DocIE@XLLM25: In-Context Learning for Information Extraction using Fully Synthetic Demonstrations

Nicholas Popovič **Ashish Kangen** **Tim Schopf** **Michael Färber**
ScaDS.AI & TU Dresden, Germany
{nicholas.popovic,ashish_yashwanth.kangen,tim.schopf,michael.faerber}@tu-dresden.de

Abstract

Large, high-quality annotated corpora remain scarce in document-level entity and relation extraction in zero-shot or few-shot settings. In this paper, we present a fully automatic, LLM-based pipeline for synthetic data generation and in-context learning for document-level entity and relation extraction. In contrast to existing approaches that rely on manually annotated demonstrations or direct zero-shot inference, our method combines synthetic data generation with retrieval-based in-context learning, using a reasoning-optimized language model. This allows us to build a high-quality demonstration database without manual annotation and to dynamically retrieve relevant examples at inference time. Based on our approach we produce a synthetic dataset of over $5k$ Wikipedia abstracts with approximately $59k$ entities and $30k$ relation triples. Finally, we evaluate in-context learning performance on the DocIE shared task, extracting entities and relations from long documents in a zero-shot setting. We find that in-context joint entity and relation extraction at document-level remains a challenging task, even for state-of-the-art large language models.

1 Introduction

Information extraction (IE) is a key task in natural language processing (NLP) research. While significant progress has been made in terms of NLP and IE benchmarks in general, few-shot and long context tasks remain relevant and challenging (Tan et al., 2022; Popovic and Färber, 2022; Gui et al., 2024; Xue et al., 2024; Diaz-Garcia and Lopez, 2024; Yilahun et al., 2025), even in the age of large language models (LLMs). In this work, we explore this direction in the context of the Document-level Information Extraction (DocIE) shared task (Organizers), which challenges systems to perform joint entity and relation extraction over long, unstructured documents in a zero-shot setting.

Specifically, we approach the topic via two recently popular approaches, LLMs which have been optimized for reasoning-heavy tasks (DeepSeek-AI et al., 2025), as well as synthetic data augmentation, which has become popular for IE in particular (Li et al., 2023; Josifoski et al., 2023; Rogulsky et al., 2024) as scalable data annotation still represents a major challenge. In order to combine the two, we construct a simple, retrieval-based in-context learning setting in which an LLM is tasked with extracting entities and relations from a text based on a single example demonstration retrieved based on its similarity to the given text. In order to preserve the zero-shot setting, we add the constraint that the example demonstration must not be manually annotated, but instead is a synthetically generated example. We therefore develop an approach for a synthetic data generation pipeline which produces high quality annotated examples of schema-constrained entity and relation extraction. Our evaluations on the shared task, as well as the Re-DocRED (Tan et al., 2022) dataset show that in-context joint entity and relation extraction at the document-level remains a challenging task, even for state-of-the-art LLMs.

Our contributions in this paper are the following:

- We propose a fully automatic pipeline for synthetic data generation based on LLMs.
- We apply our pipeline to Wikipedia abstracts and produce a dataset of roughly $5k$ documents annotated with approximately $59k$ entities and $30k$ triples, which we make available for future research[1].
- We present the evaluation of an in-context pipeline which relies on our synthetic demon-

[1]The code and synthetic dataset are made available at https://github.com/nicpopovic/docie-xllm25.

strations for in-context learning on the DocIE shared task (Organizers).

2 Task Description

The DocIE shared task (Organizers) evaluates long form information extraction, specifically entity and relation extraction, in a zero-shot schema-constrained setting: At test time, a system is provided with only a text document and a schema consisting of strings naming the entity and relation types which are to be extracted. As is typical for few-shot and zero-shot tasks, the supplied training and development data resembles the test data only in structure. It differs in terms of schemata and text domains.

3 Approach

For this paper, we are interested specifically in putting several key technologies to the test in a pipeline, namely retrieval-based in-context learning, synthetic data (Li et al., 2023; Josifoski et al., 2023; Rogulsky et al., 2024), and reasoning language models (DeepSeek-AI et al., 2025). Further, we apply our pipeline to the zero-shot setting using none of the provided training data and foregoing any model fine-tuning.

The core idea behind our approach is to construct a fully synthetic demonstration database, from which, given a query, we retrieve the most relevant example and provide it to a reasoning language model as an in-context example. Below, we first describe the inference pipeline, as it could also be applied to manually annotated data, before describing the synthetic data generation pipeline in detail in Section 4.

3.1 Inference Pipeline

For inference, and thus the evaluation of the DocIE shared task, we use an in-context learning setting split into two LLM calls[2]. For the in-context examples, we use a single example document from our synthetic demonstration database, retrieved based on similarity to the query document. Both calls use the same prompt structure, given in Appendix A, Figure 4. In the first call, we query the model with just the first paragraph of the query document, while the second LLM call supplies the entire document with the annotations provided for the beginning paragraph. We find that this strategy drastically decreases failures of the model to adhere to the annotation format for long documents.

4 Synthetic Data Generation

Figure 1 provides an overview of the synthetic data generation pipeline which requires only text data as its input and produces complete, high quality annotations over a two-phase process.

4.1 Text Data Collection

As a basis for the synthetic dataset we use Wikipedia's Vital Articles (Level 4)[3], which is a collection of 10k articles deemed essential based on criteria such as coverage, notability, and impact on other Wikipedia content. These articles represent key topics across various domains and offer a broad scope of vital knowledge. Detailed information about the category distributions can be found in Appendix B, Figure 5. We create our final synthetic dataset from abstracts which we truncate as a result of initial experiments outlined in section 5.1.

4.2 Annotation Phase

The main annotation phase consists of an initial LLM-based annotation followed by rule-based verification of the returned results.

The initial annotation run is performed in a zero-shot setting using a reasoning language model, specifically DeepSeek-R1-Distill-Qwen2.5-32B, and a prompt provided in Appendix A, Figure 2. The prompt was developed through multiple iterations of manual engineering to meet the following criteria:

- **Zero-Shot Setting**: No use of DocIE shared task data in the prompt.[4]
- **Machine-Parseable Output**: Requiring the model to return a JSON object yields reliable results.[5]

[2] All experiments are performed using DeepSeek-R1-Distill-Qwen2.5-32B (DeepSeek-AI et al., 2025) at temperature 0.

[3] https://en.wikipedia.org/wiki/Wikipedia:Vital_articles/Level/4

[4] Note that the training data was seen by the team constructing the prompts.

[5] We find that properly escaping the input is crucial to avoid syntax errors.

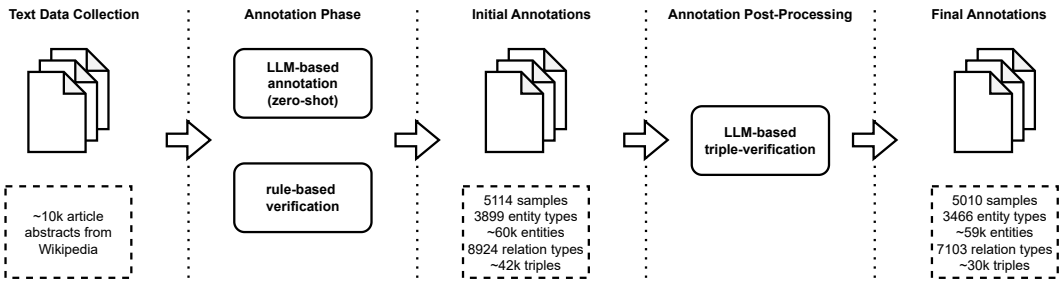

Figure 1: Overview of the pipeline used for synthetic data generation.

- **Unified Schema**: We use a single prompt for schema generation, entity extraction, and relation extraction. Defining distinct keys for each step in the JSON object ensures coverage and prevents omissions.

- **Full NER**: Includes mention detection (as spans), entity typing, and coreference resolution. Though not required for the task, character spans align with standard IE benchmarks and support future encoder-based fine-tuning. Prompting the model to echo the input with inline HTML/XML-style tags for mentions proves robust and allows for automatic validation.

- **Verification Hooks for Relations**: We prompt the model to describe each extracted triple in natural language before emitting the structured triple, enabling verification in the post-processing stage. We also require the model to use IDs to refer to the previously extracted entities, which enables automatic consistency checks (e.g., confirming that the subject and object spans actually appear in the source text).

To prioritize annotation quality over quantity and enable post-processing, we implement several verification mechanisms after the initial model output. These steps focus particularly on validating extracted relations and ensuring annotation completeness. First, we verify entities by checking that all provided mentions correctly occur within the input text. Second, we validate extracted triples by enforcing that subject and object references are made via entity IDs; we then cross-check that the names referenced in the triples match the previously annotated entities. This ensures the model does not fabricate tuples based on nonexistent mentions.

4.3 Annotation Post-Processing

During the construction of the zero-shot prompt, we identified a common source of errors in the directionality of relations (e.g., swapping the subject and object). The natural language descriptions generated as part of the verification hooks (as outlined above) help mitigate this issue in two ways: First, in our observations, producing a description already improves the accuracy of the extracted triples. Second, these descriptions enable further verification.

For each extracted triple, we prompt the model (using the template shown in Appendix A, Figure 3) to assess whether the structured triple faithfully matches its corresponding natural language description. If an inconsistency is detected, we discard not only the affected triple but the entire set of relations of that type within the document, to avoid including problematic relation types. This conservative approach prioritizes annotation quality over quantity, in line with our overall dataset construction philosophy.

In addition to the triple filtering, we discard annotations in two cases, first if assigned entity types are identical to the entities themselves and second if all entities in a document have the same type.

5 Experiments and Results

5.1 Effects of Text Length on Zero-Shot Annotation

Instead of using full abstracts for the synthetic dataset, we truncate the input texts for two reasons: First, shorter texts reduce inference time per document, enabling the annotation of a more diverse set of articles. Second, initial observations indicated that longer texts led to a higher frequency of verification failures when using our prompt.

In order to validate the latter observation, we ran an initial annotation round for all abstracts with a length of up to 300 words and tracked error rates. The results of this experiment, shown in Appendix C, Figure 6, reveal the following:

- Text length and total verification failures are highly correlated.
- Syntax errors, making the output unparseable and mismatches between the entity IDs used in the triples and the ones extracted from the text are the most common errors. Failures in the entity extraction step, such as extracted entities not occuring in the text or missing span annotations, are less common.

5.2 Statistics of Synthetic Annotation Database

Based on the results outlined above, to balance inference efficiency, annotation yield, and text length (especially considering that downstream queries will involve longer documents), we implement the following truncation strategy for the remaining data: We split each abstract into sentences and iteratively add sentences until the cumulative text length has reached 100 words or more and then omit the remaining sentences. Additionally, upon failed verification, we rerun the zero-shot prompt once with a temperature of 0.2. This results in 5114 annotated documents, a yield of 52.89% for the initial annotation. After the post-processing phase, the final dataset consists of 5010 documents, with approx. $59k$ across 3466 entity types and approx. $30k$ triples across 7103 relation types. The most common entity and relation types are shown in Appendix D Figures 7 and 8. An example of synthetic data is shown in Appendix E, Figure 9.

5.3 DocIE Test Set Evaluation

For evaluation on the test set of the DocIE shared task (Organizers), we fetch the most similar document from our synthetic dataset based on the similarity measured between the truncated query text (truncated using the same strategy as used during our dataset construction) and the synthetically annotated documents[6] using the sentence-transformers (Reimers and Gurevych, 2019) model all-MiniLM-L6-v2[7].

Task	P (%)	R (%)	F1 (%)
Entity Ident.	52.36	23.94	32.86
Entity Class.	25.79	11.80	16.19
RE (General)	5.03	2.44	3.29
RE (Strict)	4.61	2.23	3.01

Table 1: Evaluation results on the test dataset of the DocIE shared task for entity identification, entity classification, and relation extraction under general and strict modes. Precision (P), Recall (R), and F1 are reported as percentages.

The results, shown in Table 1, place our approach in the fourth place for the shared task. We note that unparseable outputs occur in a substantial portion of documents, with only 63.91% of outputs being valid. While we initially hypothesize that this is due, in large parts, to the long documents increasing the frequency of syntax errors in the produced outputs (as this is the trend we observed in our experiments in Section 5.1) our experiments with shorter documents, outlined in Section 5.4, do not support this. Even when factoring unparseable outputs into the results, the scores remain low especially for relation extraction. This suggests that IE, especially relation extraction remains a challenging task in the age of strong LLMs.

5.4 Evaluation on Re-DocRED

Task	P (%)	R (%)	F1 (%)
Mention det.	72.20	25.91	38.13
Entity Ident.	58.61	23.67	33.72
Entity Class.	52.25	21.10	30.06
RE (General)	13.79	1.67	2.98
RE (Strict)	8.55	1.04	1.85

Table 2: Evaluation results on the test dataset of Re-DocRED (Tan et al., 2022). Since we have access to the ground truth labels for this data, we are able to also calculate the mention detection scores.

In Table 2 we show the results obtained on the test set of the Re-DocRED dataset (Tan et al., 2022). Compared to the DocIE dataset, fewer entity types are used for annotation (person, organization, location, time, number, and miscellaneous), and the documents tend to be shorter.

[6] We remove 2 documents from our synthetic dataset which are part of the test set in order to avoid contamination.

[7] https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2

For the overall results, with the exception of entity classification[8], the results are broadly comparable to those reported in the DocIE shared task. This suggests that joint entity and relation extraction at the document level remains a challenging problem for current state-of-the-art large language models, even when applied to relatively short documents.

Task	P (%)	R (%)	F1 (%)
Mention det.	72.20	72.62	72.41
Entity Ident.	58.61	63.73	61.06
Entity Class.	52.25	56.82	54.44
RE (General)	13.79	4.57	6.87
RE (Strict)	8.55	2.84	4.26

Table 3: Evaluation results on the test dataset of Re-DocRED (Tan et al., 2022), restricted to only those documents where the pipeline produced a valid output.

Even though the documents are shorter, with only 38.8% of outputs being valid, producing parseable outputs remains a major challenge. In order to assess the potential gain of addressing this issue, we measure the results using only those documents where our pipeline produced a valid output. The corresponding results are shown in Table 3.

6 Conclusion

We present a fully automatic, LLM-driven pipeline for high-quality synthetic data generation for document-level entity and relation extraction, and apply it to create a dataset of over $5k$ Wikipedia abstracts with roughly $59k$ entities and $30k$ relation triples. Through our two-phase annotation process, zero-shot prompt-based extraction followed by rule- and model-based verification, we demonstrate that automated checks substantially improve annotation fidelity, yielding a dataset that, exhibits consistency and wide coverage of entity and relation types.

Our evaluations on the DocIE shared task and Re-DocRED confirm that zero-shot IE remains a hard problem: precision and recall for both entities and relations are modest, and a large number of model outputs still fail to parse cleanly. Moreover, ambiguities in type definitions and relation directionality is a challenge for both extraction and evaluation.

By releasing our synthetic dataset we aim to provide a valuable resource for future research on few-shot and zero-shot document-level IE. We hope that these tools will catalyze further advances in robust, scalable information extraction methods.

[8]We attribute this to the smaller number and more generic kind of entity types used in Re-DocRED.

Acknowledgements

This work was funded by the Federal Ministry of Education and Research of Germany and the Saxon State Ministry for Science, Culture and Tourism as part of the Center for Scalable Data Analytics and Artificial Intelligence Dresden/Leipzig (ScaDS.AI, project ID: SCADS24B). The authors also acknowledge the computing resources provided by the high-performance computing center at the NHR Center of TU Dresden, which is supported jointly by the Federal Ministry of Education and Research and the participating state governments within the NHR framework.

References

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, and 181 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

Jose A. Diaz-Garcia and Julio Amador Diaz Lopez. 2024. A survey on cutting-edge relation extraction techniques based on language models. *Preprint*, arXiv:2411.18157.

Honghao Gui, Lin Yuan, Hongbin Ye, Ningyu Zhang, Mengshu Sun, Lei Liang, and Huajun Chen. 2024. IEPile: Unearthing large scale schema-conditioned information extraction corpus. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 127–146, Bangkok, Thailand. Association for Computational Linguistics.

Martin Josifoski, Marija Sakota, Maxime Peyrard, and Robert West. 2023. Exploiting asymmetry for synthetic training data generation: SynthIE and the case of information extraction. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1555–1574, Singapore. Association for Computational Linguistics.

Junpeng Li, Zixia Jia, and Zilong Zheng. 2023. Semi-automatic data enhancement for document-level relation extraction with distant supervision from large

language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5495–5505, Singapore. Association for Computational Linguistics.

Shared Task Organizers. XLLM ACL 2025 Shared Task-IV: Document-level Information Extraction. https://xllms.github.io/DocIE/.

Nicholas Popovic and Michael Färber. 2022. Few-shot document-level relation extraction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5733–5746, Seattle, United States. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. 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 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Steven Rogulsky, Nicholas Popovič, and Michael Färber. 2024. The effects of hallucinations in synthetic training data for relation extraction. In *Joint Proceedings of the 2nd Workshop on Knowledge Base Construction from Pre-Trained Language Models (KBC-LM 2024) and the 3rd Challenge on Language Models for Knowledge Base Construction (LM-KBC 2024) Co-Located with the 23nd International Semantic Web Conference (ISWC 2024), Baltimore, USA, November 12, 2024*, volume 3853 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Qingyu Tan, Lu Xu, Lidong Bing, Hwee Tou Ng, and Sharifah Mahani Aljunied. 2022. Revisiting DocRED - addressing the false negative problem in relation extraction. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8472–8487, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Lilong Xue, Dan Zhang, Yuxiao Dong, and Jie Tang. 2024. AutoRE: Document-level relation extraction with large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 211–220, Bangkok, Thailand. Association for Computational Linguistics.

Hankiz Yilahun, Hangtian Zhao, and Askar Hamdulla. 2025. Fredc: A few-shot relation extraction dataset for chinese. *Applied Sciences*, 15(3).

A Prompts

Figure 2 shows the full zero-shot annotation prompt. Figure 3 shows the triple verification prompt for post-processing extracted relations. Figure 4 shows the inference prompt (with in-context demonstration) used at test time.

B Category distribution of Wikipedia Vital Articles

Figure 5 shows the pie-chart distribution of Level 4 Vital Articles by domain category.

C Effects of Text Length on Errors in Zero-Shot Annotation

Figure 6 shows how error rates (syntax failures, span mismatches, etc.) vary with input text length.

D Most Common Entity and Relation Types

Figure 7 shows the top 30 entity types and their frequencies in the synthetic dataset. Figure 8 shows the top 30 relation types and their frequencies.

E Data Sample

Figure 9 shows an example from the synthetic dataset.

Zero-Shot Annotation Prompt (DeepSeek-R1-Distill-Qwen-32B)

````
Help me build a knowledge graph schema. I will provide a text and you tell me
which entity types and which relation types (properties) to add to my knowledge
graph schema to model the data in the text.
This is the text in question:

{text}

Return your answer in the following format:
```json
{
  'text_with_spans': # html annotated text where every mention and coreference
   of an entity is annotated, for example: '<ent id="0" type="Person">Alice
  </ent> (or <ent id="0" type="Person">Ali</ent> as her friends call her) knows
   <ent id="1" type="Person">Bob</ent> because <ent id="0" type="Person">she
   </ent> met <ent id="1" type="Person">him</ent> at
   <ent id="2" type="Educational institution">school</ent>.',
   'entities': [
     {'id': 0, 'name': <name_of_entity>, 'type': <type_of_entity>},
    ..., # add all entities with the types above, even if they are not relevant
     for a triple
   ],
   'triples': [
     {'description': <text_describing_triple>, 'triple_string':
     '(<name_of_subject>, <name_of_relation_type>, <name_of_object>)',
    'subject': <id_of_subject_entity>, 'predicate': <name_of_relation_type>,
     'object': <id_of_object_entity>},
     ...,
   ],
   'relation_types': [<name_of_relation_type>, ...],
   'entity_types': [<name_of_entity_type>, ...],
}
```

Make sure that for every entity type and relation type you annotate *all*
occurrences!
````

Figure 2: Prompt used for zero-shot text annotation.

Triple Verification Prompt (DeepSeek-R1-Distill-Qwen-32B)

````
Which of the following is a good description of the meaning of the sentence
"{description}"?

A:
```json
{{"subject": "{subject}", "predicate": "{predicate}", "object": "{object}"}}
```
B:
```json
{{"subject": "{object}", "predicate": "{predicate}", "object": "{subject}"}}
```
C:
Both. Only use this option if the predicate/property is a symmetric one.

D:
None of the above. Only use this option if the above are nonsensical or vastly
different from the text.

format your answer like so:
\boxed{{<A_or_B_or_C_or_D>}}
````

Figure 3: Prompt used for post-processing of triples.

Inference Prompt (DeepSeek-R1-Distill-Qwen-32B)

````
Help me build a knowledge graph. I will provide a text and you annotate it.
Here is what correct annotation looks like:
```json
{
    'text': '...',
    'entity_types': [...],
    'text_with_spans': '...',
    'entities': [...],
    'relation_types' + ": [...],
    'relations': [...]
}
```

(Note how the entity ids start from 0 and allow for coreference resolution,
as multiple spans in the annotated text can refer to the same entity.)

Here is the annotation I want you to complete:
```json
{
    'text': '...',
    'entity_types': [...],
    'text_with_spans': '...',
    'entities': [...],
    'relation_types' + ": [...],
    'relations': [...]
}
```

Do not add any entity or relation types! Use only the ones provided in the JSON.
Where possible, reuse the entity ids from the annotation I've started.
If I've missed any entities (or failed to resolve coreferences) or triples,
please fix accordingly.
Return the completed JSON, not just your changes.
````

Figure 4: Prompt used for inference. An example of an in-context demonstration is shown in Appendix E, Figure 9.

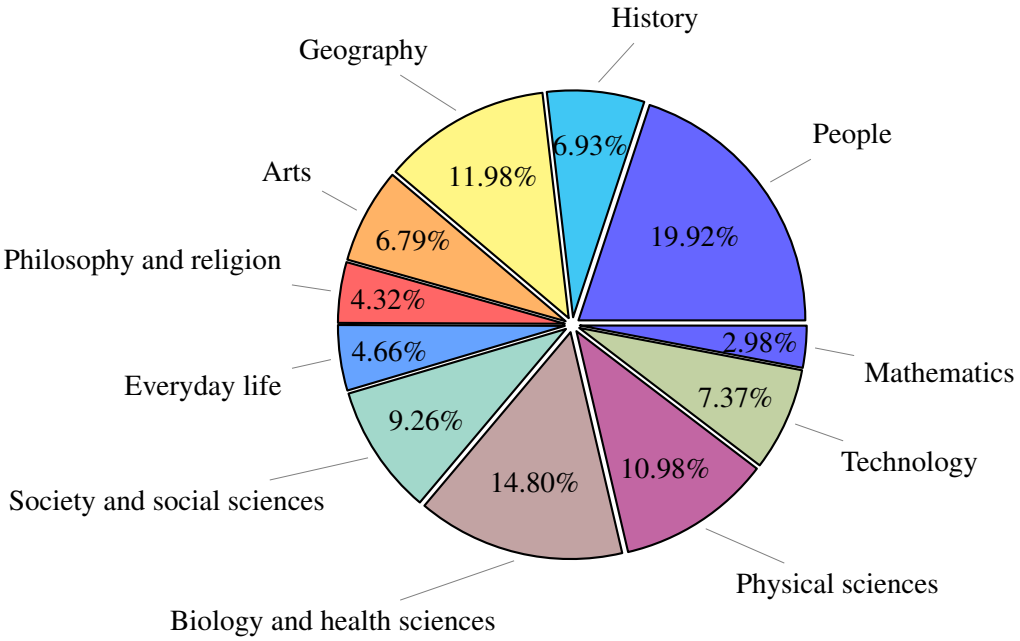

Figure 5: Category distribution of level 4 vital articles (approx. 10k).

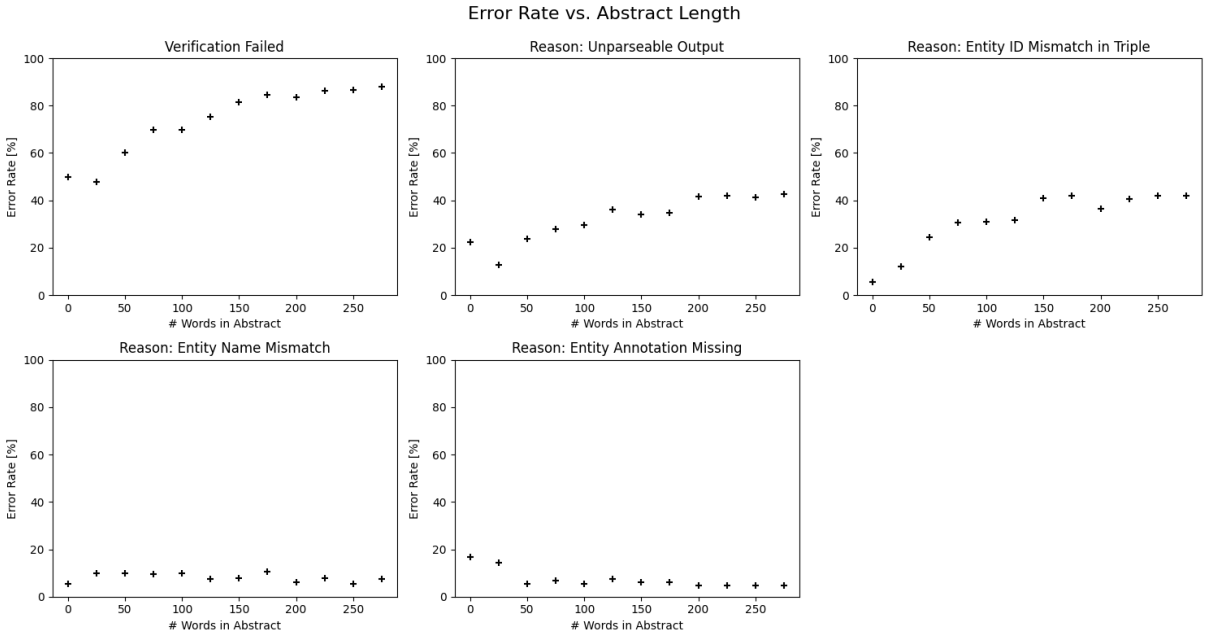

Figure 6: Error distribution for initial set of experiments of first annotation phase.

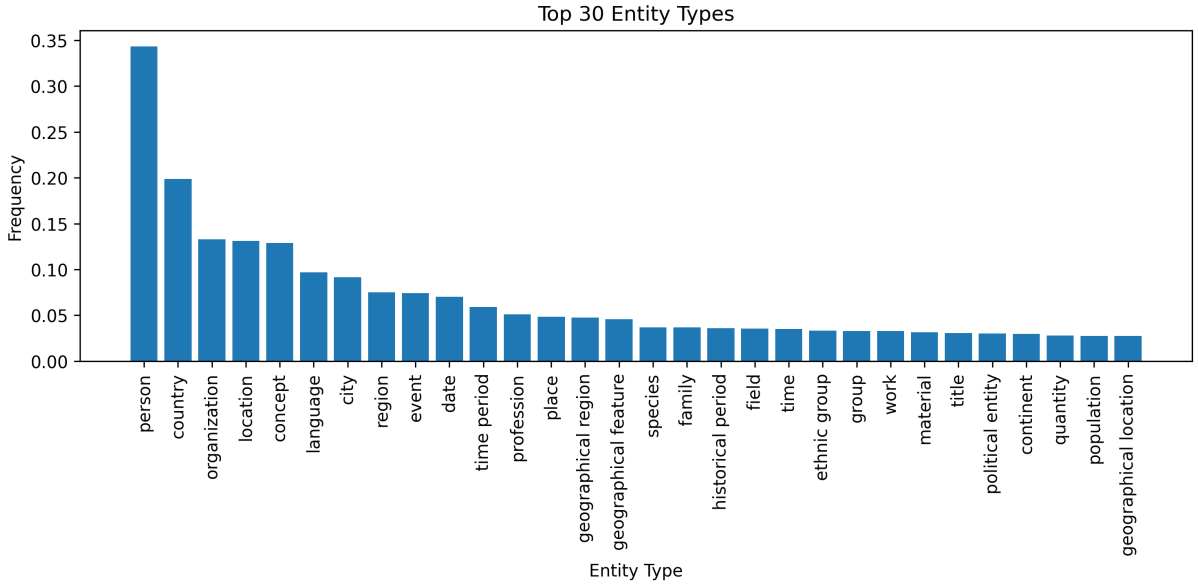

Figure 7: Top 30 entity types in final synthetic dataset and their frequency.

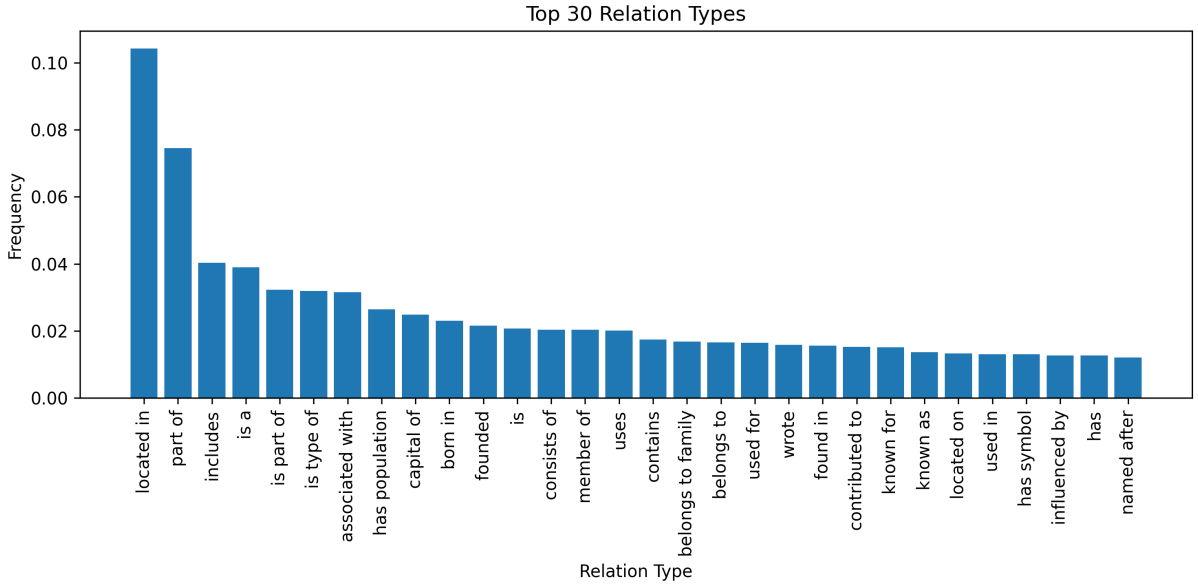

Figure 8: Top 30 relation types in final synthetic dataset and their frequency.

Synthetic Data Example


```
{
   "text": "The University of Vienna (German: Universit\u00e4t Wien) is a public research university located in Vienna, Austria. Founded by Duke
   Rudolph IV in 1365, it is the oldest university in the German-speaking world and among the largest institutions of higher learning in Europe.
    The university is associated with 17 Nobel Prize winners and has been the home to many scholars of historical and academic importance.",

     "annotated_text": "<ent id=\"0\" type=\"Educational institution\">The University of Vienna</ent> (German:
     <ent id=\"0\" type=\"Educational institution\">Universit\u00e4t Wien</ent>) is a public research university located in
     <ent id=\"1\" type=\"Location\">Vienna</ent>, <ent id=\"2\" type=\"Country\">Austria</ent>. Founded by
   <ent id=\"3\" type=\"Person\">Duke Rudolph IV</ent> in 1365, it is the oldest university in the German-speaking world and among the largest
   institutions of higher learning in Europe. The university is associated with 17 <ent id=\"4\" type=\"Award\">Nobel Prize</ent> winners and
    has been the home to many <ent id=\"5\" type=\"Person\">scholars</ent> of historical and academic importance.",

    "entities": [
        {
            "id": 0,
            "name": "The University of Vienna",
            "type": "Educational institution"
        },
        {
            "id": 1,
            "name": "Vienna",
            "type": "Location"
        },
        {
            "id": 2,
            "name": "Austria",
            "type": "Country"
        },
        {
            "id": 3,
            "name": "Duke Rudolph IV",
            "type": "Person"
        },
        {
            "id": 4,
            "name": "Nobel Prize",
            "type": "Award"
        },
        {
            "id": 5,
            "name": "scholars",
            "type": "Person"
        }
    ],

    "relations": [
        {
            "description": "The University of Vienna is located in Vienna.",
            "triple_string": "(The University of Vienna, located_in, Vienna)",
            "subject": 0,
            "predicate": "located_in",
            "object": 1
        },
        {
            "description": "The University of Vienna is located in Austria.",
            "triple_string": "(The University of Vienna, located_in, Austria)",
            "subject": 0,
            "predicate": "located_in",
            "object": 2
        },
        {
            "description": "The University of Vienna was founded by Duke Rudolph IV.",
            "triple_string": "(The University of Vienna, founded_by, Duke Rudolph IV)",
            "subject": 0,
            "predicate": "founded_by",
            "object": 3
        },
        {
            "description": "The University of Vienna has been the home to scholars.",
            "triple_string": "(The University of Vienna, home_of, scholars)",
            "subject": 0,
            "predicate": "home_of",
            "object": 5
        }
    ]
}
```


Figure 9: Example of final synthetic data.