is a biologist Annotator D is a linguistic

Annotators A and B are grouped to work together and produce a perfect work and Annotators C and D are also grouped in a similar pattern.

Annotator E, F and G all were tested before given the code and had a score of 93% in a specialized test different from the normal tests before being accepted for review. Annotators H and I, are postgraduate students. All annotators are from Nigeria.

They were all recruited by a recruitment expert Sophia Anuyah. They all spoke English and submitted their resumes and were invited for an online interview.

The annotators were paid in their local currency weekly, at an average of 40,000 NGN a week over the period of 5 weeks. The project was self-funded by the authors.

### E.3 Creating the Sentence Structure Data Set

Three initial annotators - Annotator E, B and F selected 7,500 sentences spanning five syntactic categories (compound-complex, compound, complex, simple, incomplete). Then two senior experts - G and H selected from a cohort of 12 people who took a classification test and scored above 93% were selected and then adjudicated these and reached consensus on 7,269 sentences out of 7,500 sentences making a 96.92% agreement rate. The authors chose not to resolve the 231 disagreements due to the current size of the dataset. The final dataset comprises of 2,118 (29.1%) compound-complex, 1,191 (16.4%) compound, 865 (11.9%) complex,

1,585 (21.8%) simple, and 1,510 (20.8%) incomplete sentences. From this set, we drew a balanced training sample of 2,000 sentences (400 per class) and reserved the remaining 5,269 sentences for testing. The dataset is available on github.

# F Prompt for Sentence Conversion

## F.1 Converting Complex Sentences to Simple Sentences

Given the sentence:

> "A prospective cohort study was conducted in Leeds, UK, based on routinely collected data from a service that allowed patients with symptoms of lung cancer to request CXR" (Bradley et al., 2021).

The process in this conversion is to identify the singular independent clause and the other dependent clauses

- Independent Clause: "A prospective cohort study was conducted in Leeds, UK."
- Dependent Clauses and Modifiers: (a) "that allowed patients with symptoms of lung cancer to request CXR." (b) "based on routinely collected data from a service"

In this example, there was one dependent clause, and one modifier which in our context still depends on the subject for RE, hence, they are looped together in our prompt. Once the LLM can correctly identify the independent clause, the next stage would be parse each relationship separately meaning we have three simple sentences i.e. (1) the independent clause (2) the subject of the IC and the DC and (3) the subject of the IC and the modifier. In this case:

$$S = \underbrace{\text{(A prospective cohort study was conducted in Leeds, UK)}}_{S_{\text{main}}} \cup \underbrace{\text{(that allowed patients … to request CXR)}}_{DC} . \tag{4}$$

We define an extraction operator $\mathcal{E}(\cdot)$ that maps a complex sentence to a set of simple (independent) sentences:

$$\mathcal{E}(S_{\text{complex}}) = \left\{ S_{\text{main}},\ \text{R}(DC_1),\ \text{R}(DC_2),\ \ldots \right\} . \tag{5}$$

where R $\rightarrow$ Rewrite

$$S = S_{\text{main}} \cup DC \tag{6}$$

we get:

$$\mathcal{E}(S) = \{\text{S1}, \text{S2},\ \text{S3}\} .$$

$S1 \rightarrow$ A prospective cohort study was conducted in Leeds, UK. $S2 \rightarrow$ The study was based on routinely collected data from a service. $S3 \rightarrow$ The service allowed patients with symptoms of lung cancer to request CXR

## F.2 Converting Compound Sentences to Simple Sentences

Given the sentence:

> "Lung cancer stands prominently among the foremost contributors to human mortality, distinguished by its elevated fatality rate and the second-highest incidence rate among malignancies, and the metastatic dissemination of lung cancer stands as a primary determinant of its elevated mortality and recurrence rates."

Our goal is to break down this compound sentence into simpler, stand-alone statements.

- Independent Clause 1 (IC1): "Lung cancer stands prominently among the foremost contributors to human mortality."
- Independent Clause 2 (IC2): "The metastatic dissemination of lung cancer stands as a primary determinant of its elevated mortality and recurrence rates."
- Dependent Modifier (DM): "distinguished by its elevated fatality rate and the second-highest incidence rate among malignancies"

Here, IC1 and IC2 are connected by a coordinating conjunction (i.e., "and"), which is typical in compound sentences. The phrase "distinguished by its elevated fatality rate ... among malignancies" modifies "Lung cancer" (from IC1).

To convert this compound sentence into simple sentences, we isolate each clause, ensuring each stands alone:

$$S = (IC1 \cup DM \cup IC2)$$

We define an extraction operator $\mathcal{E}(\cdot)$ that maps a compound sentence $S_{\text{compound}}$ to a set of simple (independent) sentences:

$$\mathcal{E}(S_{\text{compound}}) = \{S_1,\ S_2,\ S_3\}\,.$$

Applying it to our sentence:

$$S = (IC1 \cup DM \cup IC2)$$

we obtain three simple sentences:

> $S1 \rightarrow$ Lung cancer stands prominently among the foremost contributors to human mortality.
> $S2 \rightarrow$ It is distinguished by its elevated fatality rate and the second-highest incidence rate among malignancies.
> $S3 \rightarrow$ The metastatic dissemination of lung cancer stands as a primary determinant of its elevated mortality and recurrence rates.

### F.3 Converting Compound-Complex Sentences to Simple Sentences

Given the sentence:

> "Although lung cancer is the leading cause of US cancer-related deaths, lung cancer screening with a low radiation dose chest computed tomography scan is now standard of care for a high-risk eligible population, and clinicians and surgeons must evaluate the trade-offs of benefits and harms, including the identification of many benign lung nodules, overdiagnosis, and complications."

- Independent Clause 1 (IC1): "Lung cancer screening with a low radiation dose chest computed tomography scan is now standard of care for a high-risk eligible population"
- Independent Clause 2 (IC2): "Clinicians and surgeons must evaluate the trade-offs of benefits and harms, "
- Dependent Clause (DC): "Although lung cancer is the leading cause of US cancer-related deaths"
- Modifiers: "including the identification of many benign lung nodules, overdiagnosis, and complications"

We see that DC modifies or sets a contrasting context for IC1, and IC1 is coordinated with IC2 via "and." To convert this into simple sentences, each clause (or key part of a clause) should form its own standalone statement:

$$S = (DC \cup IC1 \cup IC2)$$

Using our extraction operator $\mathcal{E}(\cdot)$:

$$\mathcal{E}(S_{\text{compound-complex}}) = \{S_1,\ S_2,\ S_3,\ ...,S_n\}\,,$$

we obtain:

> $S1 \rightarrow$ Lung cancer is the leading cause of US cancer-related deaths.
> $S2 \rightarrow$ Lung cancer screening with a low-dose chest computed tomography scan is now standard of care for a high-risk eligible population.
> $S3 \rightarrow$ Lung cancer screening is recommended for a high-risk, eligible population.
> $S4 \rightarrow$ Clinicians and surgeons must evaluate the trade-offs of benefits and harms,
> $S5 \rightarrow$ Evaluated trade-offs of benefits and harms include the identification of many benign lung nodules.
> $S6 \rightarrow$ Evaluated trade-offs of benefits and harms include the risk of overdiagnosis.
> $S7 \rightarrow$ Evaluated trade-offs of benefits and harms include complications from lung-cancer screening

### F.4 Prompts

**COT+FICL Complex Sentence Conversion**

Below is a step-by-step process. For each example, think step by step, then output only the simplified sentences in the form:

S1 $\rightarrow$ ... S2 $\rightarrow$ ... ...

one per line, and nothing else.

**Example 1:**
**Input:**

> "A prospective cohort study was conducted in Leeds, UK, based on routinely collected data from a

service that allowed patients with symptoms of lung cancer to request CXR."

**Chain-of-Thought:**

1. Identify the independent clause: "A prospective cohort study was conducted in Leeds, UK."
2. Identify dependent clauses/modifiers:
   - Modifier A: "based on routinely collected data from a service"
   - Dependent clause B: "that allowed patients with symptoms of lung cancer to request CXR"
3. Rewrite each as a standalone simple sentence:

**Output:**

- S1 → A prospective cohort study was conducted in Leeds, UK.
- S2 → The study was based on routinely collected data from a service.
- S3 → The service allowed patients with symptoms of lung cancer to request CXR.

**Example 2:**
**Input:**

"After the cells were treated with the drug, which had been synthesized in our lab, we measured the change in fluorescence using a spectrophotometer."

**Chain-of-Thought:**

1. Independent clause: "We measured the change in fluorescence using a spectrophotometer."
2. Dependent clauses/modifiers:
   - Dependent clause A: "After the cells were treated with the drug"
   - Modifier B: "which had been synthesized in our lab"
3. Rewrite each as standalone simple sentences:

**Output:**

- S1 → We measured the change in fluorescence using a spectrophotometer.
- S2 → The cells were treated with the drug.
- S3 → The drug had been synthesized in our lab.

**Now apply the same process to this new sentence:**
**Input:** `"{{sentence}}"`

*****OUTPUT ONLY the simplified sentences, one per line in the form S1 → …, S2 → …, etc., and nothing else.*****
Now process this abstract:
"'ABSTRACT GIVEN HERE"'

**COT+FICL Compound Sentence Conversion**

Below is a process to convert a compound sentence into simple sentences. For each example, think step by step, then output only the simplified sentences in the form:

S1 → . . .  S2 → . . .  . . .

one per line, and nothing else.

**Example 1:**
**Input:**

"Lung cancer stands prominently among the foremost contributors to human mortality, distinguished by its elevated fatality rate and the second-highest incidence rate among malignancies, and the metastatic dissemination of lung cancer stands as a primary determinant of its elevated mortality and recurrence rates."

**Chain-of-Thought:**

1. Identify the independent clauses:
   - IC1: "Lung cancer stands prominently among the foremost contributors to human mortality."

- IC2: "The metastatic dissemination of lung cancer stands as a primary determinant of its elevated mortality and recurrence rates."

2. Identify modifiers:
   - Modifier: "distinguished by its elevated fatality rate and the second-highest incidence rate among malignancies" (modifies IC1)
3. Rewrite all parts as simple, standalone sentences.

**Output:**

- S1 → Lung cancer stands prominently among the foremost contributors to human mortality.
- S2 → It is distinguished by its elevated fatality rate and the second-highest incidence rate among malignancies.
- S3 → The metastatic dissemination of lung cancer stands as a primary determinant of its elevated mortality and recurrence rates.

**Example 2:**
**Input:**

> "Climate change accelerates the melting of polar ice, and rising sea levels threaten coastal communities around the world."

**Chain-of-Thought:**

1. Identify the independent clauses:
   - IC1: "Climate change accelerates the melting of polar ice."
   - IC2: "Rising sea levels threaten coastal communities around the world."
2. No dependent clauses or modifiers.
3. Rewrite each as a standalone simple sentence.

**Output:**

- S1 → Climate change accelerates the melting of polar ice.
- S2 → Rising sea levels threaten coastal communities around the world.

**Now apply the same process to this new sentence:**
**Input:** "`{{sentence}}`"

**OUTPUT ONLY the simplified sentences, one per line in the form S1 → . . . , S2 → . . . , etc., and nothing else.**

**COT+FICL Compound-Complex Sentence Conversion**

Below is a process to split a compound-complex sentence into standalone simple sentences. Think step by step, then apply.

**Example 1:**
**Input:**

> "Although lung cancer is the leading cause of US cancer-related deaths, lung cancer screening with a low radiation dose chest computed tomography scan is now standard of care for a high-risk eligible population, and clinicians and surgeons must evaluate the trade-offs of benefits and harms, including the identification of many benign lung nodules, overdiagnosis, and complications."

**Chain-of-Thought:**

1. Dependent Clause (DC): "Although lung cancer is the leading cause of US cancer-related deaths"
2. Independent Clause 1 (IC1): "Lung cancer screening with a low-dose chest computed tomography scan is now standard of care for a high-risk eligible population"
3. Independent Clause 2 (IC2): "Clinicians and surgeons must evaluate the trade-offs of benefits and harms"

4. Modifier list: “including the identification of many benign lung nodules, overdiagnosis, and complications”
5. Rewrite into standalone simple sentences.

**Output:**

- S1 → Lung cancer is the leading cause of US cancer-related deaths.
- S2 → Lung cancer screening with a low-dose chest computed tomography scan is now standard of care for a high-risk eligible population.
- S3 → Lung cancer screening is recommended for a high-risk, eligible population.
- S4 → Clinicians and surgeons must evaluate the trade-offs of benefits and harms.
- S5 → Evaluated trade-offs include the identification of many benign lung nodules.
- S6 → Evaluated trade-offs include the risk of overdiagnosis.
- S7 → Evaluated trade-offs include complications from lung-cancer screening.

**Example 2:**
**Input:**

> “Although warmed by the sun, the fields remained dry, and farmers worried about the drought.”

**Chain-of-Thought:**

1. Dependent Clause (DC): “Although warmed by the sun”
2. Independent Clause 1 (IC1): “The fields remained dry”
3. Independent Clause 2 (IC2): “Farmers worried about the drought”
4. Rewrite into standalone simple sentences.

**Output:**

- S1 → The sun warmed the fields.
- S2 → The fields remained dry.
- S3 → Farmers worried about the drought.

**Now apply to this new sentence:**
**Input:** `"{{sentence}}"`

**OUTPUT ONLY the simplified sentences, one per line in the form S1 → . . . , S2 → . . . , etc., and nothing else.**

**COT + FICL Relationship Extraction for Knowledge Graph**

You are a knowledge graph relationship extraction agent. Your task is to extract structured relationships from simple sentences to create knowledge graph triples. Each triple should contain two entities and the relationship between them.

- Analyze the sentence structure and identify key components.
- Extract all meaningful entities (nouns, noun phrases, proper nouns, concepts).
- Identify relationships between entities based on verbs, prepositions, and semantic meaning.
- Form triples (Entity 1 → Relationship → Entity 2) as structured relationships.
- Validate that each triple captures meaningful semantic information.

**Examples:**

> "Regulating miR-497-5p provides a potential targeted therapy for lung cancer treatment."

```
[{
"Entity 1": "regulating miR-497-5p",
"Entity 2": "lung cancer targeted
treatment",
"Relationship": "provides"
}]
```

"The activation of caspase signal pathway was the reason for stronger apoptosis."

```
[{
"Entity 1": "activation of caspase
signal pathway",
"Entity 2": "stronger apoptosis",
"Relationship": "was the reason for"
}]
```

"With clinical significance features selection, over-sampling methods achieved the highest AUC results."

```
[{
"Entity 1": "clinical significance
features selection",
"Entity 2": "over-sampling methods",
"Relationship": "With"},
{"Entity 1": "over-sampling methods",
"Entity 2": "highest AUC results",
"Relationship": "achieved"}]
```

Now extract knowledge graph relationships from this sentence: `{sentence}`

All other prompt types are in our code base.

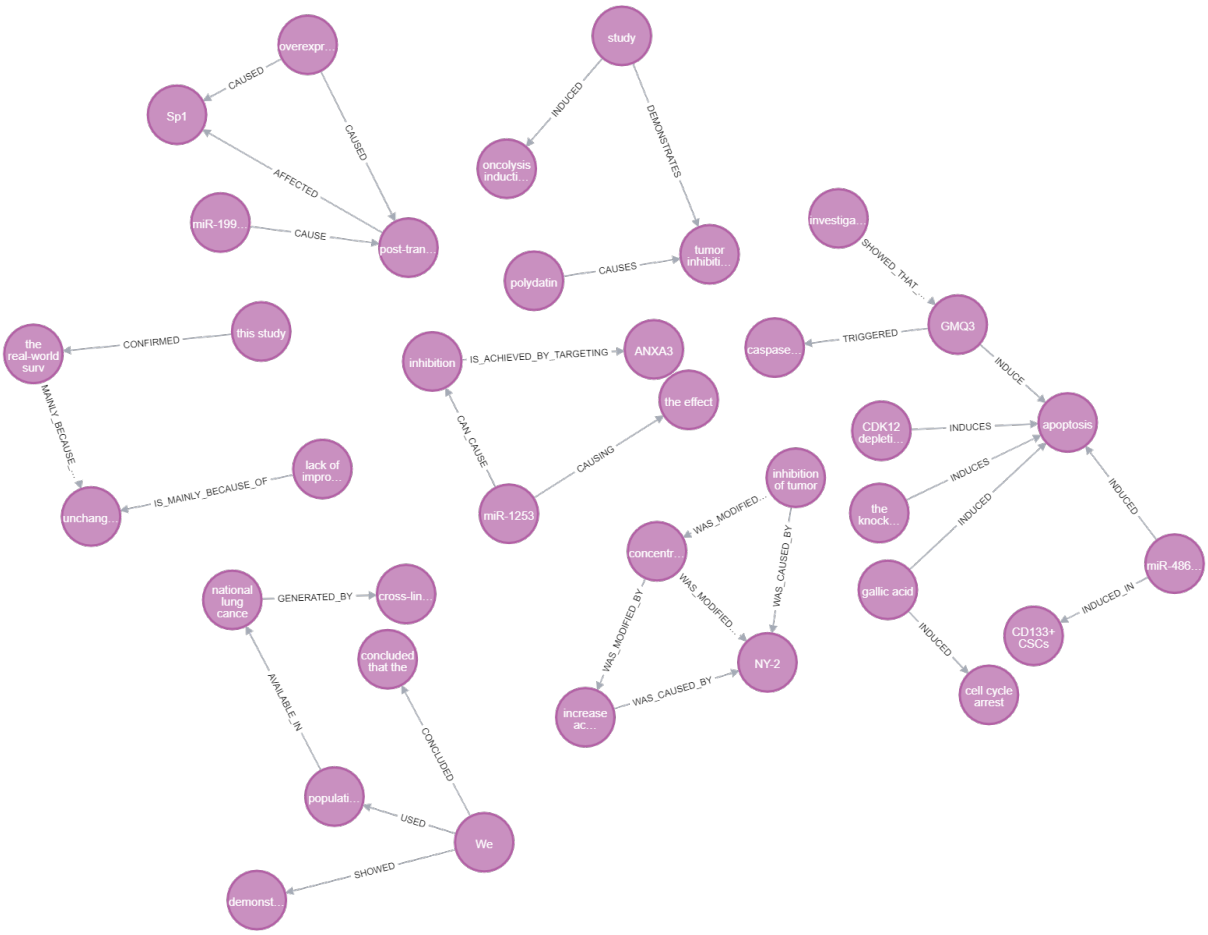


Figure 3: Subsection of the Knowledge Graph.