# GUIDEX: Guided Synthetic Data Generation for Zero-Shot Information Extraction

**Neil De La Fuente**[1,3] **Oscar Sainz**[1,2] **Iker García-Ferrero**[1,2] **Eneko Agirre**[1,2]

[1]HiTZ Basque Center for Language Technology - Ixa NLP Group
[2]University of the Basque Country (UPV/EHU) [3]Technical University of Munich (TUM)

neil.de@tum.de

## Abstract

Information Extraction (IE) systems are traditionally domain-specific, requiring costly adaptation that involves expert schema design, data annotation, and model training. While Large Language Models have shown promise in zero-shot IE, performance degrades significantly in unseen domains where label definitions differ. This paper introduces GUIDEX, a novel method that automatically defines domain-specific schemas, infers guidelines, and generates synthetically labeled instances, allowing for better out-of-domain generalization. Fine-tuning Llama 3.1 with GUIDEX sets a new state-of-the-art across seven zero-shot Named Entity Recognition benchmarks. Models trained with GUIDEX gain up to 7 F1 points over previous methods without human-labeled data, and nearly 2 F1 points higher when combined with it. Models trained on GUIDEX demonstrate enhanced comprehension of complex, domain-specific annotation schemas. Code, models, and synthetic datasets are available at neilus03.github.io/guidex.com

## 1 Introduction

Information Extraction (IE) tasks (Grishman, 1997) are structured around two core components: a formal schema specifying target entities/relations and human-readable guidelines defining their interpretation. Despite their utility, IE systems face significant scalability challenges due to domain dependence. Adapting to new domains requires substantial resources, including (1) domain experts to design schemas and annotation rules, (2) trained annotators to label data accordingly, and (3) machine learning specialists to develop performant models. This complex process creates a bottleneck for real-world applications where label definitions frequently evolve across contexts.

Early attempts to remove the need for annotated data (zero-shot IE) framed the task through

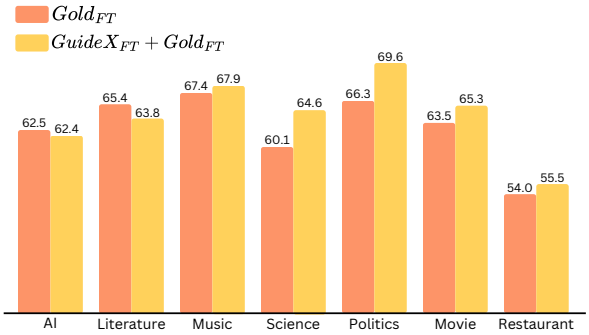


Figure 1: Impact of GUIDEX for zero-shot NER on different domains. In $Gold_{FT}$, the model is trained solely on gold training data, whereas in $GuideX_{FT} + Gold_{FT}$ it is also trained on our synthetic dataset.

Question Answering (Levy et al., 2017) or Natural Language Inference (Obamuyide and Vlachos, 2018; Sainz et al., 2021) paradigms, leveraging supervised data from these auxiliary tasks. While showing initial promise, these methods remained constrained by their reliance on manually crafted schemas and limited cross-domain generalization.

More recent advancements leveraging Large Language Models (LLMs) have streamlined schema definition processes (Li et al., 2024), facilitating adaptation to new domains. However, performance gaps persist when deploying these systems in unseen domains where label semantics diverge from training distributions (Sainz et al., 2024). Our work addresses this challenge by automating the generation of domain-specific schemas, guidelines, and annotated data to bridge the seen-unseen domain divide.

Data Augmentation (Feng et al., 2021) and Synthetic Data Generation (Teknium, 2023; Xu et al., 2025) have proven particularly valuable in the era of LLMs. In IE, traditional techniques like distant supervision (Mintz et al., 2009) aim to enhance model performance but often introduce noisy annotations. While effective for achieving high recall, distant supervision frequently generates spurious



*Findings of the Association for Computational Linguistics: ACL 2025*, pages 24248–24262

labels due to its reliance on imperfect heuristics. Similarly, LLM distillation (Hinton, 2015)—where smaller student models learn from larger teacher models—faces inherent limitations: student performance is constrained by the teacher's capabilities and often fails to capture all valid instances. Both methods also require predefined annotation schemas, severely limiting their adaptability to novel domains where label definitions may shift.

In this paper, we tackle the aforementioned limitations by introducing GUIDEX, a novel data generation method inspired by the work of domain experts. It is designed to **generate schemas, guidelines and annotated examples for any new domain** which allows to improve IE performance on new as well as on unseen domains. This approach consists of four main steps. Given a set of documents from the target domain, an LLM is used to identify the key information within each document, summarizing and synthesizing its content into a set of bulleted ideas. Next, the extracted information is structured into a standardized format, typically a JSON file. Then, the model is asked to generate the annotation schema and the corresponding annotation guidelines based on the previously structured annotations. This approach ensures that the annotations align with the guidelines and remain comprehensive. This step is particularly crucial, as it guarantees the correctness of the schema and significantly reduces potential annotation errors in the generated data. Finally, we ask the model to generate the final annotations following a standard code-style format.

We validated our data generation approach by training state-of-the-art models using the synthetic data produced through our methodology. When fine-tuning base models, such as Llama 3.1 (Grattafiori et al., 2024), exclusively on our dataset, we observed an average improvement of 10 F1 points across seven Named Entity Recognition (NER) benchmarks in zero-shot evaluation. Furthermore, when leveraging our data to enhance state-of-the-art approaches, we achieved a notable improvement of nearly 2 F1 points on the same benchmarks (see Figure 1), establishing a new state-of-the-art.

## 2 Related Work

In this section we review the literature related to our work. We begin by highlighting the most significant studies that involve LLMs for IE. Following that, we will focus on methods for generating synthetic data aimed at the same task.

**LLM-Based IE.** Recent advances in IE increasingly leverage LLMs to tackle tasks such as Named Entity Recognition (NER), Relation Extraction (RE), and Event Extraction (EE) in both zero-shot and few-shot settings (Brown et al., 2020; Raffel et al., 2020; Xu et al., 2024a). By prompting (Li et al., 2022; Ashok and Lipton, 2023; Wang et al., 2021; Wei et al., 2024, 2023; Xu et al., 2024b; Mo et al., 2024) or fine-tuning (Zhou et al., 2023; Lou et al., 2023a; Gui et al., 2025) these models can be guided to extract relevant spans (e.g., entities, relationships) in raw text. Methods like InstructUIE (Wang et al., 2023), and RUIE (Liao et al., 2025) build on the idea of formulating IE as an instruction-following or retrieval-based generation task, showing that well-structured prompts can improve performance without extensive human-annotated data.

A specialized subtrend emphasizes schema and guideline guided strategies, where the LLM is trained to follow explicit annotation rules (Sainz et al., 2024; Pang et al., 2023; Bai et al., 2024). These methods demonstrate that providing precise definitions and examples of valid or invalid annotations can significantly enhance zero-shot IE outcomes, reducing error rates from ambiguous token boundaries or unclear entity types. Similarly, KnowCoder (Li et al., 2024) encodes structured knowledge into LLMs to facilitate universal IE across multiple domains, underscoring the value of carefully specified label schemas. Although these guideline-oriented approaches maintain higher consistency across domains, they often depend on time-consuming, manual curation of instructions. As new tasks or domains emerge, the annotation guidelines must be updated or expanded, posing a key scalability challenge.

**Synthetic Data Generation for IE.** Distant supervision remains one of the earliest forms of synthetic data labeling, aligning knowledge-base facts with textual mentions to automate the creation of large-scale IE datasets (Mintz et al., 2009). While this technique helped address the scarcity of annotated data, it is susceptible to label noise because the mere co-occurrence of entities in a sentence does not necessarily confirm their relationship (Surdeanu et al., 2012). More recent developments incorporate multi-instance multi-label learning and noise reduction heuristics to mitigate erroneous

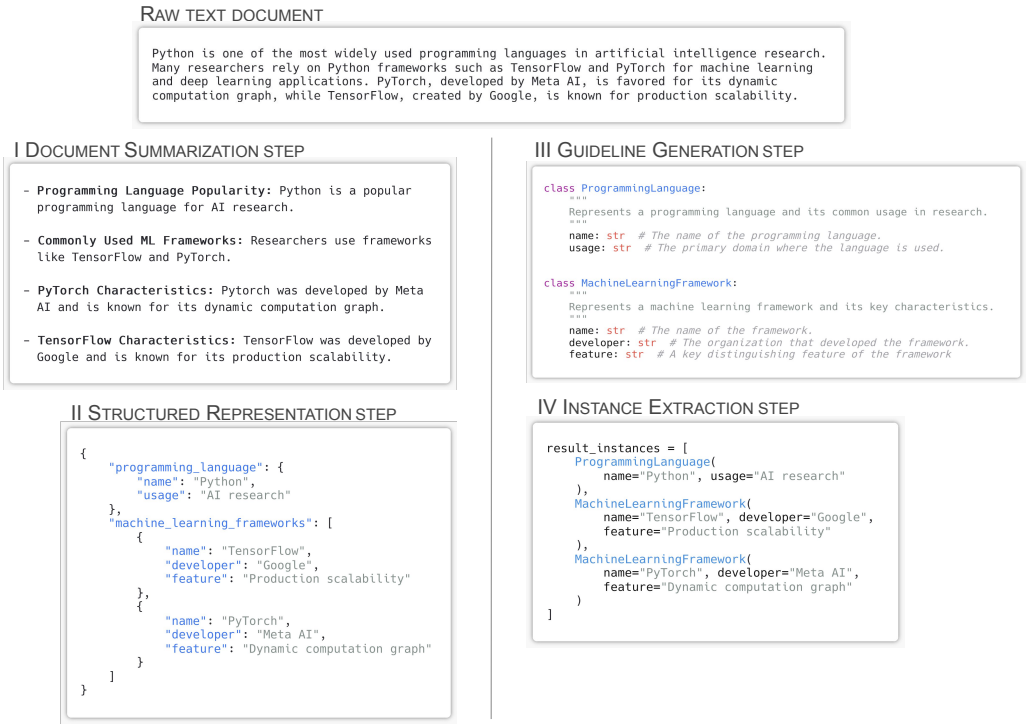


Figure 2: GUIDEX process overview. The approach transforms raw text into structured annotations by dynamically inferring schemas, generating executable guidelines, and resulting annotations.

matches (Hoffmann et al., 2011; Lin et al., 2016; Han and Sun, 2016; Xiao et al., 2020), yet these pipelines often rely on rigid predefined schemas, making it challenging to adapt to novel entity or relation types.

Beyond distant supervision, emerging methods leverage LLMs themselves to generate synthetic data for IE tasks (Josifoski et al., 2023; Chen et al., 2017). UniNER (Zhou et al., 2023) and KnowCoder (Li et al., 2024) generate additional training instances by prompting an LLM to produce sentences conforming to a particular label schema. This broadens coverage and reduces the requirement for human annotation. However, the resulting data can suffer from stylistic repetition, domain mismatch, or incomplete label application if the model fails to follow the schema consistently (Xu et al., 2024a). Moreover, existing synthetic data generation techniques often do not provide robust mechanisms to manage data structure or coverage, resulting in partially accurate guidelines and noisy annotations. Complementary lines of work employ information-theoretic criteria such as *V-information* to automatically select or weight synthetic examples, demonstrating measurable gains for classification and slot-filling tasks (Ethayarajh et al., 2022). Orthogonally, weak-supervision pipelines built on data-programming have been revisited in the LLM era; for instance, PROMPTRE combines prompting outputs with Snorkel-style label aggregation to create document-level relation corpora without manual annotation (Gao et al., 2024). These approaches reduce labeling cost but still assume a predefined schema, and thus remain complementary to our automatic schema-induction strategy.

In light of these observations, our work introduces GUIDEX, a data generation approach that integrates both schema-driven and synthetic paradigms while addressing key limitations in diversity and noise control. Rather than relying on fully manual guidelines or naive LLM-based generation, GUIDEX dynamically constructs annotation schemas and guidelines for each document, then synthesizes labeled text aligned with those rules. This approach reduces the costs of manual schema creation and mitigates the spurious annotations often seen in unconstrained synthetic data generation. Crucially, we show that incorporating the resulting dataset into zero-shot IE training significantly boosts performance across multiple benchmarks, surpassing existing approaches whether they rely on synthetic data and guidelines or not. By unifying explicit guideline creation with data synthesis, GUIDEX offers a low-noise strategy for robust zero-shot IE across increasingly diverse domains.

## 3 GUIDEX: Guided Synthetic Data Generation

In this section, we introduce GUIDEX, a structured synthetic data generation approach designed to enhance IE capabilities of LLMs. Unlike traditional LLM distillation approaches or distant supervision, which often rely on predefined annotation schemas, GUIDEX dynamically infers annotation schemas. This approach reduces annotation inconsistencies, enhances flexibility across different IE tasks, and enables high-quality, guideline-driven annotations.

As shown in Figure 2, the GUIDEX approach consists of four sequential steps: document summarization, structured representation synthesis, annotation guideline generation, and instance extraction. Prior work has shown that LLMs struggle with information extraction in zero-shot settings, and attempting to perform the entire task in a single step leads to poor results. Instead, our structured process progressively refines raw text through these four stages, ensuring high-quality, structured annotations that align with inferred guidelines. Complete prompt templates for the four stages are listed in Figure 3 in the appendix so that the readers can reproduce every step. Below, we describe each step in detail.

**Document summarization.** The first step in the pipeline focuses on identifying the most important concepts within a document. To achieve this, the LLM generates a summary highlighting the key points, effectively recognizing relevant entities and events while structuring the extracted information. Instead of relying on a predefined annotation schema, this approach allows the model to determine the relevant elements autonomously, resulting in more diverse and domain-specific annotations.

Figure 2.I presents a sample summary generated for an article discussing Machine Learning frameworks in AI research. The model successfully identifies key frameworks (*TensorFlow* and *PyTorch*), along with relevant entity details, including their developers and notable features.

**Structured representation.** IE aims to identify specific spans of text that contain relevant information. In the second step of our approach, the model leverages both the previously generated summary and the original document to organize the extracted information into a structured JSON format. This representation ensures that key elements are systematically categorized while maintaining direct references to the source text. To improve accuracy and conciseness, we enforce constraints that limit the extracted spans to the shortest possible length while preserving their full meaning.

As depicted in Figure 2.II, the extracted information is transformed into a structured JSON format, where each entity is assigned meaningful labels and attributes. This structured representation enables better organization and downstream usability, ensuring that information is both interpretable and machine-readable.

**Guideline generation.** Annotation guidelines play a critical role in improving zero-shot IE performance by ensuring consistency and reducing ambiguity in annotations (Sainz et al., 2024; Li et al., 2024). However, manually crafting such guidelines is a complex task, as it requires defining precise rules that account for variations across different domains. To address this challenge, our approach dynamically generates annotation guidelines based on the structured JSON representation, the document summary, and the original text. Instead of relying on predefined schemas, the model autonomously derives comprehensive descriptions for each entity type, ensuring that all relevant attributes are clearly captured.

Each generated guideline is formulated as a Python dataclass, embedding a long and detailed description of the entity type within the class docstring. The expected attributes of the entity are also explicitly defined, with each field accompanied by comments explaining its meaning and expected format. By structuring the annotation schema in this manner, the model produces a standardized and interpretable representation of the information.

Figure 2.III shows how this process results in a set of structured dataclasses that define both entity types and their relationships while preserving flexibility. Unlike static annotation guidelines, which may be limited in adaptability, this approach ensures that each document receives tailored guidelines that align precisely with its content. By encoding these guidelines into Python code, the approach maintains a high level of structural consistency.

**Instance Extraction** The final step of the GUIDEX approach involves extracting concrete instances of the entities and attributes defined in the annotation guidelines. Using the structured JSON representation and the inferred annotation schema, the model populates entity classes with specific values derived directly from the original

document. This step ensures that the extracted instances strictly adhere to the predefined structure and maintain high fidelity to the source text.

To achieve this, the model generates a Python list where each entry corresponds to an instance of one of the dataclasses created in the guideline generation step. Each instance is populated with concise values extracted from the text, prioritizing single words or short phrases over verbose descriptions to maximize precision and clarity. The model is explicitly instructed to return only the structured instances, without additional explanations or extraneous text.

As illustrated in Figure 2.IV, this process results in a structured dataset where each entity and its attributes are instantiated according to the inferred schema. This format ensures seamless integration into downstream applications, enabling models to leverage high-quality, structured annotations for training or evaluation. By enforcing a standardized representation of extracted instances, GUIDEX maintains consistency across different documents and domains

GUIDEX enhances synthetic data reliability by structuring annotation guidelines and extracted instances as executable Python code, enabling automated validation to detect hallucinations and inconsistencies. A consistency-checking mechanism systematically executes each dataset entry, flagging logical errors such as undefined entity types or misaligned attributes. An automated filtering script discards invalid annotations while retaining only schema-compliant ones, significantly reducing spurious relationships and annotation noise. This structured validation process ensures high-quality, guideline-aligned synthetic data, leading to more robust and trustworthy IE models.

## 4 The GUIDEX dataset

This section presents the dataset constructed using the proposed method. We begin by detailing the document collection process that served as the foundation for our dataset. Then, we provide statistical insights to illustrate its composition and characteristics.

**Document Collection.** The dataset was constructed using FineWeb-edu (Penedo et al., 2024), a high-quality subset of the larger FineWeb dataset, specifically curated for educational content. From this collection, we randomly sampled ~10,000 documents. The dataset exhibits a wide range of document lengths, spanning from 194 to 22.6k words. To preserve the coherence of the textual structure and maximize contextual understanding, we retain entire documents rather than segmenting them into smaller units such as paragraphs or sentences.

| Most common | | Least common | |
|---|---|---|---|
| **Freq.** | **Label** | **Freq.** | **Label** |
| 1820 | Symptom | 1 | MusicOrigin |
| 1459 | Benefit | 1 | AttitudesTowardsMusic |
| 929 | Resource | 1 | MusicStudy |
| 927 | Topic | 1 | DietRecommendations |
| 837 | Cause | 1 | FilterInformation |
| 830 | Location | 1 | AsyncDataSharing |
| 786 | Event | 1 | SoundMakingInformation |
| 689 | Study | 1 | MOOCDefinition |
| 679 | Treatment | 1 | MOOCDesign |
| 609 | HistoricalEvent | 1 | MOOCContent |
| 586 | Application | 1 | TeachingAndLearning |
| 572 | Activity | 1 | BenefitsAndChallenges |

Table 1: Most and Least frequent labels in the GUIDEX dataset.

**Dataset Statistics.** The GUIDEX dataset covers a diverse range of topics, as illustrated by the distribution of the most and least frequent labels in Table 1. It includes a strong presence of categories related to Medicine (*Symptom*, *Cause*, *Study*, *Treatment*), Economics (*Benefit*, *Resource*, *Application*, *Activity*), and History (*Event*, *HistoricalEvent*, *Study*). Additionally, it covers domains such as Music (*MusicOrigin*, *AttitudesTowardsMusic*, *MusicStudy*) and Education (*MOOCDefinition*, *MOOCDesign*, *TeachingAndLearning*). In total, the dataset has 28,677 unique labels, with an average of 5.34 distinct labels per document. Each document contains an average of 11.39 annotations, highlighting the dataset's richness and granularity.

**Entity–type overlap with existing corpora.** We compared the 28,677 unique entity–type names that appear in GUIDEX against the label spaces of 35 widely–used IE datasets covering NER, RE, EE, EAE and SF. Across all *train* splits we found 243 distinct gold labels, of which 103 (**42.4%**) are already present verbatim in GUIDEX. A very similar ratio holds for *test* splits (98 / 235 labels, **41.7%**). In other words, GUIDEX captures roughly two–fifths of the known label space *without* having been designed for any of those benchmarks.

Coverage is naturally uneven. Generic NER datasets such as CONLL 03, BroadTwitter, HarveyNER and BC5CDR achieve **100%** type overlap, while the CrossNER family is above 90%. Con-

versely, event–centric resources like ACE05-RE, CASIE and E3C contribute long-tail, highly specialised labels that GUIDEX does not yet include (overlap $\leq 15\%$). Taken together, these numbers show that GUIDEX already offers substantial coverage of standard NER schemas while still leaving room for complementary, task-specific labels provided by human annotation. They also explain why models benefitted most on domains whose gold labels are either absent or only partially represented in GUIDEX.

## 5 Experimental Setting

In this section we outline the details of our experiments. We begin by introducing the models used for both synthetic dataset generation and fine-tuning on manually annotated (gold) data. Next, we describe the IE datasets utilized for training and evaluation. Finally, we present the baselines and state-of-the-art systems used for comparison.

### 5.1 Models

**Synthetic data generation.** For generating synthetic data, we utilized the 70B variant of the Llama 3.1 Instruct model. After evaluating various alternatives, this model demonstrated consistent reliability in generating high-quality outputs. Although proprietary models exhibited marginally better performance in some cases, we prioritized reproducibility by selecting an open-source solution for this research.

**Model fine-tuning.** All the models we trained are based on the 8B variant of Llama 3.1 (Grattafiori et al., 2024). We adopted the standard code-style format for IE (Sainz et al., 2024; Li et al., 2024, 2023; Qi et al., 2024). To avoid negative impacts caused by the input format discrepancies, we use the base Llama 3.1 over the instruct variant.

### 5.2 Datasets

**Training Datasets.** Beyond the GUIDEX dataset, we trained our system using the same manually annotated gold-standard data as Sainz et al. (2024). This gold-standard data comes from multiple sources and covers several IE tasks including Named Entity Recognition (NER), Event Extraction (EE), Event Argument Extraction (EAE), Relation Extraction (RE), and Slot Filling (SF). Specifically, we utilized ACE 2005 (Walker et al., 2006) for NER, EE, EAE, and RE, while TACRED (Zhang et al., 2017) was employed for SF[1]. Additional NER datasets used for training include BC5CDR (Li et al., 2016), CoNLL 2003 (Tjong Kim Sang and De Meulder, 2003), DIANN (Fabregat et al., 2018), NCBDisease (Doğan et al., 2014), Ontonotes 5 (Pradhan et al., 2013), and WNUT17 (Derczynski et al., 2017).

**Evaluation Datasets.** To assess the effectiveness of our approach, we conducted evaluations on standard zero-shot NER benchmarks. Specifically, we tested on multiple CrossNER (Liu et al., 2020) splits, as well as the MIT Movie and MIT Restaurant datasets (Liu et al., 2013).

### 5.3 Baselines

We established two primary baselines: the base Llama 3.1 model and its fine-tuned counterpart using the manually annotated training datasets described in Section 5.2. From now on, we refer to the fine-tuned version as $\text{Gold}_{\text{FT}}$. These baselines will be compared to their variants pretrained on GUIDEX synthetic data: $\text{GUIDEX}_{\text{FT}}$ (trained solely on synthetic data) and $\text{GUIDEX}_{\text{FT}}$ + $\text{Gold}_{\text{FT}}$ (trained sequentially on both datasets).

Beyond these controlled comparisons, we benchmark our approach against seven state-of-the-art models. We include general-purpose conversational LLMs such as Vicuna (Chiang et al., 2023) and ChatGPT (Ouyang et al., 2022), as reported by Zhou et al. (2023). Additionally, we evaluate against specialized IE models, including USM (Lou et al., 2023b), InstructUIE (Wang et al., 2023), GoLLIE (Sainz et al., 2024), KnowCoder (Li et al., 2024), and GLiNER (Zaratiana et al., 2023). We also compare our method to UniNER (Zhou et al., 2023), a synthetic data generation-based approach for NER.

### 5.4 Implementation details

All models were trained using QLoRA (Dettmers et al., 2023), with hyperparameters optimized based on the validation splits of the training datasets (see Appendix A). We followed the same code-based input format as Sainz et al. (2024). The data generation process was conducted on four NVIDIA A100 GPUs with 80GB of memory each, while model training was performed using two GPUs of the same type.

[1]Originally designed for RE, we followed Sainz et al. (2024) in converting it into an SF task.

| GUIDEX$_{FT}$ | Gold$_{FT}$ | AI | Literature | Music | Science | Politics | Movie | Restaurant | AVERAGE |
|---|---|---|---|---|---|---|---|---|---|
| ✗ | ✗ | 24.13 | 25.83 | 33.87 | 21.67 | 31.94 | 40.86 | 32.27 | 30.08 |
| ✓ | ✗ | 35.30 ±3.2 | 42.35 ±3.6 | 40.17 ±7.73 | 29.28 ±5.3 | 36.50 ±2.0 | 31.62 ±3.9 | 44.78 ±0.9 | 37.14 ±3.4 |
| ✗ | ✓ | **62.56** ±1.7 | **65.41** ±2.0 | 67.40 ±2.77 | 60.14 ±0.2 | 66.29 ±1.9 | 63.58 ±0.8 | 53.98 ±1.3 | 62.77 ±1.2 |
| ✓ | ✓ | 62.41 ±1.2 | 63.79 ±2.8 | **67.92** ±0.5 | **64.59** ±1.1 | **69.58** ±1.3 | **65.25** ±0.9 | **55.50** ±1.3 | **64.15** ±0.7 |

Table 2: Impact of GUIDEX fine-tuning and gold fine-tuning on zero-shot NER performance with Llama 3.1 8B. GUIDEX$_{FT}$ denotes finetuning on GUIDEX data, while Gold$_{FT}$ fine-tuning on manually annotated data. ✓ shows the presence of a training stage, ✗ shows its absence. Results are reported as F1-scores on out-of-domain datasets.

## 6 Results

In this section, we discuss our experimental findings by examining GUIDEX's impact on performance and comparing our approach against the state-of-the-art.

**Impact of synthetic data.** Table 2 summarizes our main comparisons by dividing them into two scenarios: when no manually annotated data is available, and when it is. In the first scenario, we evaluate the raw impact of our dataset, particularly analyzing the extent to which it can assist a baseline LLM in learning the task. In the second scenario, we observe how the addition of manually annotated data (and thus domain-specific knowledge) further affects performance.

The top two rows of Table 2 show results when no manual annotations are available for fine-tuning. Without any task-specific data points, Llama 3.1 achieves an average F1 score of 30.08 across all seven datasets. While significant, it falls short of the best performing models. However, when trained with GUIDEX, the average F1 score improves significantly, with an increase of 7.06 points. Although not all seven tasks show improvement, those that do improve see notable gains. This demonstrates that our synthetic data effectively teaches the task to a baseline model and integrates domain-specific knowledge where it matters.

The third row illustrates the impact of manually annotated data (Gold$_{FT}$). As anticipated, these sentence-level annotations substantially boost results. Unlike our GUIDEX dataset—largely based on documents—both the training and evaluation sets in Gold$_{FT}$ focus on sentence-level IE tasks. This alignment phase allows the model to tackle sentence-specific zero-shot tasks more accurately. Even so, the improvements we achieve with Gold$_{FT}$ are complementary to those achieved with GUIDEX$_{FT}$.

Lastly, the bottom row presents a model fine-tuned first with GUIDEX$_{FT}$ and then with Gold$_{FT}$. It can be seen that both steps jointly increase performance by an average of 34 F1 points over the plain Llama 3.1 baseline (27 points over GUIDEX$_{FT}$ alone and 1.4 over Gold$_{FT}$). Moreover, applying GUIDEX$_{FT}$ to a model already trained with Gold$_{FT}$ improves on five out of the seven datasets. This indicates that the GUIDEX$_{FT}$ data can significantly enhance the model's performance in various specific domains. We analyze the impact on a label-by-label basis in Section 7.

**Comparison with the state-of-the-art.** Table 3 presents a comparison of our best-performing model, which was fine-tuned on GUIDEX$_{FT}$ and Gold$_{FT}$, against various state-of-the-art zero-shot NER systems. Our best approach achieves an average F1 score of 64.2, surpassing all other models in the benchmark. Notably, it outperforms GoLLIE by 6.2 F1 points. GoLLIE is a system similar to Gold$_{FT}$ but based on CodeLlama. Additionally, when compared to KnowCoder, another system akin to Gold$_{FT}$ that uses a pretraining dataset to better follow annotation schemas, our approach shows a 4.1-point improvement. It is worth mentioning that the pretraining proposed by Li et al. (2024) could provide complementary enhancements to our method. In addition to their overall performance, models trained on GUIDEX show strong generalization across various domains, achieving the highest F1 scores in two out of seven benchmarks: Movie (65.3), Restaurant (55.5), and being the best overall. Particularly in Politics, our model achieves a +12.4 F1 point improvement over GoLLIE (57.2), showcasing its ability to capture domain-specific nuances. Even in Music, where GLiNER-L slightly outperforms our model (69.6 vs. 67.9 F1), GUIDEX remains competitive, despite GLiNER's explicit focus on generalist NER modeling.

These results highlight the effectiveness of our approach in adapting to unfamiliar domains,

| Model | Params | Backbone | Movie | Restaurant | AI | Literature | Music | Politics | Science | Avg |
|---|---|---|---|---|---|---|---|---|---|---|
| Vicuna-7B | 7B | Llama | 06.0 | 05.3 | 12.8 | 16.1 | 17.0 | 20.5 | 13.0 | 13.0 |
| Vicuna-13B | 13B | Llama | 00.9 | 00.4 | 22.7 | 22.7 | 26.6 | 27.0 | 22.0 | 17.5 |
| USM | 0.3B | | 37.7 | 17.7 | 28.2 | 56.0 | 44.9 | 36.1 | 44.0 | 37.8 |
| ChatGPT | – | – | 05.3 | 32.8 | 52.4 | 39.8 | 66.6 | 68.5 | **67.0** | 47.5 |
| InstructUIE | 11B | FlanT5 | 63.0 | 21.0 | 49.0 | 47.2 | 53.2 | 48.1 | 49.2 | 47.2 |
| UniNER-7B | 7B | Llama | 42.4 | 31.7 | 53.6 | 59.3 | 67.0 | 60.9 | 61.1 | 53.7 |
| UniNER-13B | 13B | Llama | 48.7 | 36.2 | 54.2 | 60.9 | 64.5 | 61.4 | 63.5 | 55.6 |
| GoLLIE | 7B | CodeLlama | 63.0 | 43.4 | 59.1 | 62.7 | 67.8 | 57.2 | 55.5 | 58.0 |
| KnowCoder | 7B | Llama 2 | 50.0 | 48.2 | 60.3 | 61.1 | **70.0** | 72.2 | 59.1 | 60.1 |
| GLiNER-L | 0.3B | DeBERTa-V3 | 57.2 | 42.9 | 57.2 | 64.4 | 69.6 | **72.6** | 62.6 | 60.9 |
| $\text{Gold}_{\text{FT}}$ | 8B | Llama 3.1 | 63.6 ±0.8 | 54.0 ±1.3 | **62.6** ±1.7 | **65.4** ±2.0 | 67.4 ±2.8 | 66.3 ±1.9 | 60.1 ±0.2 | 62.8 ±1.2 |
| $\text{GUIDEX}_{\text{FT}}$ + $\text{Gold}_{\text{FT}}$ | 8B | Llama 3.1 | **65.3** ±0.9 | **55.5** ±1.3 | 62.4 ±1.2 | 63.8 ±2.8 | 67.9 ±0.5 | 69.6 ±1.3 | 64.6 ±1.1 | **64.2** ±0.7 |

Table 3: Zero-Shot F1-scores on Out-of-Domain NER Benchmarks, reporting state-of-the-art systems and two systems fine-tuned with and without GUIDEX. Results are averaged across 3 runs.

surpassing conventional instruction tuning and guideline-based baselines. By utilizing LLMs and domain-specific documents to generate synthetic data, GUIDEX offers a reliable method for enhancing zero-shot IE.

## 7 Analysis

This section examines the impact of GUIDEX across different labels, highlighting its benefits and limitations. Table 4 provides a breakdown of performance gains in various domains and identifies cases where GUIDEX still struggles.

**Do Guidelines Improve Domain-Specific Labels?** GUIDEX effectively mitigates the overgeneralization tendency in zero-shot IE (Sainz et al., 2024) by teaching models to differentiate between broad and fine-grained entity labels through structured annotation schemas. Table 4 shows how baseline models frequently default to generic labels like *Person* instead of recognizing domain-specific entities such as *Scientist* and *Politician*, leading to misclassifications. Fine-tuning with GUIDEX significantly improves precision, with F1 gains of up to 12.8 points for these cases, as well as a 6.75-point increase in distinguishing *PoliticalParty* from the broader *Organization* category. The model, trained on explicit contextual definitions, applies labels more accurately, reducing errors where, for instance, political parties were misclassified as organizations due to linguistic similarities. These results highlight the impact of structured guideline-driven learning in improving model adaptability, reinforcing context-aware predictions, and enabling more precise entity differentiation in specialized domains. For inherently generic labels, such as *Location* and *Country*, the model already achieves strong performance without GUIDEX. This suggests that our approach is most beneficial for refining entity granularity rather than improving well-established, domain-agnostic categories.

**Remaining challenges.** Some labels, such as *Other* and *Miscellaneous*, remain problematic even with GUIDEX (see Table 4). These categories often lack clear definitions, making it difficult to apply them consistently. Since GUIDEX generates precise guidelines, the absence of well-defined annotation criteria for these broad labels limits its effectiveness. This aligns with findings (Sainz et al., 2024) suggesting that guideline-driven models struggle with vague or catch-all entity types.

## 8 Conclusions

In this paper, we introduce GUIDEX, a novel approach for synthetic data generation aimed at IE. We utilize GUIDEX to generate data suitable for a variety of domains using documents from FineWeb-edu. As a demonstration of the method, we generate a dataset with 10,000 annotated documents, featuring a wide range of labels, from generic to highly domain-specific. Using this generated data, we train an IE model that surpasses the performance of current state-of-the-art zero-shot NER systems. Furthermore, we demonstrate that our method effectively generates domain-specific annotations, which can be utilized to train robust IE systems across multiple domains.

GUIDEX paves the way for two key research directions: (1) advancing document-level IE methodologies, and (2) developing automated techniques for handling ill-defined or generic labels like *Other* and *Miscellaneous*. These challenges are critical for IE applications but remain largely unexplored, offering potential for future work. Our method provides a foundation to address these open problems

| Dataset | Label | Summarized Guideline | Gold$_{FT}$ | GUIDEX + Gold$_{FT}$ |
|---|---|---|---|---|
| Natural Science | Scientist | A person who is studying or has expert knowledge of a natural science field. | 38.43 | 51.21 |
| | Person | Individuals that are not scientist. | 48.53 | 53.02 |
| Politics | Politician | A person who is actively engaged in politics, holding a public office, involved in political activities or part of a political party. | 35.12 | 44.37 |
| | Person | Individuals that are not politician. | 59.48 | 62.86 |
| Politics | PoliticalParty | An organization that coordinates candidates to compete in a particular country's elections. | 58.55 | 65.30 |
| | Organization | A structured group, institution, company, or association that is not a political party. | 54.34 | 58.38 |
| AI | Location | A specific geographical or structural location. | 75.36 | 75.86 |
| Music | Country | A sovereign nation. | 81.62 | 82.45 |
| Music | Other | Named entities that are not included in any other category. | 15.99 | 13.02 |
| Literature | Other | Named entities that are not included in any other category. | 23.19 | 20.78 |

Table 4: F1 scores for specific labels from different datasets with summarized guideline descriptions. On green, the labels where the domain knowledge acquired by training in GUIDEX is helpful for the model. On blue, the labels where there is no improvement. And, on red, those labels that both systems struggle to identify correctly.

systematically.

Finally, it is worth noting that all four prompts, the driving script, and the consistency filter act as drop-in tools that could be pointed at any plain-text corpus. GuideX will return a ready-to-train dataset within hours.

## Limitations

In this work, we present an approach for generating synthetic data, which we used to enhance the performance of IE models on NER datasets across various domains. However, our evaluation does not encompass all potential applications of this approach. For example, the automatically generated dataset consists of document-level texts rather than individual sentences. Our evaluation framework, on the other hand, focuses solely on sentence-level tasks. In the future, we aim to investigate the impact of our approach on document-level tasks.

A second limitation is the use of catch-all tags such as *Other* and *Miscellaneous* highlighted in Table 4. These categories function as residual classes, aggregating semantically diverse spans that lack a consistent boundary. As a result, they introduce ambiguity and noise during training. A possible direction for future work is to encode these spans into a shared embedding space, apply unsupervised clustering, and use an LLM to induce candidate schema definitions for coherent subsets. This would enable a second GuideX iteration, replacing vague categories with more fine-grained, guideline-compatible types.

## Acknowledgements

This work has been partially supported by the Basque Government (Research group funding IT1570-22 and IKER-GAITU project). We are also thankful to several projects funded by MCIN/AEI/10.13039/501100011033: (i) DeepKnowledge (PID2021-127777OB-C21) and by FEDER, EU; and AWARE (TED2021-131617B-I00) and by the European Union NextGenerationEU/PRTR. Neil De La Fuente was supported by the Basque Government through the IKASIKER 2025 scholarship programme.

## References

Dhananjay Ashok and Zachary C. Lipton. 2023. Promptner: Prompting for named entity recognition. *Preprint*, arXiv:2305.15444.

Fan Bai, Junmo Kang, Gabriel Stanovsky, Dayne Freitag, Mark Dredze, and Alan Ritter. 2024. Schema-driven information extraction from heterogeneous tables. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 10252–10273, Miami, Florida, USA. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss,

Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Yubo Chen, Shulin Liu, Xiang Zhang, Kang Liu, and Jun Zhao. 2017. Automatically labeled data generation for large scale event extraction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 409–419, Vancouver, Canada. Association for Computational Linguistics.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Leon Derczynski, Eric Nichols, Marieke van Erp, and Nut Limsopatham. 2017. Results of the WNUT2017 shared task on novel and emerging entity recognition. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pages 140–147, Copenhagen, Denmark. Association for Computational Linguistics.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: efficient finetuning of quantized llms. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA. Curran Associates Inc.

Rezarta Islamaj Doğan, Robert Leaman, and Zhiyong Lu. 2014. Ncbi disease corpus: a resource for disease name recognition and concept normalization. *Journal of biomedical informatics*, 47:1–10.

Kawin Ethayarajh, Yejin Choi, and Swabha Swayamdipta. 2022. Understanding dataset difficulty with $\mathcal{V}$-usable information. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 5988–6008. PMLR.

Hermenegildo Fabregat, Juan Martinez-Romo, and Lourdes Araujo. 2018. Overview of the diann task: Disability annotation task. In *IberEval@ SEPLN*, pages 1–14.

Steven Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard Hovy. 2021. A survey of data augmentation approaches for nlp. *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*.

Chufan Gao, Xulin Fan, Jimeng Sun, and Xuan Wang. 2024. PromptRE: Weakly-supervised document-level relation extraction via prompting-based data programming. In *Proceedings of the 1st Workshop on Towards Knowledgeable Language Models (KnowLLM 2024)*, pages 132–145, Bangkok, Thailand. Association for Computational Linguistics.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Ralph Grishman. 1997. Information extraction: Techniques and challenges. In *International Summer School on Information Extraction*.

Honghao Gui, Shuofei Qiao, Jintian Zhang, Hongbin Ye, Mengshu Sun, Lei Liang, Jeff Z. Pan, Huajun Chen, and Ningyu Zhang. 2025. Instructie: A bilingual instruction-based information extraction dataset. In *The Semantic Web – ISWC 2024*, pages 59–79, Cham. Springer Nature Switzerland.

Xianpei Han and Le Sun. 2016. Global distant supervision for relation extraction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 30.

Geoffrey Hinton. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Raphael Hoffmann, Congle Zhang, Xiao Ling, Luke Zettlemoyer, and Daniel S. Weld. 2011. Knowledge-based weak supervision for information extraction of overlapping relations. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 541–550, Portland, Oregon, USA. 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.

Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. 2017. Zero-shot relation extraction via reading comprehension. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 333–342, Vancouver, Canada. Association for Computational Linguistics.

Dongfang Li, Baotian Hu, and Qingcai Chen. 2022. Prompt-based text entailment for low-resource named entity recognition. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 1896–1903, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Jiao Li, Yueping Sun, Robin J Johnson, Daniela Sciaky, Chih-Hsuan Wei, Robert Leaman, Allan Peter Davis, Carolyn J Mattingly, Thomas C Wiegers, and Zhiyong Lu. 2016. Biocreative v cdr task corpus: a resource for chemical disease relation extraction. *Database*, 2016.

Peng Li, Tianxiang Sun, Qiong Tang, Hang Yan, Yuanbin Wu, Xuanjing Huang, and Xipeng Qiu. 2023. Codeie: Large code generation models are better few-shot information extractors. *arXiv preprint arXiv:2305.05711*.

Zixuan Li, Yutao Zeng, Yuxin Zuo, Weicheng Ren, Wenxuan Liu, Miao Su, Yucan Guo, Yantao Liu, Xiang Li, Zhilei Hu, and 1 others. 2024. Knowcoder: Coding structured knowledge into llms for universal information extraction. *arXiv preprint arXiv:2403.07969*.

Xincheng Liao, Junwen Duan, Yixi Huang, and Jianxin Wang. 2025. RUIE: Retrieval-based unified information extraction using large language model. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 9640–9655, Abu Dhabi, UAE. Association for Computational Linguistics.

Yankai Lin, Shiqi Shen, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. 2016. Neural relation extraction with selective attention over instances. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2124–2133, Berlin, Germany. Association for Computational Linguistics.

Jingjing Liu, Panupong Pasupat, Scott Cyphers, and James R. Glass. 2013. Asgard: A portable architecture for multilingual dialogue systems. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2013, Vancouver, BC, Canada, May 26-31, 2013*, pages 8386–8390. IEEE.

Zihan Liu, Yan Xu, Tiezheng Yu, Wenliang Dai, Ziwei Ji, Samuel Cahyawijaya, Andrea Madotto, and Pascale Fung. 2020. Crossner: Evaluating cross-domain named entity recognition.

Jie Lou, Yaojie Lu, Dai Dai, Wei Jia, Hongyu Lin, Xianpei Han, Le Sun, and Hua Wu. 2023a. Universal information extraction as unified semantic matching. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 13318–13326.

Jie Lou, Yaojie Lu, Dai Dai, Wei Jia, Hongyu Lin, Xianpei Han, Le Sun, and Hua Wu. 2023b. Universal information extraction as unified semantic matching. In *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'23/IAAI'23/EAAI'23. AAAI Press.

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 1003–1011.

Ying Mo, Jiahao Liu, Jian Yang, Qifan Wang, Shun Zhang, Jingang Wang, and Zhoujun Li. 2024. C-ICL: Contrastive in-context learning for information extraction. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 10099–10114, Miami, Florida, USA. Association for Computational Linguistics.

Abiola Obamuyide and Andreas Vlachos. 2018. Zero-shot relation classification as textual entailment. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pages 72–78, Brussels, Belgium. Association for Computational Linguistics.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. *Preprint*, arXiv:2203.02155.

Chaoxu Pang, Yixuan Cao, Qiang Ding, and Ping Luo. 2023. Guideline learning for in-context information extraction. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 15372–15389, Singapore. Association for Computational Linguistics.

Guilherme Penedo, Hynek Kydlíček, Loubna Ben allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, and Thomas Wolf. 2024. The fineweb datasets: Decanting the web for the finest text data at scale. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Hwee Tou Ng, Anders Björkelund, Olga Uryupina, Yuchen Zhang, and Zhi Zhong. 2013. Towards robust linguistic analysis using ontonotes. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 143–152.

Yunjia Qi, Hao Peng, Xiaozhi Wang, Bin Xu, Lei Hou, and Juanzi Li. 2024. Adelie: Aligning large language models on information extraction. *arXiv preprint arXiv:2405.05008*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(1).

Oscar Sainz, Oier Lopez de Lacalle, Gorka Labaka, Ander Barrena, and Eneko Agirre. 2021. Label verbalization and entailment for effective zero- and few-shot relation extraction. *arXiv preprint arXiv:2109.03659*.

Oscar Sainz, Iker García-Ferrero, Rodrigo Agerri, Oier Lopez de Lacalle, German Rigau, and Eneko Agirre. 2024. GoLLIE: Annotation guidelines improve zero-shot information-extraction. In *The*

*Twelfth International Conference on Learning Representations*.

Mihai Surdeanu, Julie Tibshirani, Ramesh Nallapati, and Christopher D Manning. 2012. Multi-instance multi-label learning for relation extraction. In *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning*, pages 455–465.

Teknium. 2023. Openhermes 2.5: An open dataset of synthetic data for generalist llm assistants.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Christopher Walker, Stephanie Strassel, Julie Medero, and Kazuaki Maeda. 2006. ACE 2005 multilingual training corpus LDC2006T06.

Chenguang Wang, Xiao Liu, Zui Chen, Haoyun Hong, Jie Tang, and Dawn Song. 2021. Zero-shot information extraction as a unified text-to-triple translation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1225–1238, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Xiao Wang, Weikang Zhou, Can Zu, Han Xia, Tianze Chen, Yuansen Zhang, Rui Zheng, Junjie Ye, Qi Zhang, Tao Gui, Jihua Kang, Jingsheng Yang, Siyuan Li, and Chunsai Du. 2023. Instructuie: Multi-task instruction tuning for unified information extraction. *Preprint*, arXiv:2304.08085.

Xiang Wei, Xingyu Cui, Ning Cheng, Xiaobin Wang, Xin Zhang, Shen Huang, Pengjun Xie, Jinan Xu, Yufeng Chen, Meishan Zhang, Yong Jiang, and Wenjuan Han. 2024. Chatie: Zero-shot information extraction via chatting with chatgpt. *Preprint*, arXiv:2302.10205.

Xiang Wei, Xingyu Cui, Ning Cheng, Xiaobin Wang, Xin Zhang, Shen Huang, Pengjun Xie, Jinan Xu, Yufeng Chen, Meishan Zhang, and 1 others. 2023. Zero-shot information extraction via chatting with chatgpt. *arXiv preprint arXiv:2302.10205*.

Chaojun Xiao, Yuan Yao, Ruobing Xie, Xu Han, Zhiyuan Liu, Maosong Sun, Fen Lin, and Leyu Lin. 2020. Denoising relation extraction from document-level distant supervision. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3683–3688, Online. Association for Computational Linguistics.

Derong Xu, Wei Chen, Wenjun Peng, Chao Zhang, Tong Xu, Xiangyu Zhao, Xian Wu, Yefeng Zheng, Yang Wang, and Enhong Chen. 2024a. Large language models for generative information extraction: A survey. *Frontiers of Computer Science*, 18(6):186357.

Jun Xu, Mengshu Sun, Zhiqiang Zhang, and Jun Zhou. 2024b. ChatUIE: Exploring chat-based unified information extraction using large language models. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 3146–3152, Torino, Italia. ELRA and ICCL.

Zhangchen Xu, Fengqing Jiang, Luyao Niu, Yuntian Deng, Radha Poovendran, Yejin Choi, and Bill Yuchen Lin. 2025. Magpie: Alignment data synthesis from scratch by prompting aligned LLMs with nothing. In *The Thirteenth International Conference on Learning Representations*.

Urchade Zaratiana, Nadi Tomeh, Pierre Holat, and Thierry Charnois. 2023. Gliner: Generalist model for named entity recognition using bidirectional transformer. *Preprint*, arXiv:2311.08526.

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D Manning. 2017. Position-aware attention and supervised data improve slot filling. In *Conference on empirical methods in natural language processing*.

Wenxuan Zhou, Sheng Zhang, Yu Gu, Muhao Chen, and Hoifung Poon. 2023. Universalner: Targeted distillation from large language models for open named entity recognition.

# A Implementation Details

In this section, we further detail the whole process of GUIDEX. In Section A.1 we explain how we built the dataset, focusing on the hyperparameters used, the multi-step generation prompts, and the filtering process that ensures consistent annotations. In A.2 we describe the hyperparameters for both GUIDEX$_{\text{FT}}$ and Gold$_{\text{FT}}$, outlining how they were tuned to achieve the outcomes showcased on Section 6.

## A.1 GUIDEX Dataset generation

The GUIDEX dataset was built through a structured multi-step process designed to ensure high-quality, consistent annotations and guidelines. The dataset generation pipeline follows a systematic approach involving prompt-based multi-step generation, filtering for consistency, and the use of hyperparameter tuning to optimize the outputs.

**Model and Hyperparameters.** The synthetic dataset was generated using Llama 3.1-70B Instruct, leveraging vLLM for efficient inference. The detailed hyperparameter settings used in the generation process are available in Table 5.

**Multi-Step Generation Process.** To ensure structured and meaningful outputs, the dataset was built using a four-step generation pipeline. The first step involved extracting key points from the input text, reducing redundancy while preserving essential information. The second step transformed this summarized content into a structured JSON representation, ensuring a consistent and standardized format. Next, annotation guidelines were generated to define the expected attributes and structure, facilitating consistency across all annotations. Finally, the model extracted instances based on these guidelines, ensuring that the final dataset adhered to a coherent format. The complete set of prompts used for each step can be found in Figure 3.

## A.2 GUIDEX$_{\text{FT}}$ & Gold$_{\text{FT}}$

This section summarizes the fine-tuning configurations for GUIDEX$_{\text{FT}}$ and Gold$_{\text{FT}}$. Both models use Llama 3.1-8B with 4-bit LoRA, AdamW optimization, and a cosine scheduler. GUIDEX$_{\text{FT}}$ supports longer sequences (8192 tokens) as it is suited for document-level input texts, and employs gradient accumulation, while Gold$_{\text{FT}}$ uses a larger per-device batch size without accumulation. Training is conducted on 2× A100 (80GB) GPUs with DeepSpeed Zero-3, differing in sequence length, batch sizes, and the number of training epochs. A full overview is provided in Table 6.

| Category | Hyperparameter | Value |
|---|---|---|
| **Model Setup** | Model Name | Llama 3.1-70B Instruct |
| | Tokenizer | Llama 3.1-70B Instruct |
| | Dtype | `bfloat16` |
| | Max Seq. Length | 8192 tokens |
| | Tensor Parallel Size | 2 |
| **Generation Config** | Temperature | 0.7 |
| | Top-p | 0.95 |
| | Max New Tokens | 1024 |
| **Batching** | Batch Size | 32 |
| | Processing Mode | Batched prompts via `vLLM` |
| **Hardware** | GPUs Used | 4× A100 (80GB) |
| | CPU per Task | 16 |

Table 5: Hyperparameter Settings for the GUIDEX Synthetic Data Generation Stage.

### I Document Summarization prompt

```
Use bullet points to summarize the main ideas of the following text, keeping the most important information. Text: '{text}'. Summarize these
points concisely.
```

### II Structured Representation prompt

```
Based on the extracted summary and the original text, synthesize the information into a JSON output.Keep it as less verbose as possible. The
strings in the JSON must match the original text.The name of each key should be properly chosen and general, similar fields should be merged.

You should populate all the attributes and be as concise as possible. The attributes can't be populated with full sentences; they must be as
short as possible. Remember that it should match the contents of the text accurately.

The only thing that you should return is a single JSON that contains the required content, nothing else, no text introducing the JSON, no
text saying how you hope it is ok. JUST THE JSON.
```

### III Guideline Generation prompt

````
Based on the JSON output, the summary and the original text, generate annotation guidelines that include:
1. A high-quality, complete, and extensive description that is long enough.
2. The expected format for each field.

Then, turn these annotations into a Python file consisting of Python dataclass objects.Description of the class should be given as a docstring
and descriptions of the attributes must be given as comments, without examples. Create a class that wraps all the information together. Return
only one Python file that contains these annotation guidelines. The kind of output I am expecting is similar (but not limited) to this, but
substituting the placeholders with the specifics of the text being discussed, and creating as many classes as needed and naming them properly:

```python
from dataclasses import dataclass
from typing import List, Optional

@dataclass
class EntityA:
    """
    A generic but long enough description for EntityA, explaining its purpose and characteristics
    in a general context without specific details.
    """
    identifier: str
    """
    A brief explanation of what the identifier represents in a general sense.
    """
    attribute1: str # A short comment explaining this attribute generically
    attribute2: List[str]
    """
    A multi-line comment explaining what this list typically contains
    and its significance to EntityA.
    """
    # Add more attributes as needed

@dataclass
class EntityB:
    """
    A generic description for EntityB, explaining its purpose and characteristics
    in a general context without specific details.
    """
    mention: str
    """
    An explanation of what the mention represents for EntityB.
    """
    date: str # A comment about what this date signifies
    location: str # A comment about what this location represents
    element: str # A comment about what this element represents
    # Add more attributes as needed
```
````

### IV Instance Extraction prompt

````
Based on the annotation guidelines, you will provide a Python list called `result_instances` containing instances of the dataclasses based on the
information in the text. Be as concise as possible when extracting the instances, which should be single words or numbers when possible, not
longer. You should only provide the Python `result_instances` list with the instances, nothing else. Do not provide explanations nor extra data
apart from what's contained in the `result_instances` list.

The kind of output I am expecting is similar (but not limited) to this, but substituting the placeholders with the specifics of the text being
discussed:

```python
result_instances = [
    EntityB(name="EntityB_Name1", date="Date1", location1="Location1A", element="element1"),
    EntityA(identifier="EntityA_ID1", attribute1="AttributeValue1", attribute2=["Item1", "Item2", "Item3"])
]
```
````

Figure 3: GUIDEX follows a multi-step prompting pipeline which allows for the creation of the synthetic guidelines and annotations that conform GUIDEX and are used for $\text{GUIDEX}_{\text{FT}}$.

| Category | Hyperparameter | GUIDEX$_{\text{FT}}$ | Gold$_{\text{FT}}$ |
|---|---|---|---|
| **Model & Quantization** | Base Model | Llama 3.1-8B | Llama 3.1-8B |
| | Quantization | 4-bit LoRA | 4-bit LoRA |
| | LoRA Rank ($r$) | 128 | 128 |
| | LoRA $\alpha$ | 256 | 256 |
| | LoRA Dropout | 0.08 | 0.05 |
| | Dtype | `bfloat16` | `bfloat16` |
| **Optimization** | Optimizer | AdamW | AdamW |
| | Learning Rate | $3 \times 10^{-4}$ | $3 \times 10^{-4}$ |
| | Weight Decay | 0.001 | 0.001 |
| | Scheduler | Cosine | Cosine |
| | Warmup Steps | 10% of total steps | 10% of total steps |
| **Batching & Seq. Length** | Per-device Batch Size | 4 | 16 |
| | Gradient Accumulation Steps | 2 | None |
| | Effective Batch Size | 8 | 16 |
| | Max Sequence Length (tokens) | 8192 | 2048 |
| **Epochs & Checkpoints** | Epochs | 3 | 1 |
| | Checkpoint Strategy | End of each epoch | End of each epoch |
| **Hardware** | GPUs Used | 2× A100 (80GB) | 2× A100 (80GB) |
| | Multi-GPU Support | DeepSpeed Zero-3 | DeepSpeed Zero-3 |
| | CPU per Task | 22 | 22 |

Table 6: Hyperparameter Settings for GUIDEX$_{\text{FT}}$ and Gold$_{\text{FT}}$.