LLM-based Atomic Propositions Help Weak Extractors: Evaluation of a Propositioner for Triplet Extraction

Luc Pommeret[1], Thomas Gerald[1], Patrick Paroubek[1], Sahar Ghannay[1], Christophe Servan[2], Sophie Rosset[1]
[1]Université Paris-Saclay, CNRS, LISN, 91400, Orsay, France
[2] AMIAD, Pôle Recherche, 91120, Palaiseau, France
surname.name@lisn.fr

Abstract

Knowledge Graph construction from natural language requires extracting structured triplets from complex, information-dense sentences. In this paper, we investigate if the decomposition of text into *atomic propositions* (minimal, semantically autonomous units of information) can improve the triplet extraction. We introduce `MPropositionneur-V2`, a small multilingual model covering six European languages trained by knowledge distillation from `Qwen3-32B` into a `Qwen3-0.6B` architecture, and we evaluate its integration into two extraction paradigms: entity-centric (`GLiREL`) and generative (`Qwen3`). Experiments on SMiLER, FewRel, DocRED and CaRB show that atomic propositions benefit weaker extractors (`GLiREL`, `CoreNLP`, 0.6B models), improving relation recall and, in the multilingual setting, overall accuracy. For stronger LLMs, a fallback combination strategy recovers entity recall losses while preserving the gains in relation extraction. These results show that atomic propositions are an interpretable intermediate data structure that complements extractors without replacing them.

Keywords: atomic propositions, knowledge graph, triplet extraction, OpenIE, propositioner, multilingual NLP

1. Introduction

The interpretability of Natural Language Processing (NLP) models is a requirement for applications like fact-checking and automated construction of Knowledge Graphs (KGs). While neural models have achieved state-of-the-art results, their internal mechanisms remain opaque. Most current explainability methods are *post hoc*, seeking to explain decisions after the training.

In this paper, we argue for a shift towards interpretability by design, where the data structure itself is traceable. Our approach is based on natural language (NL) *atomic propositions*, a minimal, semantically autonomous unit of information. For defining and producing NL atomic propositions, we rely on the formalism of Semantic Information Theory(Bar-Hillel and Carnap, 1953), which defines the decomposition of non-atomic propositions into atomic ones [1]. For instance, the sentence "The cat and the dog are in the kitchen" atomizes into two NL atomic propositions: "The cat is in the kitchen" and "The dog is in the kitchen". Neither of these two propositions can be decomposed further, because otherwise we would add to the original semantic content hallucinations (spurious information). If the original proposition was not a conjunction but a disjunction ("The cat **or** the dog") decomposing it would introduce new information since we would have a disjunction of two atomic propositions each one holding more semantic information than the original text. More details are available in section 3.

For the implementation of the sentence decomposition process into NL atomic propositions, we rely in this first experiment on a limited-depth recursive prompting of a multilingual LLM, an atomic proposition being defined as a prompting fixed-point. To perform the atomisation, we propose a small multilingual model, the `MPropositionneur-V2`, that can perform coreference resolution to produce atomic propositions from text. For more details, please see section 4.

We aim to show that in Open Information Extraction (OpenIE), using NL atomic propositions can improve performance while preserving interpretability. Here, the interpretability is the auditability of the data structure, i.e. the NL atomic propositions. These propositions serve as an intermediary representation in the process of mapping a natural language sentence S to a set of formal triplets $\{(s, r, o)\}$, where s is the subject entity, o the object entity, and r the relation. In our experiment, we evaluate whether splitting textual information into atomic propositions could positively impact triplet extraction. We hypothesize that extracting entities and relations could be made easier from less information-dense text chunks, especially NL atomic propositions.

Our contributions consist of a new multilingual propositioner and a pipeline leveraging atomic propositions to enhance entity-relation extraction based on LLM.

[1] In logic, atomic propositions are closed formulae that cannot be further decomposed into several smaller propositions without loss of semantic integrity or introduction of "bad" informational cuts (see Section 3)

Proceedings of the Knowledge Graphs and Large Language Models Workshop 2026 (KG-LLM@LREC26), pages 134–143
May 16, 2026.

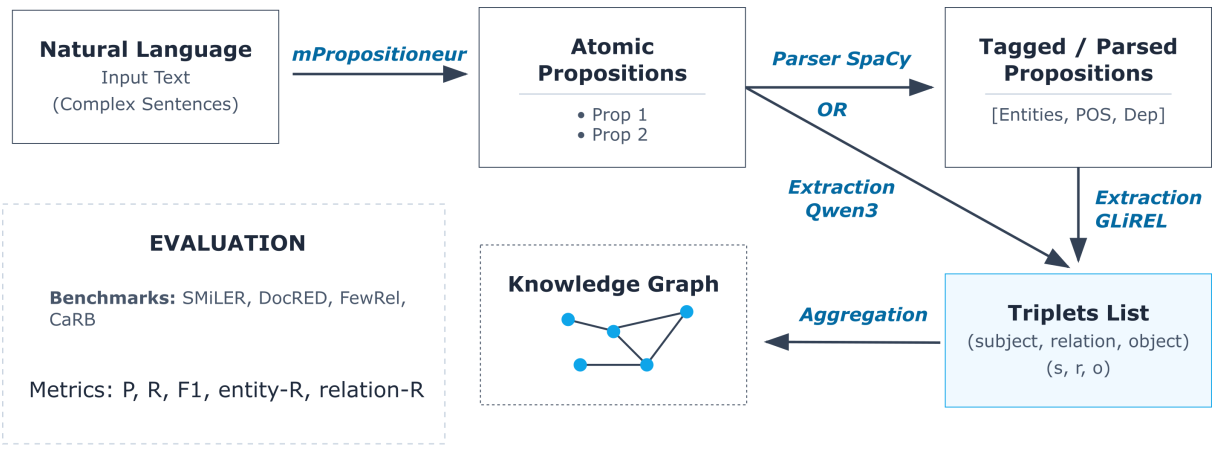

Figure 1: The upper schema depicts the pipeline's stages: at stage 1, we extract atomic propositions from the source text; at stage 2, we extract triplets using either a dependency parser or generative LLMs; and finally, we build the knowledge graph from the entities and relations retrieved. For evaluation, we limit to the triplet entity-relation benchmark.

We report the evaluation of the multilingual propositioner trained through distillation of a large language model (LLM), compared to a baseline composed of a rule-based method leveraging a dependency parser. To assess the effectiveness of our model and the extraction pipeline, we evaluate the method on different entity-relation benchmarks for relation extraction and triplet validation. These benchmarks cover a variety of tasks, such as open and closed information extraction, sentence- and document-based extraction, and extraction based solely on relations and triplets.

The paper is organised as follows: First, in Section 2, we review related works, describing previous triplet extraction approaches and methods based on propositioners. Section 3 presents the formalism of the atomic proposition. Section 4 describes our pipeline in depth. We then present the experimental protocol to answer the research question in Section 5. Section 6 discusses the different results, and Section 7 concludes with suggestions for future work.

2. Related Works

For inference and/or information retrieval (Xiang et al., 2026), it is common to represent information as a Knowledge Graph (KG). While a wide range of KGs automatically extracted from many different sources is available, for instance, using wikidata taxonomy and entities (Waagmeester et al., 2020; Hassanzadeh, 2021), extracting KGs from natural language source of information remains a significant challenge (Waagmeester et al., 2020). We can split the wide range of approaches for extracting automatically KG into two categories.

The first one performs named entity recognition (NER), and then applies entity linking and relation extraction approaches to build a knowledge graph. For example, in the sentence 'The Eiffel Tower is in Paris', the entities must be linked to the knowledge graph (Paris is entry Q90 in Wikidata) and the relation must be linked to a vocabulary of relations (here, hasLocation, property P131 in Wikidata) (Hogan et al., 2021). However, this approach lacks flexibility when dealing with the intrinsic complexity of natural language.

The second kind of approach is called Open Information Extraction (OpenIE), and it aims to extract triplets (s, r, o)[2], without relying on predefined sets of entities and/or relations (Etzioni et al., 2008).

To extract triplets from sentences, modern approaches use LLMs, especially transformer encoder or decoder models. Models like `mREBEL` (Huguet Cabot et al., 2023) generate triplets directly from complex input sentences. Such generative models often struggle with long-range dependencies, nested clauses, and coordination, leading to lower recall on complex structures. More recently, `GLiREL` (Boylan et al., 2025) performs zero-shot relation classification by jointly encoding entity pairs and candidate relation labels. It is more robust but can be sensitive to the quality of the initial entity recognition. One way to address this duality is through the recourse of sentence simplification for relation extraction.

Simplifying text before extracting relations has been explored, and, for instance, (Miwa et al., 2010) proposed an entity-focused sentence simplification method to improve relation extraction. More recently, (Niklaus et al., 2016) introduced

[2] (subject, relation, object)

a rule-based sentence simplification system that rewrites complex sentences into simpler sentences for Open Information Extraction. However, these approaches rely on syntactic rules or dependency parsing, are limited to a single language, and lack coreference resolution.

Simplification by structural atomisation of information is of interest for many tasks: in Retrieval Augmented Generation (RAG), the indexing of atomic propositions reduces the noise in dense retrieval (Chen et al., 2024); in Natural Language Inference (NLI), context augmentation by atomic propositions increases performance (Stacey et al., 2023); in fact-checking, it is possible to decompose the text and verify each atomic proposition recursively (Min et al., 2023); and finally, in Summary Evaluation, this method allows one to approach human judgement (Herserant and Guigue, 2025).

Recently, (Min et al., 2023) proposed the use of LLM-based atomic propositions in NLP. The atomic propositions approach combines cutting the information into small sentence pieces (the smallest we can, without loss of information, see Section 3) and coreference resolution. One advantage of atomic propositions is the *structure*. This approach gives a fixed structure to the information we want to verify, which is more easily parseable and has been used for Information Retrieval (Chen et al., 2024) and Summary Evaluation (Herserant and Guigue, 2025).

Based on these recent works, we propose using atomic propositions to construct knowledge graphs. We propose using a propositioner based on a distilled multilingual language model to perform the atomic propositions extraction. This model handles simplification and coreference resolution across six languages and is grounded in a formal framework. Once the propositions are extracted, the text input is flattened. Then, we experiment with different methods to extract entity-relation triplets. For each triplet, we produce a graph that links the entities with the relations produced by the model.

3. Formal Framework

The abstraction underlying our objectives is the theoretical atomic proposition. To enlighten the atomization process, we use the formalism of Semantic Information Theory (Bar-Hillel and Carnap, 1953). This formalism provides a strong understanding of information in terms of signification and gives a criterion for cutting a proposition in a way that preserves information.

3.1. Information Content

Let ϕ be a formula. We define $I(\phi) = -\log_2(\frac{|\mathrm{Cont}(\phi)|}{|W|})$, where $\mathrm{Cont}(\phi)$ is the set of worlds[3] satisfying ϕ. A cut of ϕ into ψ is **safe** if ψ is a sub-formula of ϕ and $I(\phi) > I(\psi)$. It is **bad** if $I(\phi) \leq I(\psi)$.

3.2. The CNF Condition

We prove (in Annex B) that a proposition is atomic if and only if it is a clause in a Conjunctive Normal Form (CNF).

- **Conjunctions (**$A \wedge B$**)**: Splitting into A or B is safe ($I(A \wedge B) > I(A)$).
- **Disjunctions (**$A \vee B$**)**: Splitting into A is bad ($I(A) > I(A \vee B)$), as it "hallucinates" specificity.

Thus, our propositioner is designed to split conjunctions while preserving the integrity of disjunctive facts.

3.3. BCP Paradox

The BCP paradox states that a contradiction (like $A \wedge \neg A$) has infinite information in the extended reals $\overline{\mathbb{R}} = \mathbb{R} \cup \{-\infty, +\infty\}$:

$$-\log_2(\frac{|\mathrm{Cont}(A \wedge \neg A)|}{|W|}) = -\log_2(0) = +\infty$$

However, for clauses of a CNF, this paradox cannot arise because they cannot be contradictions (Cori and Lascar, 2003). Therefore, atomic propositions cannot have infinite information.

In the current implementation, we expect the model to follow the previous formal framework. However, we have only performed a manual and partial evaluation. If this is positive, we cannot guarantee that this will be the case for all extracted propositions. Additional experiments will be conducted outside the scope of this work.

4. Proposed Approach

The aim of this paper is to show that triplet extraction could benefit from atomic propositions. We define different stages to extract triplets, with a full pipeline depicted in Figure 1. The global pipeline consists of three stages:

1. **Atomization**: The complex text is processed by `MPropositionneur-V2`. This model, distilled from `Qwen3-32B` into a `Qwen3-0.6B` architecture, recursively splits the text until each proposition is stable and autonomous by using the prompt proposed in Figure 2.

[3] A world W is a determined assignment of truth values for each atomic subformula of ϕ.

2. **LLM prompting**: We use the `Qwen3-4B` model to generate triplets directly from an atomized chunk using a dedicated prompt (Figure 3)
3. **KG Building**: Extracted triplets are aggregated into a Knowledge Graph, where nodes represent entities and edges represent relations.

As a baseline, we replace stage 2 with two sub-stages by using the **Parsing** and **Triplet Extraction** as follows:

2.1. **Parsing**: Each atomic proposition is parsed (here using `SpaCy` or `Stanza`) to extract part-of-speech (POS) tags, named entity tags, and dependency trees.

2.2. **Triplet Extraction**: A large language model or a neural based pattern matcher (for example `GLiREL`) extracts triplet candidates (s, r, o) from the simplified, parsed atoms.

```
You are an expert in disambiguation
and information extraction.
```

```
You must decompose the text into
atomic propositions (single facts)
that are FULLY AUTONOMOUS.
ABSOLUTE RULES:
1. ZERO PRONOUNS: "He", "She", "They",
"His", "Her", "Its", "This one"
ARE FORBIDDEN.
ALWAYS replace them with the
full name of the entity.
2. CONTEXT: Each sentence must be
readable alone without knowing
its source.
3. REPETITION: Repeat the subject
in EACH sentence.
```

```
OUTPUT FORMAT: Only a JSON array of
strings.
Title: title
Content: content
Output:
```

Figure 2: Prompt Template used for the distillation of the propositioner used in stage 1.

Figure 4 illustrates the input and the output of the entire pipeline.

```
Extract all factual
(subject, predicate, object)
triples from the sentence.
One triple per line in the format:
subject | predicate | object
No explanations. If no triple can be
extracted, write nothing.
```

```
Sentence: text
```

Figure 3: Prompt Template used for the stage 2.

```
Input: "Marie Curie, a Polish-born
physicist, won the Nobel Prize in Physics."
```

```
Atomic Props: ["Marie Curie is a physicist.",
"Marie Curie was born in Poland.", "Marie
Curie won the Nobel Prize in Physics."]
```

```
Parsed Triplets: (Marie Curie, occupation,
physicist), (Marie Curie, birthplace,
Poland), (Marie Curie, award, Nobel Prize in
Physics).
```

Figure 4: Example of a sentence input processed through the whole pipeline, the atomic output, and the triplets extracted.

5. Experimental Protocol

5.1. Propositioner

We train a propositioner[4], *i.e.* a model that transforms a text input into a list of atomic propositions, via knowledge distillation (Hinton et al., 2015), using `Qwen3-32B` as the teacher model and `Qwen3-0.6B` as the student. The training data consist of chunks of Wikipedia articles in six European languages: English, French, Spanish, Italian, German, and Portuguese (Foundation). We trained the models for 2 epochs, on an A6000 NVIDIA GPU.

5.2. Nat. Lang. Recursive Propositioner

In Algorithm 1, we describe the recursive propositioner method. This algorithm is designed to recursively apply the `MPropositionneur-V2` to all natural language propositions generated ($\mathcal{P}_i$) until either all have been proved to be a propositioner fixed-point or the recursion depth has reached an empirical threshold value (here $N = 5$). In the latter case, we return only the subset of proved atomic propositions ($\mathcal{A}_i$).

Algorithm 1 propositioner

Require: $N = 5$, t is a text, $i \in \mathbb{N}$, $\mathcal{M}$ is the propositioner,

$i \leftarrow 0$; $\mathcal{P}_0 \leftarrow \mathcal{M}(t)$;

$\mathcal{A}_0 \leftarrow \{p \in \mathcal{P}_0,\ \ \{p\} = \mathcal{M}(p)\}$

while $(\mathcal{P}_i \neq \mathcal{A}_i) \wedge (i < N)$ **do**

  $\mathcal{P}_{i+1} \leftarrow \bigcup_{x \in \mathcal{P}_i} \mathcal{M}(x)$

  $\mathcal{A}_{i+1} \leftarrow \{x \in \mathcal{P}_{i+1},\ \ \{x\} = \mathcal{M}(x)\}$

  $i \leftarrow i + 1$

end while

return $\mathcal{A}_i$

[4]Available here : https://huggingface.co/Zual/MPropositionneur-V2

5.3. Datasets and Benchmark

We evaluate our pipeline on the SMiLER dataset (Seganti et al., 2021) (a multi-domain relation extraction benchmark covering 14 languages and various relation types) where the entities are already known and the task is to find the relations between entity pairs, on FewRel dataset (Han et al., 2018), on DocRED (Yao et al., 2019) (a Document-based triplet extraction dataset) and on CaRB (Bhardwaj et al., 2019) (an openIE dataset for triplet extraction).

5.4. Metrics

To evaluate triplet extraction performances we will consider the following metrics:

- Precision (P), Recall (R), and F1-Score. Area Under the Curve (AUC) is also used for the CaRB benchmark natively.
- Entity Recall: The percentage of gold entities found in the output (exact match with substring and macrostring accepted).
- Relation Recall: The accuracy of extracted triplets vs. gold standards, by mapping to the finite vocabulary of `GLiREL` using a semantic mapper (cosine similarity with BERT, threshold optimised on the dev set).

5.5. Baselines & Configurations

We compare the performance of direct pipeline (`GLiREL` or `Qwen3`) vs. `MPropositionneur-V2` + direct pipeline. Our hypothesis is that atomization reduces the syntactic noise that typically hinders relation extraction on complex sentences.

For the evaluation, we analyse three different configurations:

- **Direct**: Triplets are directly extracted from source text.
- **Prop**: Triplets are extracted only from the atoms produced by the propositioner.
- **Comb**: For SMiLER and FewRel benchmarks, since the entity oracle is known, if entities are found, the associated triples are saved; otherwise, the triples are extracted from atomic propositions. Comb is therefore a fallback method.
- **Union**: The triplets are extracted from atomic propositions and from the raw paragraph/document independently. Triplets from both the **direct** and **prop** pipelines are merged. The objective is to evaluate the contribution of atomic propositions to the direct pipeline.

The baseline approach is a triplet extractor using `GLiREL`. Then we compare the baseline with prompted approaches using LLMs, where models are queried to generate the triplets from either the source text (**direct**) or from propositions (**prop**). For prompt models, we selected `Qwen3-0.6B` and `Qwen3-4B` instruct models; these models are comparable in terms of size (number of weights) to the size of the propositioner model. It should be noted that these models were not fine-tuned for the triplet extraction task. Rather, a zero-shot instruction approach was employed (see the prompt used in Figure 3).

6. Results and Analysis

In this section we report and discuss the results of the designed experiments. We evaluate quantitatively the propositioner for the triplet extraction method. By flattening the text, relations are made explicit, allowing the extractors to capture facts that would otherwise be missed in complex sentences.

6.1. Evaluation on Multilingual Triplet Extraction

We report in Table 1 results in accuracy (number of triplet correctly extracted), entity-recall and relation-recall on both SMiLER and FewRel benchmarks compared to `GLiREL` approach.

We also report results for the different configurations: "Direct", where models try to extract the triplet directly from the raw text; "Prop", where the models are only considering the set of propositions to extract relations; "Comb", which is the combination of raw text and the list of atomic propositions.

First, looking at the Macro-avg, we can observe that in almost all cases, the number of correctly extracted relations benefits from the atomic propositions. While direct pipelines achieve better accuracy and entity recall, a combination of raw text and atomic propositions yields better performance for small models, demonstrating that both methods support the retrieval of different entities or relations. This combination of the two approaches has a positive impact on the proposition in the triplet extraction pipelines.

However, within this pipeline we consider that we know a relation exists between two entities, and thus it corresponds to relations classification setting rather than a triplet extraction setting. From a model perspective, we show that a larger LLM-based approach is more powerful across all metrics (`Qwen3-4B`), although it comes at a higher computational and memory cost.

Evaluation on triplet extraction on documents. Table 2 presents results that favour proposition

Benchmark	Pipeline	GLiREL			Qwen3-0.6B			Qwen3-4B		
		Acc	e-rec	r-rec	Acc	e-rec	r-rec	Acc	e-rec	r-rec
SMiLER	English – Direct	49.8	99.0	50.3	35.7	100.0	35.7	71.4	100.0	71.4
	English – Prop	43.7	86.6	50.4	35.1	86.6	40.5	59.6	86.6	68.8
	English – Comb	**51.5**†	**99.1**	**51.9**	–	–	–	–	–	–
	French – Direct	65.2	**99.8**	65.3	32.2	100.0	32.2	81.8	100.0	81.8
	French – Prop	48.9	76.7	63.8	39.7	76.7	51.7	64.6	76.7	84.3
	French – Comb	**66.3**	**99.8**	**66.4**	–	–	–	–	–	–
	German – Direct	59.2	99.6	59.4	22.9	100.0	22.9	76.5	100.0	76.5
	German – Prop	49.2	85.1	57.8	46.3	85.1	54.5	68.7	85.1	80.8
	German – Comb	**59.4**	**99.7**	**59.5**	–	–	–	–	–	–
	Spanish – Direct	**46.5**	**99.6**	46.7	24.8	100.0	24.8	65.0	100.0	65.0
	Spanish – Prop	38.5	79.2	**48.6**	27.4	79.2	34.6	58.9	79.2	74.3
	Spanish – Comb	**46.5**	**99.6**	46.7	–	–	–	–	–	–
	Portuguese – Direct	59.3	**99.4**	59.7	32.8	100.0	32.8	77.6	100.0	77.6
	Portuguese – Prop	56.4	87.7	64.3	43.2	87.7	49.2	71.9	87.7	82.0
	Portuguese – Comb	**63.2**†	**99.4**	**63.5**	–	–	–	–	–	–
	Italian – Direct	63.9	99.4	64.3	36.2	100.0	36.2	81.8	100.0	81.8
	Italian – Prop	54.1	82.2	65.8	41.8	82.2	50.8	69.5	82.2	84.5
	Italian – Comb	**67.7**†	**99.6**	**68.0**	–	–	–	–	–	–
	Macro-avg – Direct	57.3	**99.5**	57.6	30.8	**100.0**	30.8	**75.7**	**100.0**	75.7
	Macro-avg – Prop	48.5	82.9	58.5	**38.9**	82.9	**46.9**	65.5	82.9	**79.1**
	Macro-avg – Comb	**59.1**†	**99.5**	**59.3**	–	–	–	–	–	–
FewRel	Direct	48.7	**100.0**	48.7	**42.2**	**100.0**	42.2	**67.7**	**100.0**	67.7
	Prop	40.2	75.8	**53.1**	40.3	75.8	**53.2**	51.5	75.8	**68.0**
	Comb	**50.0**†	**100.0**	50.0	–	–	–	–	–	–

Table 1: Results on the SMiLER and FewRel benchmarks. † indicates a statistically significant improvement of Comb over Direct (bootstrap test, $p < 0.05$). Atomization significantly improves relation recall. But entity recall decreases. The accuracy increases for the small LLM (Qwen3-0.6B). Bold indicates the best result per extractor column for each benchmark/language.

granularity for GLiREL, which achieves a higher F1 score when using the propositioner. However larger models (`Qwen3-4B`) reach higher scores (precision, recall and F1). Thus, for larger documents, larger models are preferable to the propositioner. We hypothesize that this difference in score could be reduced by considering the constructed graph and leveraging relation transitivity. Deeper experiments should be conducted to verify this assumption.

Evaluation on triplet extraction on openIE context. As shown in Table 3, we observe that using the propositioner upstream of CoreNLP (Prop method) improves recall and AUC, but not precision. This drop in precision can be explained by the increased number of triplets extracted by the atomic propositions method. We observe that in OpenIE context, the union of propositions and original text is optimal for recall and AUC, having higher Recall and AUC for all pipelines.

6.2. Qualitative Results

Even with no errors in the pipeline components, it remains possible that some transitive relations present in the input text are absent from the knowledge graph built from the triplets.

Due to the atomic splitting of information during intermediary representation translation, such relations may become latent. However, it is important to note that these propositions can nevertheless be recovered through the use of inference, which is supported by the formal properties of the atomic propositions.

For instance, when the following sentence is entered:

`"Šafov is a village and municipality (obec) in Znojmo District in the South Moravian Region of the Czech Republic.",`

the propositioner's output is as follows:

`[..., "Šafov is located in Znojmo District.", "Znojmo District is located in the South Moravian Region.", "The South Moravian Region is located in the Czech Republic.", ...]`

The corresponding graph, displayed in Figure 5, illustrates the aforementioned transitive relation.

	GLiREL			Qwen3-0.6B			Qwen3-4B		
Pipeline	P	R	F1	P	R	F1	P	R	F1
Sentence	3.7	9.4	5.3	22.6	8.3	12.1	**36.6**	13.3	19.5
Document	1.4	14.8	2.5	20.3	**15.7**	**17.7**	32.0	**24.4**	**27.7**
Proposition	**5.9**	6.9	**6.3**	**25.9**	6.7	10.6	36.5	9.4	14.9
Sentence+Proposition	3.8	12.3	5.8	–	–	–	–	–	–
Document+Proposition	1.6	**18.1**	2.9	–	–	–	–	–	–

Table 2: Results on the DocRED benchmark. Atomization significantly improves precision and F1 when using a weaker extractor. However, when using LLM extraction, performance decreases.

	CoreNLP				Qwen3-0.6B				Qwen3-4B			
Pipeline	P	R	F1	AUC	P	R	F1	AUC	P	R	F1	AUC
Direct	**17.6**	24.3	20.4	0.144	**29.4**	7.9	12.4	0.051	**46.2**	33.3	**38.7**	0.244
Prop	16.4	**31.3**	**21.5**	**0.182**	24.0	11.1	15.2	0.069	31.6	35.0	33.2	0.230
Union	–	–	–	–	23.7	**14.7**	**18.1**	**0.091**	30.8	**43.6**	36.1	**0.285**

Table 3: Results on the CaRB benchmark. Atomization significantly improves recall and F1 when a weaker extractor is used. However, when using LLM extraction, performance decreases.

Precisely, while triplet (“Šafov”, “hasLocation”, “Czech Republic”) is not directly extracted from propositions, we could deduce it by taking into account transitivity.

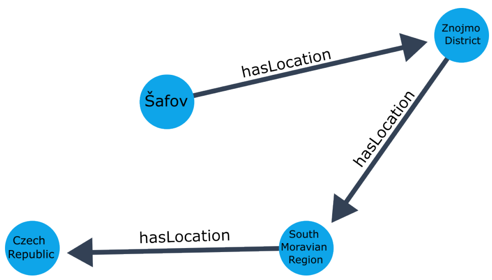

Figure 5: The Knowledge Graph built with triplets extracted by `GLiREL` on the atomic propositions for the above sentence.

7. Conclusion

In this work, we empirically demonstrate the benefits of the atomic proposition for triplet entity relation extraction. In particular, we show that decomposing documents or paragraphs into atoms helps retrieve the relation efficiently, improving performance on both the FewRel and SMiLER benchmarks.

In addition, we observe that the combination of original input text with atomised text yields better performance across all metrics. The results obtained on both the CaRB and DocRED benchmarks validate the previous hypothesis, showing better recall performance.

However, future studies should be conducted to strengthen the methods, especially a human evaluation of the propositioner, to ensure that text units are indeed all atomic according to the definition (see Section 3).

Additionally, we think that such triplet extraction methods and the constructed graph could be used in applications such as information retrieval systems.

We think that the interpretability of such text units (atoms) could provide a statistical basis for creating or deducing a logical rule, which could benefit the KG traversal algorithm. These research leads are left for further work.

8. Limitations

One limitation of this study is the relevance of the recursive propositioner. To date, we have not compared propositions obtained with and without recursive refinement. The decision to use such an algorithm to refine atoms recursively was based on preliminary experiments in which we observed that some propositions were not atomic. In future works, we plan to compare the recursive propositioner to the non-recursive one.

Another limitation is the absence of evaluation using larger LLMs than `Qwen3-4B` for fair comparison and computational efficiency. Nonetheless, it is worth noticing that a larger model could be compared in future studies.

9. Bibliographical References

Yehoshua Bar-Hillel and Rudolf Carnap. 1953. Semantic information. *The British Journal for the Philosophy of Science*, 4(14):147–157.

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.

Jack Boylan, Chris Hokamp, and Demian Gholipour Ghalandari. 2025. GLiREL - generalist model for zero-shot relation extraction. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8230–8245, Albuquerque, New Mexico. Association for Computational Linguistics.

Tong Chen, Hongwei Wang, Sihao Chen, Wenhao Yu, Kaixin Ma, Xinran Zhao, Hongming Zhang, and Dong Yu. 2024. Dense X retrieval: What retrieval granularity should we use? In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 15159–15177, Miami, Florida, USA. Association for Computational Linguistics.

René Cori and Daniel Lascar. 2003. *Logique mathématique : cours et exercices corrigés, Tome 1 - Calcul propositionnel, algèbres de Boole, calcul des prédicats*. Sciences Sup. Dunod, Paris. Avec la collaboration de Jean-Louis Krivine.

Oren Etzioni, Michele Banko, Stephen Soderland, and Daniel S. Weld. 2008. Open information extraction from the web. *Commun. ACM*, 51(12):68–74.

Wikimedia Foundation. Wikimedia downloads.

Xu Han, Hao Zhu, Pengfei Yu, Ziyun Wang, Yuan Yao, Zhiyuan Liu, and Maosong Sun. 2018. FewRel: A large-scale supervised few-shot relation classification dataset with state-of-the-art evaluation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4803–4809, Brussels, Belgium. Association for Computational Linguistics.

Oktie Hassanzadeh. 2021. Building a knowledge graph of events and consequences using wikidata. *Wikidata@ ISWC*, 2982.

Tanguy Herserant and Vincent Guigue. 2025. Seval-ex : Un paradigme basé sur les phrases atomiques pour une évaluation explicable de la qualité des résumés. In *Actes de la 20e Conférence en Recherche d'Information et Applications, CORIA 2025, Marseille, France, Juin 30 - Juillet 4, 2025*, pages 217–229. ATALA&ARIA.

Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the knowledge in a neural network. *ArXiv*, abs/1503.02531.

Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia D'amato, Gerard De Melo, Claudio Gutierrez, Sabrina Kirrane, José Emilio Labra Gayo, Roberto Navigli, Sebastian Neumaier, Axel-Cyrille Ngonga Ngomo, Axel Polleres, Sabbir M. Rashid, Anisa Rula, Lukas Schmelzeisen, Juan Sequeda, Steffen Staab, and Antoine Zimmermann. 2021. Knowledge graphs. *ACM Comput. Surv.*, 54(4).

Pere-Lluís Huguet Cabot, Simone Tedeschi, Axel-Cyrille Ngonga Ngomo, and Roberto Navigli. 2023. Red$^{\text{fm}}$: a filtered and multilingual relation extraction dataset. In *Proc. of the 61st Annual Meeting of the Association for Computational Linguistics: ACL 2023*, Toronto, Canada. Association for Computational Linguistics.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. FActScore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12076–12100, Singapore. Association for Computational Linguistics.

Makoto Miwa, Rune Sætre, Yusuke Miyao, and Jun'ichi Tsujii. 2010. Entity-focused sentence simplification for relation extraction. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 788–796, Beijing, China. Coling 2010 Organizing Committee.

Christina Niklaus, Bernhard Bermeitinger, Siegfried Handschuh, and André Freitas. 2016. A sentence simplification system for improving relation extraction. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: System Demonstrations*, pages 170–174, Osaka, Japan. The COLING 2016 Organizing Committee.

Alessandro Seganti, Klaudia Firląg, Helena Skowronska, Michał Satława, and Piotr Andruszkiewicz. 2021. Multilingual entity and relation extraction dataset and model. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1946–1955, Online. Association for Computational Linguistics.

Joe Stacey, Pasquale Minervini, Haim Dubossarsky, Oana-Maria Camburu, and Marek Rei. 2023. Atomic inference for nli with generated facts as atoms. In *Conference on Empirical Methods in Natural Language Processing*.

Andra Waagmeester, Gregory Stupp, Sebastian Burgstaller-Muehlbacher, Benjamin M Good, Malachi Griffith, Obi L Griffith, Kristina Hanspers, Henning Hermjakob, Toby S Hudson, Kevin Hybiske, et al. 2020. Wikidata as a knowledge graph for the life sciences. *Elife*, 9:e52614.

Zhishang Xiang, Chuanjie Wu, Qinggang Zhang, Shengyuan Chen, Zijin Hong, Xiao Huang, and Jinsong Su. 2026. When to use graphs in rag: A comprehensive analysis for graph retrieval-augmented generation.

Yuan Yao, Deming Ye, Peng Li, Xu Han, Yankai Lin, Zhenghao Liu, Zhiyuan Liu, Lixin Huang, Jie Zhou, and Maosong Sun. 2019. DocRED: A large-scale document-level relation extraction dataset. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 764–777, Florence, Italy. Association for Computational Linguistics.

A. Prompts

A.1. Evaluation Prompts

Closed IE (SMiLER, FewRel, DocRED). The following prompt is used to classify the relation between two identified entities:

```
Given the text, identify the relation
between the two entities.

Text: {text}
Entity 1: {e1}
Entity 2: {e2}

Choose exactly one relation from
this list: {labels}

Answer with just the relation name,
nothing else.
```

Open IE (CaRB). The following prompt is used for open-domain triplet extraction:

```
Extract all factual
(subject, predicate, object)
triples from the sentence.
One triple per line in the format:
subject | predicate | object
No explanations. If no triple can be
extracted, write nothing.

Sentence: {text}
```

A.2. Propositioner Training Prompt

The following prompt is used during the knowledge distillation training of `MPropositionneur-V2`. The teacher model (`Qwen3-32B`) is prompted to decompose a Wikipedia passage into fully autonomous atomic propositions, which are then used as targets for the student model (`Qwen3-0.6B`).

```
You are an expert in disambiguation
and information extraction.
You must decompose the text into
atomic propositions (single facts)
that are FULLY AUTONOMOUS.

ABSOLUTE RULES:
1. ZERO PRONOUNS: "He", "She", "They",
   "His", "Her", "Its", "This one"
   ARE FORBIDDEN.
   ALWAYS replace them with the
   full name of the entity.
2. CONTEXT: Each sentence must be
   readable alone without knowing
   its source.
3. REPETITION: Repeat the subject
   in EACH sentence.

OUTPUT FORMAT: Only a JSON array of
strings.

Title: {title}
Content: {content}
Output:
```

B. Proofs for the Formal Grounding

Definition 1 (Safe Cut)**.** *A formula's cut ϕ in a formula ψ is **safe** if ψ is a sub-formula of ϕ and if $I(\phi) > I(\psi)$*

Definition 2 (Bad cut)**.** *A cut of a formula ϕ in a formula ψ is **bad** if ψ is a sub-formula of ϕ and if $I(\phi) \leq I(\psi)$*

Lemma 1 (Divisibility of Conjunction — Lemma)**.** *Let $\phi = A \wedge B$ where A and B are logically independent. Extracting the component A is a **strictly safe** operation.*

Proof. By definition of conjunction, $\mathrm{Cont}(A \wedge B) = \mathrm{Cont}(A) \cap \mathrm{Cont}(B)$. Because A and B are independent, $\mathrm{Cont}(A \wedge B) \subsetneq \mathrm{Cont}(A)$. By monotonicity of μ, we have $\mu(\mathrm{Cont}(A \wedge B)) < \mu(\mathrm{Cont}(A))$. The function $-\log_2$ is strictly decreasing, so:

$$I(A \wedge B) > I(A).$$

The information content of A is strictly less than that of $A \wedge B$: the operation is strictly safe. $\square$

Lemma 2 (Indivisibility of Disjunction — Lemma)**.** *Let $\phi = A \vee B$ where A and B are logically independent. Extracting the component A is a **bad** operation.*

Proof. By definition of disjunction, $\mathrm{Cont}(A \vee B) = \mathrm{Cont}(A) \cup \mathrm{Cont}(B)$. We have the strict inclusion $\mathrm{Cont}(A) \subsetneq \mathrm{Cont}(A \vee B)$, hence $\mu(\mathrm{Cont}(A)) < \mu(\mathrm{Cont}(A \vee B))$, which yields:

$$I(A) > I(A \vee B).$$

The information content of A is strictly greater than that of $A \vee B$: decomposing a disjunction hallucinates a more specific fact than what was originally stated. $\square$

Case of implication . Cutting $A \to B$ into A is bad, because $A \to B \equiv \neg A \vee B$, and Lemma 2 applies directly by substitution.

Theorem 1 (Structural characterisation — Theorem)**.** *A formula ϕ is **atomic** if and only if it is logically equivalent to a **clause** (finite disjunction of literals).*

Proof. Let ϕ be in Conjunctive Normal Form (CNF) :

$$\phi \equiv C_1 \wedge C_2 \wedge \cdots \wedge C_n,$$

where each C_i is a clause (disjunction of literals).

(⇒) Let ϕ be atomic. *Ad absurdum*, let $n \geq 2$. By Lemma 1, the operation $\phi \mapsto C_1$ is strictly safe, which contradicts atomicity. Therefore, $n = 1$: ϕ is a single clause.

(⇐) Let $\phi = L_1 \vee \cdots \vee L_k$. By Lemma 2 (generalised by induction on k), all strict sub-formulae have greater information content than ϕ. Therefore, every cut is bad, and ϕ atomic. $\square$