# GoLLIE : Annotation Guidelines improve Zero-Shot Information-Extraction

**Oscar Sainz***, **Iker García-Ferrero***
**Rodrigo Agerri**, **Oier Lopez de Lacalle**, **German Rigau**, **Eneko Agirre**
HiTZ Basque Center for Language Technology - Ixa NLP Group
University of the Basque Country (UPV/EHU)
{oscar.sainz, iker.garciaf}@ehu.eus

## Abstract

Large Language Models (LLMs) combined with instruction tuning have made significant progress when generalizing to unseen tasks. However, they have been less successful in Information Extraction (IE), lagging behind task-specific models. Typically, IE tasks are characterized by complex annotation guidelines that describe the task and give examples to humans. Previous attempts to leverage such information have failed, even with the largest models, as they are not able to follow the guidelines out of the box. In this paper, we propose GoLLIE (**G**uideline-f**o**llowing **L**arge **L**anguage Model for **IE**), a model able to improve zero-shot results on unseen IE tasks by virtue of being fine-tuned to comply with annotation guidelines. Comprehensive evaluation empirically demonstrates that GoLLIE is able to generalize to and follow unseen guidelines, outperforming previous attempts at zero-shot information extraction. The ablation study shows that detailed guidelines are key for good results. Code, data, and models are publicly available: https://github.com/hitz-zentroa/GoLLIE.

## 1 Introduction

The task of Information Extraction (IE) is highly challenging. This challenge is evident in the detailed guidelines, which feature granular definitions and numerous exceptions, that human annotators must follow to perform the task. The performance of current SoTA models heavily depends on the quantity of human-annotated data, as the model learns the guidelines from these examples. However, this performance significantly decreases when tested in new annotation schema (Liu et al., 2021a). The common practice in IE to achieve good results is to manually annotate each new domain and schema from scratch, as almost no transfer exists across application domains. Unfortunately, this is unfeasible, both, in terms of financial cost and human effort.

Recent advancements in Large Language Models (LLM) (Min et al., 2023) have enabled the development of models capable of generalizing to unseen tasks. Thus, current zero-shot IE systems

*Equal contribution

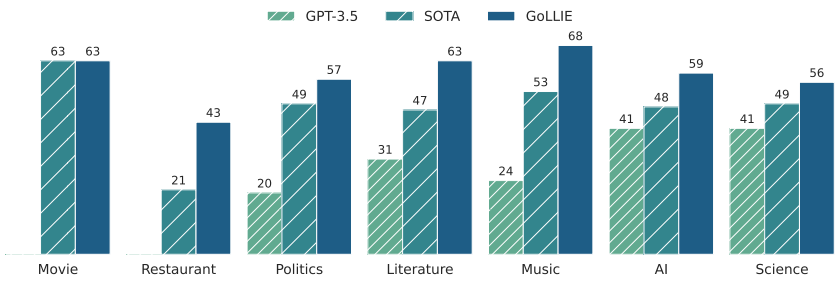



Figure 1: Out of domain zero-shot NER results. GPT results are not available for all domains.

leverage the knowledge encoded in LLMs to annotate new examples (Sainz et al., 2022a; Wang et al., 2023a). As a by-product of the pre-training process, models possess now a strong representation of what a person or an organization is. Therefore, they can be prompted to extract mentions of those categories from a text. However, this has a clear limitation: not every annotation schema* defines "person" (or any other label) in the same way. For example, ACE05 (Walker et al., 2006) annotates pronouns as persons, while, CoNLL03 (Tjong Kim Sang & De Meulder, 2003) does not. IE tasks require more information than just label names, they require annotation guidelines.

Current LLMs have been trained to follow instructions, but they fail to follow annotation guidelines out of the box. For instance, Figure 1 shows results on domain-specific zero-shot Named Entity Recognition. The results of gpt-3.5-turbo when prompted with guidelines (Zhang et al., 2023a) are low, around 20 F1 score on Music or Politics domains. Building a system that enables high-performance zero-shot information extraction, reducing the dependence on costly human annotations, remains an open challenge.

In this work, we present GoLLIE (**G**uideline-f**o**llowing **L**arge **L**anguage Model for **IE**), an LLM fine-tuned to learn how to attend to the guidelines on a small set of well known IE tasks. Comprehensive zero-shot evaluation empirically demonstrates that GoLLIE outperforms the SoTA (Wang et al., 2023a) in zero-shot information extraction (see Figure 1).

## 2 Related Work

Large Language Models (LLMs) have made significant advancements toward the development of systems that can generalize to unseen tasks (Min et al., 2023). Radford et al. (2019) trained LLMs using a vast amount of internet data, finding that pre-trained models given natural language task descriptions can perform tasks such as question answering, machine translation, or summarizing without explicit supervision. Building on this discovery, instruction tuning, often referred to as multitask fine-tuning, has emerged as the leading method to achieve generalization to unseen tasks. This process involves pre-training a model on a massive amount of unlabeled data and subsequently fine-tuning it on a diverse collection of tasks (Wang et al., 2022; Chung et al., 2022) phrased as text-to-text problems (Raffel et al., 2020). A natural language instruction or prompt is given to the model to identify the task it should solve (Schick & Schütze, 2021; Scao & Rush, 2021). Research has demonstrated that increasing the parameter count of the language model (Brown et al., 2020), coupled with improvements in the size and quality of the instruction tuning dataset, results in enhanced generalization capabilities (Chen et al., 2023; Zhang et al., 2022; Chowdhery et al., 2022; Muennighoff et al., 2023; Touvron et al., 2023a;b). LLMs have displayed impressive zero-shot generalization capabilities in various challenging tasks, including coding Wang & Komatsuzaki (2021); Black et al. (2022); Rozière et al. (2023), common sense reasoning Touvron et al. (2023a), and medical applications Singhal et al. (2023), among others.

In the field of Information Extraction (IE), recent shared tasks (Fetahu et al., 2023) have shown that encoder-only language models such as XLM-RoBERTa (Conneau et al., 2020) and mDEBERTA (He et al., 2023) remain the most effective models. Attempts to utilize LLMs and natural language instructions for IE have been less successful (Tan et al., 2023; Zhou et al., 2023; Zhang et al., 2023a), as their performance lags behind that of encoder-only models. Before the billion parameters LLMs, indirectly supervised methods improve zero-shot IE by utilizing the knowledge learned from tasks like Textual Entailment (Sainz et al., 2021; 2022a;b) and Question Answering (Levy et al., 2017). Obeidat et al. (2019) propose an entity typing method that encodes label descriptions from Wikipedia as embeddings using an LSTM, which is then used to score the inputs. Methods that leveraged external knowledge were also successful on fine-grained zero-shot NER (Chen et al., 2021). Lu et al. (2022a) introduced a unified text-to-structure generation that can model different IE tasks universally. Lou et al. (2023) proposed converting IE tasks to a semantic matching problem, allowing their method to generalize to new domains and label ontologies not seen during training. Wang et al. (2023a) framed IE tasks as natural language descriptive instructions and trained an LLM across a diverse range of IE tasks. In evaluations on tasks with unseen label ontologies, their model outperformed other instruction-tuning methods.

*We define schema as the set of labels and their definitions.

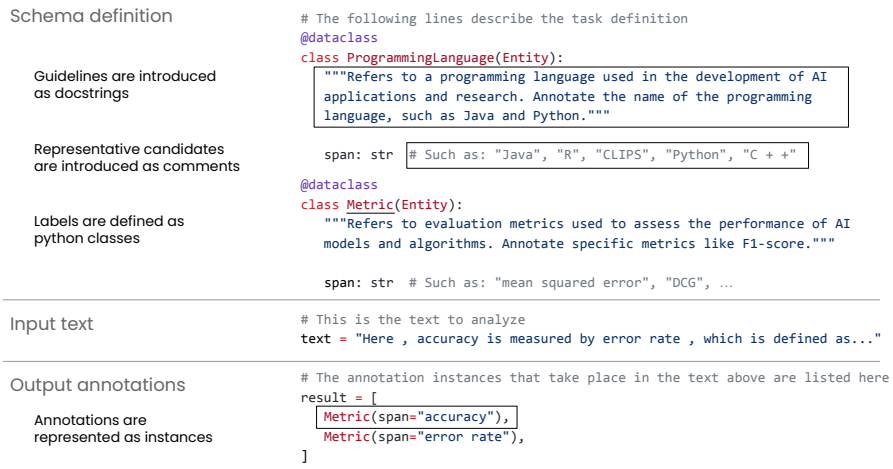



Figure 2: Example of the input and output of the model.

Most instruction tuning attempts for IE share a limitation: they only consider label names in the prompts (e.g., *"List all the Persons"*). This poses two major challenges. Firstly, not all datasets share the same definition for labels like *Person* (some exclude fictional characters or pronouns). Secondly, a label name alone doesn't sufficiently describe complex or less common labels. While there have been attempts to prompt LLMs using guidelines (Zhang et al., 2023a), strong prior knowledge of LLMs regarding task labels (Blevins et al., 2023) deter the model from adhering to those guidelines.

## 3 Approach

Different from previous approaches, GoLLIE forces the model to attend to the details in the guidelines, performing robustly on schemas not seen during training. On this section we deep dive into the details of our approach, describing how the input and output was represented and the regularization techniques used to force the model to attend to the guidelines.

### 3.1 Input-output representation

We have adopted a Python code-based representation (Wang et al., 2023b; Li et al., 2023) for both the input and output of the model. This approach not only offers a clear and human-readable structure but also addresses several challenges typically associated with natural language instructions. It enables the representation of any information extraction task under a unified format. The inputs can be automatically standardized using Python code formatters such as Black. The output is well-structured and parsing it is trivial. Furthermore, most current LLMs incorporate code in their pre-training datasets, indicating that these models are already familiar with this representation.

Figure 2 shows the three main parts of the format: schema definition, input text, and output annotations. **Schema definition** forms the initial segment of the input. This section contains information about the labels that are represented as Python classes; guidelines, articulated as docstrings; and representative annotation candidates presented in the form of code comments. The number of class definitions corresponds to the number of labels in the dataset. Classes are flexible and vary for each task. For example, classes for a NER dataset merely require an attribute to specify the text span that corresponds to the class. On the other side, more complex tasks such as Event Argument Extraction (EAE) or Slot Filling (SF) demand more class attributes to categorize the task, such as a list of participants in an event (refer to examples in Appendix A). **Input text** is the second part of the input. The input text is represented as a string variable in Python. **Output annotations** is the part generated by the model. The model starts generating after `result =`. The annotations are represented as a list of instances of the classes defined on the schema definition part. Parsing the output is straightforward; executing the generated code in Python yields a list containing the result. This ease of parsing the output stands as a significant advantage of our model. A further detailed analysis of the efficiency of this approach is available in Appendix E.

```python
@dataclass
class VulnerabilityPatch(Event):

  mention: str
  cve: List[str]
  issues_addressed: List[str]
  supported_platform: List[str]
  vulnerability: List[str]
  vulnerable_system: List[str]
  releaser: List[str]
  patch: List[str]
  patch_number: List[str]
  system_version: List[str]
  time: List[str]
```

```python
@dataclass
class VulnerabilityPatch(Event):
  """A VulnerabilityPatch Event happens when a software company addresses a
  known vulnerability by releasing or describing an appropriate update."""

  mention: str
  """The text span that triggers the event.
  Such as: patch, fixed, addresses, implemented, released
  """
  cve: List[str] # The vulnerability identifier
  issues_addressed: List[str] # What did the patch fix
  supported_platform: List[str] # The platforms that support the patch
  vulnerability: List[str] # The vulnerability
  vulnerable_system: List[str] # The affected systems
  releaser: List[str] # The entity releasing the patch
  patch: List[str] # What was the patch about
  patch_number: List[str] # Number or name of the patch
  system_version: List[str] # The version of the vulnerable system
  time: List[str] # When was the patch implemented, the date
```

Figure 3: Example of the input representation. (left) An example of an event definition w/o guidelines information. (right) The same example but with guideline information as Python comments.

### 3.2 Guidelines enhanced representation

The main contribution of this work is the use of the guidelines as part of the inference process to improve the zero-shot generalization. An example of a class definition with and without guidelines is shown in Figure 3. Different datasets usually define guidelines in many different ways: some provide a complex definition of a label with several exceptions and special treatments and others just give a few representative candidates of the fillers of the label. To normalize the input format, we included the label definitions as class docstrings and the candidates as a comment for the principal argument (which is usually *mention* or *span*). Complex tasks such as EAE or SF require additional definitions for the arguments or slots, to that end, we included small definitions as comments on each class argument. In this paper, we will refer to the model without guidelines as Baseline and the model with guidelines as GoLLIE.

### 3.3 Training regularization

We want to ensure that the model follows the guidelines and does not just learn to identify specific datasets and perform correctly on them. To do this, we introduce various kinds of noise during training. This stops the model from recognizing particular datasets, recalling specific labels, or attending only to the label names rather than learning to follow the actual description for each label in the guidelines.

We applied the following regularizations. **Class order shuffling**, for each example, the order of the input classes is randomly shuffled. This makes it more difficult for the model to memorize entire task definitions. **Class dropout**, we delete some of the input classes randomly. By eliminating few classes from both the input and output, we force the model to learn to only output instances of classes defined in the input. This not only encourages the model to focus on the schema definition but also minimizes the occurrence of hallucinations during inference. **Guideline paraphrasing**, we generate variations of the label definitions to prevent the model from easily memorizing them. We also think this will make the method more robust to different variations on the definition. **Representative candidate sampling**, similar to what we do with the paraphrases, for each input we sample 5 different candidates from a fixed pool of 10 per class. **Class name masking** involves substituting the label class names (e.g., PERSON) with placeholders, such as LABEL_1. This prevents the model from exploiting the label names during training and forces it to attend and understand the guidelines.

## 4 Experimental Setup

### 4.1 Data

Evaluating zero-shot capabilities requires dividing the data into training and evaluation datasets. However, many benchmarks for Information Extraction are based on the same domain or share part of their schema. To ensure that the zero-shot evaluation is not affected by similar data, we have divided our set of benchmarks based on the domain of the data (a related topic is data contamination,

Table 1: Datasets used on the experiments. The table shows the domain, tasks and whether are use for training, evaluation or both.

| Dataset | Domain | NER | RE | EE | EAE | SF | Training | Evaluation |
|---|---|---|---|---|---|---|---|---|
| ACE05 (Walker et al., 2006) | News | ✓ | ✓ | ✓ | ✓ | | ✓ | ✓ |
| BC5CDR (Wei et al., 2016) | Biomedical | ✓ | | | | | ✓ | ✓ |
| CoNLL 2003 (Tjong Kim Sang & De Meulder, 2003) | News | ✓ | | | | | ✓ | ✓ |
| DIANN (Fabregat et al., 2018) | Biomedical | ✓ | | | | | ✓ | ✓ |
| NCBIDisease (Islamaj Doğan & Lu, 2012) | Biomedical | ✓ | | | | | ✓ | ✓ |
| Ontonotes 5 (Pradhan et al., 2013) | News | ✓ | | | | | ✓ | ✓ |
| RAMS (Ebner et al., 2020) | News | | | | ✓ | | ✓ | ✓ |
| TACRED (Zhang et al., 2017) | News | | | | | ✓ | ✓ | ✓ |
| WNUT 2017 (Derczynski et al., 2017) | News | ✓ | | | | | ✓ | ✓ |
| BroadTwitter (Derczynski et al., 2016) | Twitter | ✓ | | | | | | ✓ |
| CASIE (Satyapanich et al., 2020) | Cybercrime | | | ✓ | ✓ | | | ✓ |
| CrossNER (Liu et al., 2021b) | *Many* | ✓ | | | | | | ✓ |
| E3C (Magnini et al., 2021) | Biomedical | ✓ | | | | | | ✓ |
| FabNER (Kumar & Starly, 2022) | Science | ✓ | | | | | | ✓ |
| HarveyNER (Chen et al., 2022) | Twitter | ✓ | | | | | | ✓ |
| MIT Movie (Liu et al., 2013) | Queries | ✓ | | | | | | ✓ |
| MIT Restaurants (Liu et al., 2013) | Queries | ✓ | | | | | | ✓ |
| MultiNERD (Tedeschi & Navigli, 2022) | Wikipedia | ✓ | | | | | | ✓ |
| WikiEvents(Li et al., 2021) | Wikipedia | ✓ | | ✓ | ✓ | | | ✓ |

which we discuss in Appendix G). For training we kept mostly datasets from **News and Biomedical** domains, for evaluation instead, we used datasets from **diverse domains**. This approach helps to avoid introducing any noise into the evaluation process. Among the evaluation datasets we included CrossNER (Liu et al., 2021b), a dataset that is split into many domains, for simplicity, we will call each domain as a separate dataset: AI, Literature, Music, Politics, and Science. Also, we will refer to MIT Movie and MIT Restaurant as Movie and Restaurant. Table 1 contains the information about the data used in the experiments.

We have trained the model to perform 5 different tasks: Named Entity Recognition (NER), Relation Extraction (RE), Event Extraction (EE), Event Argument Extraction (EAE), and Slot Filling (SF). However, we only evaluated the model on the three main tasks of interest: NER, EE, and EAE. The other two tasks are added to the training data to add diversity and improve the flexibility of the model.

A few modifications have been made to two datasets to improve the quality of the model. First, the training data of Ontonotes 5 was reduced drastically as it was automatically annotated. Second, the TACRED dataset was converted from RE to SF to increase the complexity of the task. These modifications make our system not comparable with the state of the art on those tasks. However, our focus of interest is on the zero-shot evaluation and, therefore, the benefits (see Appendix A) are more interesting than adding 2 more comparable points on the supervised setup. In the CASIE dataset, we detected that the annotated event spans are inconsistent. The models typically annotate a sub-string rather than the entire span. Therefore, we evaluate all the models based on the predicted event categories, without considering the exact text span. For arguments, we use partial matching.

We use the guidelines released by the authors of each dataset (More details in Appendix F). When such guidelines are not publicly available, we ask human experts to create them, based on the annotations from the development split. The representative candidates are extracted from the guidelines when available, otherwise, the candidates are sampled from the the train split based on word frequency or manually curated based on the guidelines. Paraphrases are automatically generated using Vicuna 33B v1.3 (Zheng et al., 2023).

### 4.2 Language Models and Baselines

**Backbone LLMs:** GoLLIE is a fine-tuned version of Code-LLaMA Rozière et al. (2023). Other backbone LLMs, such as LLaMA (Touvron et al., 2023a), LLaMA-2 Touvron et al. (2023b) or Falcon Penedo et al. (2023) were considered during the development, however, as our approach uses code to represent the input and output, Code-LLaMA model worked better on the preliminary experiments. In order to perform fair comparisons the baseline developed in this paper is based on Code-LLaMA as well. All the development of this paper was done with the 7B parameter version of Code-LLama, but, for a scaling analysis, we also trained the 13B and 34B parameter models.

Table 2: Supervised evaluation results. "*" indicates that results are not directly comparable.

| Dataset | SoTA | Baseline | 7B | 13B | 34B |
|---|---|---|---|---|---|
| $ACE05_{NER}$ | (Wang et al., 2023a) 86.6 | $89.1_{\pm 0.2}$ | $88.1_{\pm 0.6}$ | $89.4_{\pm 0.2}$ | **89.6**$_{\pm 0.1}$ |
| $ACE05_{RE}$ | (Lu et al., 2022b) 66.1 | $63.8_{\pm 0.6}$ | $63.6_{\pm 1.8}$ | $67.5_{\pm 0.5}$ | **70.1**$_{\pm 1.5}$ |
| $ACE05_{EE}$ | (Lu et al., 2022b) **73.4** | $71.7_{\pm 0.2}$ | $72.2_{\pm 0.8}$ | $70.9_{\pm 1.6}$ | $71.9_{\pm 1.1}$ |
| $ACE05_{EAE}$ | (Lu et al., 2022b) *54.8 | $65.9_{\pm 0.7}$ | $66.0_{\pm 0.8}$ | $67.8_{\pm 0.9}$ | **68.6**$_{\pm 1.2}$ |
| BC5CDR | (Zhang et al., 2023b) **91.9** | $87.5_{\pm 0.2}$ | $87.5_{\pm 0.2}$ | $87.9_{\pm 0.1}$ | $88.4_{\pm 0.2}$ |
| CoNLL 2003 | (Lu et al., 2022b) 93.0 | $92.9_{\pm 0.1}$ | $92.8_{\pm 0.3}$ | $93.0_{\pm 0.2}$ | **93.1**$_{\pm 0.1}$ |
| DIANN | (Zabala et al., 2018) 74.8 | $80.3_{\pm 0.7}$ | $79.4_{\pm 1.1}$ | $82.6_{\pm 1.3}$ | **84.1**$_{\pm 1.1}$ |
| NCBIDisease | (Wang et al., 2023a) **90.2** | $86.2_{\pm 0.1}$ | $85.4_{\pm 0.3}$ | $86.5_{\pm 0.8}$ | $85.8_{\pm 0.2}$ |
| Ontonotes 5 | - | $83.4_{\pm 0.2}$ | $83.4_{\pm 0.2}$ | $84.0_{\pm 0.2}$ | **84.6**$_{\pm 0.4}$ |
| RAMS | (Li et al., 2021) 48.6 | $48.9_{\pm 0.4}$ | $48.7_{\pm 0.7}$ | $49.6_{\pm 0.1}$ | **51.2**$_{\pm 0.3}$ |
| TACRED | - | $56.6_{\pm 0.2}$ | $57.1_{\pm 0.9}$ | $56.7_{\pm 0.5}$ | **58.7**$_{\pm 0.2}$ |
| WNUT 2017 | (Wang et al., 2021) **60.2** | $53.7_{\pm 0.7}$ | $52.0_{\pm 0.6}$ | $50.5_{\pm 0.9}$ | $54.3_{\pm 0.4}$ |
| Average | | $73.3_{\pm 0.1}$ | $73.0_{\pm 0.3}$ | $73.9_{\pm 0.3}$ | **75.0**$_{\pm 0.3}$ |

**Training setup:** To train the models we use QLoRA (Hu et al., 2022; Dettmers et al., 2023). LoRA freezes the pre-trained model weights and injects trainable rank decomposition matrices into linear layers of the Transformer architecture. In a preliminary experiment, this setup outperformed fine-tuning the entire model on the zero-shot tasks, while training much faster (more details in Appendix D.4). We applied the LoRA to all linear transformer block layers as recommended by Dettmers et al. (2023). The models were trained for 3 epochs with an effective batch size of 32 and a learning rate of 3e-4 with a cosine scheduler. Our training infrastructure was 2 NVIDIA's A100 with 80gb each. More details about the training are given in the Appendix D.

**Comparable systems:** Our main point of comparison is Instruct-UIE (Wang et al., 2023a) as it is the approach closest to our system, but does not use guidelines. Another system considered for comparison is PromptNER (Zhang et al., 2023a), which proposes to prompt GPT-3.5 and T5 with definitions using Chain-of-Though in order to perform few-shot NER. Different from us, they did not fine-tune the model to attend to the guidelines. For fair comparison, we only considered the zero-shot results reported in the paper. In addition, other SoTA systems are added for comparison when results from Instruct-UIE and PromptNER are not available. Given that our system is designed for the zero-shot scenario, the supervised experiments are intended to verify that our system does not degrade its performance. Thus we selected, for the supervised scenario, those systems among SoTA that share the most comparable setting with us.

# 5 Results

## 5.1 Supervised evaluation

The results on the supervised datasets are shown in Table 2. Comparing GoLLIE with the baseline, they both obtain very similar results, with an absolute difference of 0.3 F1 points on average. This is expected, as the baseline model implicitly learns the guidelines for annotating the datasets based on the data distribution during fine-tuning. In addition, despite the noise introduced to GoLLIE fine-tuning in order to generalize from guidelines, the performance is close to that of the baseline.

Compared to other systems our model achieves similar results in general. Focusing on the two datasets where our model under-performs significantly, WNUT and NCBIDisease, we find that task-specific techniques are still needed. For instance, Wang et al. (2021) uses external knowledge to detect emergent and rare entities. In the NCBIDisisease dataset, models pre-trained on Biomedical domain corpora achieve the best results (Kocaman & Talby, 2021). (Wang et al., 2023a) leverages Flan-T5, which has great proficiency on Biomedical domain tasks (Singhal et al., 2022). These improvements, however, are complementary to our proposal.

## 5.2 Zero-Shot evaluation

The results on the zero-shot are shown in Table 3. Overall, compared to the baseline, **the results are improved significantly when using guidelines** on almost every dataset, with an absolute difference

Table 3: Zero-shot evaluation results. "*" indicates results obtained using the original code.

| Dataset | SoTA | Baseline | GoLLIE | GoLLIE 13B | GoLLIE 34B |
|---|---|---|---|---|---|
| BroadTwitter | - | 39.0$_{\pm 0.6}$ | 49.5$_{\pm 0.8}$ | **51.4**$_{\pm 1.8}$ | 50.3$_{\pm 2.1}$ |
| CASIE$_{\text{EE}}$ | - | 33.9$_{\pm 6.5}$ | 59.3$_{\pm 2.3}$ | 62.2$_{\pm 0.9}$ | **65.5**$_{\pm 1.8}$ |
| CASIE$_{\text{EAE}}$ | - | 47.9$_{\pm 1.4}$ | 50.0$_{\pm 1.1}$ | 52.6$_{\pm 0.2}$ | **55.2**$_{\pm 0.5}$ |
| AI | (Wang et al., 2023a) 49.0 | 32.3$_{\pm 0.8}$ | 59.1$_{\pm 1.1}$ | 56.7$_{\pm 3.0}$ | **61.6**$_{\pm 1.9}$ |
| Literature | (Wang et al., 2023a) 47.2 | 39.4$_{\pm 0.7}$ | **62.7**$_{\pm 3.2}$ | 59.7$_{\pm 0.3}$ | 59.1$_{\pm 2.6}$ |
| Music | (Wang et al., 2023a) 53.2 | 56.2$_{\pm 1.3}$ | 67.8$_{\pm 0.2}$ | 65.5$_{\pm 3.6}$ | **68.4**$_{\pm 2.1}$ |
| Politics | (Wang et al., 2023a) 48.2 | 38.3$_{\pm 1.1}$ | 57.2$_{\pm 1.0}$ | 54.4$_{\pm 4.1}$ | **60.2**$_{\pm 3.0}$ |
| Science | (Wang et al., 2023a) 49.3 | 37.1$_{\pm 1.3}$ | 55.5$_{\pm 1.6}$ | 56.2$_{\pm 1.0}$ | **56.3**$_{\pm 0.4}$ |
| E3C | - | 59.8$_{\pm 0.3}$ | 59.0$_{\pm 0.7}$ | 59.0$_{\pm 0.8}$ | **60.0**$_{\pm 0.4}$ |
| FabNER | - | 06.1$_{\pm 0.4}$ | 24.8$_{\pm 0.6}$ | 25.4$_{\pm 0.5}$ | **26.3**$_{\pm 0.4}$ |
| HarveyNER | - | 23.2$_{\pm 0.4}$ | 37.3$_{\pm 1.8}$ | **41.3**$_{\pm 0.8}$ | 38.9$_{\pm 0.5}$ |
| Movie | (Wang et al., 2023a) 63.0 | 43.4$_{\pm 1.1}$ | **63.0**$_{\pm 0.6}$ | 62.5$_{\pm 1.0}$ | 62.4$_{\pm 1.4}$ |
| Restaurants | (Wang et al., 2023a) 21.0 | 31.3$_{\pm 2.2}$ | 43.4$_{\pm 0.8}$ | 49.8$_{\pm 1.4}$ | **52.7**$_{\pm 1.6}$ |
| MultiNERD | - | 55.0$_{\pm 1.1}$ | 76.0$_{\pm 0.7}$ | **77.5**$_{\pm 0.3}$ | 77.2$_{\pm 0.6}$ |
| WikiEvents$_{\text{NER}}$ | (Sainz et al., 2022b) *49.1 | 76.9$_{\pm 5.1}$ | 80.7$_{\pm 0.7}$ | 80.2$_{\pm 0.7}$ | **81.3**$_{\pm 0.5}$ |
| WikiEvents$_{\text{EE}}$ | (Sainz et al., 2022b) *10.4 | 47.5$_{\pm 0.4}$ | 43.0$_{\pm 0.6}$ | 45.7$_{\pm 0.8}$ | **47.0**$_{\pm 1.9}$ |
| WikiEvents$_{\text{EAE}}$ | Sainz et al. (2022a) 35.9 | 51.6$_{\pm 0.5}$ | 51.9$_{\pm 0.4}$ | **52.5**$_{\pm 1.2}$ | 50.7$_{\pm 0.4}$ |
| Average SoTA | 42.6 | 45.4$_{\pm 0.5}$ | 58.4$_{\pm 0.5}$ | 58.3$_{\pm 0.7}$ | **60.0**$_{\pm 1.0}$ |
| Average all | - | 42.3$_{\pm 0.2}$ | 55.3$_{\pm 0.2}$ | 56.0$_{\pm 0.2}$ | **57.2**$_{\pm 0.5}$ |

of 13 F1 points on average. Despite dividing the evaluation benchmarks based on the domain, there is always some overlap between labels of train and evaluation benchmarks. For instance, the datasets E3C and WikiEvents share a large part of their schema with datasets like BC5CDR, ACE05, and RAMS. This phenomenon is reflected in the results.

GoLLIE surpasses by a large margin the current zero-shot SoTA methods Instruct-UIE (Wang et al., 2023a) and Entailment based IE (Sainz et al., 2022b). Compared to Instruct-UIE, the main differences are the backbone model, the amount of training data, and, the use or not of the guidelines. Instruct-UIE leverages the 11B FlanT5 which is a T5 fine-tuned on 473 NLP datasets. With respect to the data, Instruct-UIE leverages a total of 34 IE datasets (counting different tasks as datasets) from diverse domains, we only leverage 12 datasets. Contrary to our method they do not use guideline information. Still, our method performs significantly better suggesting that the guidelines have an important effect on the results.

PromptNER (Zhang et al., 2023a) also adds some definition information into the prompt in order to perform zero-shot NER. We compare our approach with them (represented as GPT-3.5) in Figure 1. Although their approach leverages guidelines too, our approach performs significantly better on all datasets, showing that LLMs (even with 175B parameters) struggle to follow guidelines. They solve this by adding examples in the context but are still far behind on a comparable setting (T5-XXL).

**Seen vs unseen labels:** Not all labels in the zero-shot datasets are unseen; there is an overlap between the labels in the training and zero-shot datasets. Although these labels may have very different annotation guidelines, we also report results on the set of labels to which it has not been exposed during training, to better understand the generalization capabilities of GoLLIE. The list of seen and unseen labels, as well as an extended analysis is available in Appendix B. Figure 4 aggregates the F1 scores across datasets for seen and unseen labels in the zero-shot scenario. All models exhibit slightly lower performance on unseen labels. For the baseline model, the performance drop is more pronounced. In contrast, GoLLIE demonstrates better generalization ability, showing a smaller gap in F1 scores between the seen and unseen labels. Also, the gap is smaller as the parameter count of our model increases.

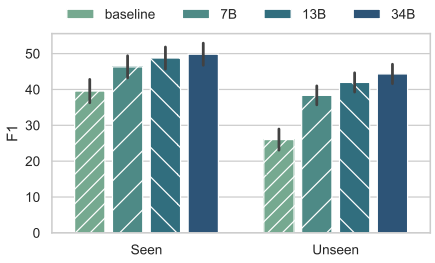



Figure 4: Seen vs unseen label zero-shot performance, results aggregated from all datasets.

**Model scaling:** Recent research has shown that increasing the parameter count of language models leads to improved generalization capabilities Brown et al. (2020). Higher parameter count yields superior average zero-shot performance. However, some datasets and tasks greatly benefit from a larger LLM, while others do not. We believe that some datasets do not see benefits from increasing the LLM size because their performance is hindered by the issues with the guidelines that we discuss in Section 5.3. While, in general, larger models achieve better results in both supervised and zero-shot settings, GoLLIE with a 7B parameter backbone already exhibits strong zero-shot capabilities.

### 5.3 Ablation study

We have performed an ablation to see the contribution of several components in the zero-shot evaluation. We analyzed the different regularization techniques proposed in Section 3.3. Additionally, we represent the baseline, i.e. when removing all components including guidelines, as "w/o *all*". Along with the mean zero-shot F1 we also provide the one-sided p-value with respect to GoLLIE.

Table 4: Ablation results.

| Model | F1 | p-value |
|---|---|---|
| GoLLIE | $55.3_{\pm 0.2}$ | - |
| w/o Shuffling | $55.9_{\pm 0.2}$ | $7.2e^{-2}$ |
| w/o Paraphrases | $54.8_{\pm 0.2}$ | $1.1e^{-1}$ |
| w/o Masking | $54.6_{\pm 0.6}$ | $1.0e^{-1}$ |
| w/o Dropout | $54.0_{\pm 0.2}$ | $4.0e^{-3}$ |
| w/o Candidates | $49.9_{\pm 0.2}$ | $2.2e^{-10}$ |
| w/o *all* (baseline) | $42.3_{\pm 0.1}$ | $5.1e^{-13}$ |

The class order shuffling, guideline paraphrasing, and class name masking seem to have no significant contribution to the final result, while class dropout although significant improvements are small. As further explained in Appendix D, the loss is only computed over the result tokens, inherently limiting the model's potential to overfit to the guidelines. In contrast, the representative annotation items give a stronger signal to the model. We see how definitions and representative candidates from the guidelines are complementary and help to improve each other.

## 6 Error analysis

In this section, we aim to better understand the effect of prompting LLMs with guidelines. We focus on specific labels across various datasets, with the results displayed in Table 5. Our analysis covers both successful and unsuccessful cases of entity labeling by GoLLIE. For the latter, we also aim to identify the reasons why the model fails to correctly label these entities. Further analyses on malformed outputs or hallucinations are discussed in Appendix C.

Table 5: This table shows the F1 scores for specific labels from different datasets. The guideline column is a small summary of the actual guideline used to prompt the model.

| Dataset | Label | Guideline | Baseline | |
|---|---|---|---|---|
| MultiNERD | Media | Titles of films, books, songs, albums, fictional characters and languages. | 13.6 | 69.1 |
| CASIE | Vul. Patch | When a software company addresses a vulnerability by releasing an update. | 27.7 | 70.5 |
| Movie | Trailer | Refers to a short promotional video or preview of a movie. | 00.0 | 76.4 |
| AI | Task | Particular research task or problem within a specific AI research field. | 02.7 | 63.9 |
| MultiNERD | Time | Specific and well-defined time intervals, such as eras, historical periods, centuries, years and important days. | 01.4 | 03.5 |
| Movie | Plot | Recurring concept, event, or motif that plays a significant role in the development of a movie. | 00.4 | 05.1 |
| AI | Misc | Named entities that are not included in any other category. | 01.1 | 05.2 |
| Literature | Misc | Named entities that are not included in any other category. | 03.7 | 30.8 |
| Literature | Writer | Individual actively engaged in the creation of literary works. | 04.2 | 65.1 |
| Literature | Person | Person name that is not a writer. | 33.5 | 49.4 |
| Science | Scientist | A person who is studying or has expert knowledge of a natural science field. | 02.1 | 05.8 |
| Science | Person | Person name that is not a scientist. | 46.1 | 45.9 |
| Politics | Polit. Party | Organization that compete in a particular country's elections. | 11.2 | 34.9 |

**The details are in the guidelines:** Labels such as MEDIA, VULNERABILITYPATCH, TRAILER, and TASK are inherently polysemous, making it challenging to determine the appropriate categorization based solely on the label name. As a result, the baseline struggles to effectively classify items under these labels due to having insufficient information. Conversely, GoLLIE successfully follows the guidelines, underscoring their utility.

**When the annotations do not comply with the guidelines:** In the case of the TIME label of the MultiNERD dataset, we found that our model labels years as TIME entities. This is correct according to the annotation guidelines. Surprisingly, years are not labeled as entities in the dataset. In this case, GoLLIE successfully follows the guidelines; unfortunately, the dataset annotations do not.

**Ambiguous labels:** The MISCELLANEOUS category, used by CoNLL03 and CrossNER datasets, refers to any named entity that is not included in the predefined categories set by the dataset. This definition is highly ambiguous and serves as a catch-all for various elements that do not fit into any of the predefined categories. Similarly, the PLOT category of the Movie dataset is used to label a wide range of elements. For example, events in a movie (e.g., murder, horse racing), characters (e.g., vampires, zombies), and the country of origin (e.g., British), among others. This lack of specificity hinders the development of consistent rules or guidelines for tagging such elements (Ratinov & Roth, 2009), which is a problem for humans and machines alike. As a consequence, GoLLIE also fails to label them accurately.

**Conflicts Between Fine-Grained and Coarse Entities:** The CrossNER dataset introduces two labels for person names within each domain. For example, in the Science domain, the labels SCIENTIST and PERSON are used. The former is used to label any person that is not a Scientist. Similarly, the Literature domain includes the labels WRITER and PERSON. The guidelines assist GoLLIE in correctly labeling entities as WRITER. However, GoLLIE still categorizes individuals as *Person* even when they are *Scientist*, despite the guidelines. This is not technically incorrect, as every scientist is, by definition, also a person.

**Strong Label Preconceptions:** In its Political domain set, CrossNER includes the label POLITICAL PARTY. GoLLIE outperforms the baseline, once again demonstrating the utility of providing the model with guidelines. However, we often find that the model categorizes political parties as organizations. As listed in Table 1, most of the pre-training datasets originate from the news domain, where political parties are a common entity. However, none of the fine-tuning datasets include the POLITICAL PARTY entity; they are instead categorized as ORGANIZATION. Consequently, during inference, the model consistently labels political parties as organizations. We believe this issue can be resolved by expanding the number and diversity of the fine-tuning datasets.

In summary, we anticipate that **GoLLIE will perform well on labels with well-defined and clearly bounded guidelines**. On the other hand, ambiguous labels or very coarse labels pose challenges. In this regard, we believe that GoLLIE would benefit from learning to follow instructions such as *"Label always the most specific class"* or *"Annotate this class in the absence of other specific class"*. We also expect that GoLLIE would benefit from expanding the number and diversity of the pre-training datasets.

## 7 CONCLUSIONS

In this paper, we introduce GoLLIE, an LLM specifically fine-tuned to comply with annotation guidelines that were devised to help humans annotate the dataset. A comprehensive zero-shot evaluation empirically demonstrates that annotation guidelines are of great value for LLMs, as GoLLIE successfully leverages them. GoLLIE achieves better zero-shot results than previous attempts at zero-shot IE which do not leverage the guidelines, or use models not finetuned for following guidelines.

GoLLIE is a significant progress towards the development of models that can generalize to unseen IE tasks. In the future, we plan to enhance GoLLIE by using a larger and more diverse set of pre-training datasets. We will also improve the model's performance with ambiguous and coarse labels by expanding the set of instructions that the model can follow.

## ACKNOWLEDGMENTS

This work has been partially supported by the Basque Government (Research group funding IT-1805-22 and ICL4LANG project, grant no. KK-2023/00094). We are also thankful to several MCIN/AEI/10.13039/501100011033 projects: (i) DeepKnowledge (PID2021-127777OB-C21) and ERDF A way of making Europe; (ii) Disargue (TED2021-130810B-C21) and European Union NextGenerationEU/PRTR; (iii) AWARE (TED2021-131617B-I00) and European Union NextGenerationEU/PRTR. This work has also been partially funded by the LUMINOUS project (HORIZON-CL4-2023-HUMAN-01-21-101135724). Oscar Sainz is supported by a doctoral grant from the Basque Government (PRE_2023_2_0137). Rodrigo Agerri currently holds the RYC-2017-23647 fellowship (MCIN/AEI/10.13039/501100011033 and by ESF Investing in your future).

## REFERENCES

Sidney Black, Stella Biderman, Eric Hallahan, Quentin Anthony, Leo Gao, Laurence Golding, Horace He, Connor Leahy, Kyle McDonell, Jason Phang, Michael Pieler, Usvsn Sai Prashanth, Shivanshu Purohit, Laria Reynolds, Jonathan Tow, Ben Wang, and Samuel Weinbach. GPT-NeoX-20B: An open-source autoregressive language model. In Angela Fan, Suzana Ilic, Thomas Wolf, and Matthias Gallé (eds.), *Proceedings of BigScience Episode #5 – Workshop on Challenges & Perspectives in Creating Large Language Models*, pp. 95–136, virtual+Dublin, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.bigscience-1.9. URL https://aclanthology.org/2022.bigscience-1.9.

Terra Blevins, Hila Gonen, and Luke Zettlemoyer. Prompting language models for linguistic structure. In Anna Rogers, Jordan L. Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pp. 6649–6663. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.acl-long.367. URL https://doi.org/10.18653/v1/2023.acl-long.367.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared 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 M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html.

Pei Chen, Haotian Xu, Cheng Zhang, and Ruihong Huang. Crossroads, buildings and neighborhoods: A dataset for fine-grained location recognition. In Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz (eds.), *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 3329–3339, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.243. URL https://aclanthology.org/2022.naacl-main.243.

Yi Chen, Haiyun Jiang, Lemao Liu, Shuming Shi, Chuang Fan, Min Yang, and Ruifeng Xu. An empirical study on multiple information sources for zero-shot fine-grained entity typing. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 2668–2678, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.210. URL https://aclanthology.org/2021.emnlp-main.210.

Zekai Chen, Mariann Micsinai Balan, and Kevin Brown. Language models are few-shot learners for prognostic prediction. *CoRR*, abs/2302.12692, 2023. doi: 10.48550/arXiv.2302.12692. URL https://doi.org/10.48550/arXiv.2302.12692.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. Palm: Scaling language modeling with pathways. *CoRR*, abs/2204.02311, 2022. doi: 10.48550/arXiv.2204.02311. URL https://doi.org/10.48550/arXiv.2204.02311.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Y. Zhao, Yanping Huang, Andrew M. Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. Scaling instruction-finetuned language models. *CoRR*, abs/2210.11416, 2022. doi: 10.48550/arXiv.2210.11416. URL https://doi.org/10.48550/arXiv.2210.11416.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pp. 8440–8451. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.acl-main.747. URL https://doi.org/10.18653/v1/2020.acl-main.747.

Leon Derczynski, Kalina Bontcheva, and Ian Roberts. Broad Twitter corpus: A diverse named entity recognition resource. In Yuji Matsumoto and Rashmi Prasad (eds.), *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pp. 1169–1179, Osaka, Japan, December 2016. The COLING 2016 Organizing Committee. URL https://aclanthology.org/C16-1111.

Leon Derczynski, Eric Nichols, Marieke van Erp, and Nut Limsopatham. Results of the WNUT2017 shared task on novel and emerging entity recognition. In Leon Derczynski, Wei Xu, Alan Ritter, and Tim Baldwin (eds.), *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pp. 140–147, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-4418. URL https://aclanthology.org/W17-4418.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms. *CoRR*, abs/2305.14314, 2023. doi: 10.48550/arXiv.2305.14314. URL https://doi.org/10.48550/arXiv.2305.14314.

Jesse Dodge, Maarten Sap, Ana Marasović, William Agnew, Gabriel Ilharco, Dirk Groeneveld, Margaret Mitchell, and Matt Gardner. Documenting large webtext corpora: A case study on the colossal clean crawled corpus. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 1286–1305, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.98. URL https://aclanthology.org/2021.emnlp-main.98.

Seth Ebner, Patrick Xia, Ryan Culkin, Kyle Rawlins, and Benjamin Van Durme. Multi-sentence argument linking. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 8057–8077, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.718. URL https://aclanthology.org/2020.acl-main.718.

Hermenegildo Fabregat, Juan Martínez-Romo, and Lourdes Araujo. Overview of the DIANN task: Disability annotation task. In Paolo Rosso, Julio Gonzalo, Raquel Martínez, Soto Montalvo, and Jorge Carrillo de Albornoz (eds.), *Proceedings of the Third Workshop on Evaluation of Human Language Technologies for Iberian Languages (IberEval 2018) co-located with 34th Conference of the Spanish Society for Natural Language Processing (SEPLN 2018), Sevilla, Spain, September 18th, 2018*, volume 2150 of *CEUR Workshop Proceedings*, pp. 1–14. CEUR-WS.org, 2018. URL https://ceur-ws.org/Vol-2150/overview-diann-task.pdf.

Besnik Fetahu, Sudipta Kar, Zhiyu Chen, Oleg Rokhlenko, and Shervin Malmasi. Semeval-2023 task 2: Fine-grained multilingual named entity recognition (multiconer 2). In Atul Kr. Ojha, A. Seza Dogruöz, Giovanni Da San Martino, Harish Tayyar Madabushi, Ritesh Kumar, and Elisa Sartori (eds.), *Proceedings of the The 17th International Workshop on Semantic Evaluation, SemEval@ACL 2023, Toronto, Canada, 13-14 July 2023*, pp. 2247–2265. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.semeval-1.310. URL https://doi.org/10.18653/v1/2023.semeval-1.310.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/pdf?id=sE7-XhLxHA.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Rezarta Islamaj Doğan and Zhiyong Lu. An improved corpus of disease mentions in PubMed citations. In Kevin B. Cohen, Dina Demner-Fushman, Sophia Ananiadou, Bonnie Webber, Jun'ichi Tsujii, and John Pestian (eds.), *BioNLP: Proceedings of the 2012 Workshop on Biomedical Natural Language Processing*, pp. 91–99, Montréal, Canada, June 2012. Association for Computational Linguistics. URL https://aclanthology.org/W12-2411.

Veysel Kocaman and David Talby. Biomedical named entity recognition at scale. In Alberto Del Bimbo, Rita Cucchiara, Stan Sclaroff, Giovanni Maria Farinella, Tao Mei, Marco Bertini, Hugo Jair Escalante, and Roberto Vezzani (eds.), *Pattern Recognition. ICPR International Workshops and Challenges*, pp. 635–646, Cham, 2021. Springer International Publishing. ISBN 978-3-030-68763-2.

Aman Kumar and Binil Starly. "fabner": information extraction from manufacturing process science domain literature using named entity recognition. *J. Intell. Manuf.*, 33(8):2393–2407, 2022. doi: 10.1007/s10845-021-01807-x. URL https://doi.org/10.1007/s10845-021-01807-x.

Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. Zero-shot relation extraction via reading comprehension. In Roger Levy and Lucia Specia (eds.), *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pp. 333–342, Vancouver, Canada, August 2017. Association for Computational Linguistics. doi: 10.18653/v1/K17-1034. URL https://aclanthology.org/K17-1034.

Peng Li, Tianxiang Sun, Qiong Tang, Hang Yan, Yuanbin Wu, Xuanjing Huang, and Xipeng Qiu. CodeIE: Large code generation models are better few-shot information extractors. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15339–15353, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.855. URL https://aclanthology.org/2023.acl-long.855.

Sha Li, Heng Ji, and Jiawei Han. Document-level event argument extraction by conditional generation. In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou (eds.), *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 894–908, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.69. URL https://aclanthology.org/2021.naacl-main.69.

Ying Lin, Heng Ji, Fei Huang, and Lingfei Wu. A joint neural model for information extraction with global features. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 7999–8009, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.713. URL https://aclanthology.org/2020.acl-main.713.

Xiao Ling and Daniel S. Weld. Fine-grained entity recognition. In Jörg Hoffmann and Bart Selman (eds.), *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence, July 22-26, 2012, Toronto, Ontario, Canada*, pp. 94–100. AAAI Press, 2012. doi: 10.1609/AAAI.V26I1.8122. URL https://doi.org/10.1609/aaai.v26i1.8122.

Jingjing Liu, Panupong Pasupat, Scott Cyphers, and James R. Glass. 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*, pp. 8386–8390. IEEE, 2013. doi: 10.1109/ICASSP.2013.6639301. URL https://doi.org/10.1109/ICASSP.2013.6639301.

Zihan Liu, Yan Xu, Tiezheng Yu, Wenliang Dai, Ziwei Ji, Samuel Cahyawijaya, Andrea Madotto, and Pascale Fung. Crossner: Evaluating cross-domain named entity recognition. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pp. 13452–13460. AAAI Press, 2021a. URL https://ojs.aaai.org/index.php/AAAI/article/view/17587.

Zihan Liu, Yan Xu, Tiezheng Yu, Wenliang Dai, Ziwei Ji, Samuel Cahyawijaya, Andrea Madotto, and Pascale Fung. Crossner: Evaluating cross-domain named entity recognition. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pp. 13452–13460. AAAI Press, 2021b. doi: 10.1609/aaai.v35i15.17587. URL https://doi.org/10.1609/aaai.v35i15.17587.

Jie Lou, Yaojie Lu, Dai Dai, Wei Jia, Hongyu Lin, Xianpei Han, Le Sun, and Hua Wu. Universal information extraction as unified semantic matching. In Brian Williams, Yiling Chen, and Jennifer Neville (eds.), *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023*, pp. 13318–13326. AAAI Press, 2023. doi: 10.1609/aaai.v37i11.26563. URL https://doi.org/10.1609/aaai.v37i11.26563.

Yaojie Lu, Qing Liu, Dai Dai, Xinyan Xiao, Hongyu Lin, Xianpei Han, Le Sun, and Hua Wu. Unified structure generation for universal information extraction. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pp. 5755–5772. Association for Computational Linguistics, 2022a. doi: 10.18653/v1/2022.acl-long.395. URL https://doi.org/10.18653/v1/2022.acl-long.395.

Yaojie Lu, Qing Liu, Dai Dai, Xinyan Xiao, Hongyu Lin, Xianpei Han, Le Sun, and Hua Wu. Unified structure generation for universal information extraction. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5755–5772, Dublin, Ireland, May 2022b. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.395. URL https://aclanthology.org/2022.acl-long.395.

Inbal Magar and Roy Schwartz. Data contamination: From memorization to exploitation. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 157–165, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-short.18. URL https://aclanthology.org/2022.acl-short.18.

Bernardo Magnini, Begoña Altuna, Alberto Lavelli, Manuela Speranza, and Roberto Zanoli. The E3C project: European clinical case corpus. In Jon Alkorta, Itziar Gonzalez-Dios, Aitziber Atutxa, Koldo Gojenola, Eugenio Martínez-Cámara, Álvaro Rodrigo, and Paloma Martínez (eds.), *Proceedings of the Annual Conference of the Spanish Association for Natural Language Processing: Projects and Demonstrations (SEPLN-PD 2021) co-located with the Conference of the Spanish Society for Natural Language Processing (SEPLN 2021), Málaga, Spain, September, 2021*, volume 2968 of *CEUR Workshop Proceedings*, pp. 17–20. CEUR-WS.org, 2021. URL https://ceur-ws.org/Vol-2968/paper5.pdf.

Bonan Min, Hayley Ross, Elior Sulem, Amir Pouran Ben Veyseh, Thien Huu Nguyen, Oscar Sainz, Eneko Agirre, Ilana Heintz, and Dan Roth. Recent advances in natural language processing via large pre-trained language models: A survey. *ACM Comput. Surv.*, 56(2), sep 2023. ISSN 0360-0300. doi: 10.1145/3605943. URL https://doi.org/10.1145/3605943.

Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M. Saiful Bari, Sheng Shen, Zheng Xin Yong, Hailey Schoelkopf, Xiangru Tang, Dragomir Radev, Alham Fikri Aji, Khalid Almubarak, Samuel Albanie, Zaid Alyafeai, Albert Webson, Edward Raff, and Colin Raffel. Crosslingual generalization through multitask finetuning. In Anna Rogers, Jordan L. Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pp. 15991–16111. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.acl-long.891. URL https://doi.org/10.18653/v1/2023.acl-long.891.

Rasha Obeidat, Xiaoli Fern, Hamed Shahbazi, and Prasad Tadepalli. Description-based zero-shot fine-grained entity typing. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 807–814, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1087. URL https://aclanthology.org/N19-1087.

Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Alessandro Cappelli, Hamza Alobeidli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. The refinedweb dataset for falcon LLM: outperforming curated corpora with web data, and web data only. *CoRR*, abs/2306.01116, 2023. doi: 10.48550/arXiv.2306.01116. URL https://doi.org/10.48550/arXiv.2306.01116.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Hwee Tou Ng, Anders Björkelund, Olga Uryupina, Yuchen Zhang, and Zhi Zhong. Towards robust linguistic analysis using OntoNotes. In Julia Hockenmaier and Sebastian Riedel (eds.), *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pp. 143–152, Sofia, Bulgaria, August 2013. Association for Computational Linguistics. URL https://aclanthology.org/W13-3516.

Ofir Press, Noah A. Smith, and Mike Lewis. Train short, test long: Attention with linear biases enables input length extrapolation. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL https://openreview.net/forum?id=R8sQPpGCv0.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67, 2020. URL http://jmlr.org/papers/v21/20-074.html.

Lev Ratinov and Dan Roth. Design challenges and misconceptions in named entity recognition. In Suzanne Stevenson and Xavier Carreras (eds.), *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL-2009)*, pp. 147–155, Boulder, Colorado, June 2009. Association for Computational Linguistics. URL https://aclanthology.org/W09-1119.

Baptiste Rozière, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Tal Remez, Jérémy Rapin, Artyom Kozhevnikov, Ivan Evtimov, Joanna Bitton, Manish Bhatt, Cristian Canton-Ferrer, Aaron Grattafiori, Wenhan Xiong, Alexandre Défossez, Jade Copet, Faisal Azhar, Hugo Touvron, Louis Martin, Nicolas Usunier, Thomas Scialom, and Gabriel Synnaeve. Code llama: Open foundation models for code. *CoRR*, abs/2308.12950, 2023. doi: 10.48550/arXiv.2308.12950. URL https://doi.org/10.48550/arXiv.2308.12950.

Oscar Sainz, Oier Lopez de Lacalle, Gorka Labaka, Ander Barrena, and Eneko Agirre. Label verbalization and entailment for effective zero and few-shot relation extraction. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 1199–1212, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.92. URL https://aclanthology.org/2021.emnlp-main.92.

Oscar Sainz, Itziar Gonzalez-Dios, Oier Lopez de Lacalle, Bonan Min, and Eneko Agirre. Textual entailment for event argument extraction: Zero- and few-shot with multi-source learning. In Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz (eds.), *Findings of the Association for Computational Linguistics: NAACL 2022*, pp. 2439–2455, Seattle, United States, July 2022a. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-naacl.187. URL https://aclanthology.org/2022.findings-naacl.187.

Oscar Sainz, Haoling Qiu, Oier Lopez de Lacalle, Eneko Agirre, and Bonan Min. ZS4IE: A toolkit for zero-shot information extraction with simple verbalizations. In Hannaneh Hajishirzi, Qiang Ning, and Avi Sil (eds.), *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: System Demonstrations*, pp. 27–38, Hybrid: Seattle, Washington + Online, July 2022b. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-demo.4. URL https://aclanthology.org/2022.naacl-demo.4.

Oscar Sainz, Jon Campos, Iker García-Ferrero, Julen Etxaniz, Oier Lopez de Lacalle, and Eneko Agirre. NLP evaluation in trouble: On the need to measure LLM data contamination for each benchmark. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 10776–10787, Singapore, December 2023a. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.722. URL https://aclanthology.org/2023.findings-emnlp.722.

Oscar Sainz, Jon Ander Campos, Iker García-Ferrero, Julen Etxaniz, and Eneko Agirre. Did chatgpt cheat on your test?, Jun 2023b. URL https://hitz-zentroa.github.io/lm-contamination/blog/.

Taneeya Satyapanich, Francis Ferraro, and Tim Finin. Casie: Extracting cybersecurity event information from text. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8749–8757, Apr. 2020. doi: 10.1609/aaai.v34i05.6401. URL https://ojs.aaai.org/index.php/AAAI/article/view/6401.

Teven Le Scao and Alexander M. Rush. How many data points is a prompt worth? In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tür, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou (eds.), *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pp. 2627–2636. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.naacl-main.208. URL https://doi.org/10.18653/v1/2021.naacl-main.208.

Timo Schick and Hinrich Schütze. Exploiting cloze-questions for few-shot text classification and natural language inference. In Paola Merlo, Jörg Tiedemann, and Reut Tsarfaty (eds.), *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021*, pp. 255–269. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.eacl-main.20. URL https://doi.org/10.18653/v1/2021.eacl-main.20.

Karan Singhal, Shekoofeh Azizi, Tao Tu, S. Sara Mahdavi, Jason Wei, Hyung Won Chung, Nathan Scales, Ajay Kumar Tanwani, Heather Cole-Lewis, Stephen Pfohl, Perry Payne, Martin Seneviratne, Paul Gamble, Chris Kelly, Nathaneal Schärli, Aakanksha Chowdhery, Philip Andrew Mansfield, Blaise Agüera y Arcas, Dale R. Webster, Gregory S. Corrado, Yossi Matias, Katherine Chou, Juraj Gottweis, Nenad Tomasev, Yun Liu, Alvin Rajkomar, Joelle K. Barral, Christopher Semturs, Alan Karthikesalingam, and Vivek Natarajan. Large language models encode clinical knowledge. *CoRR*, abs/2212.13138, 2022. doi: 10.48550/ARXIV.2212.13138. URL https://doi.org/10.48550/arXiv.2212.13138.

Karan Singhal, Shekoofeh Azizi, Tao Tu, S Sara Mahdavi, Jason Wei, Hyung Won Chung, Nathan Scales, Ajay Tanwani, Heather Cole-Lewis, Stephen Pfohl, et al. Large language models encode clinical knowledge. *Nature*, pp. 1–9, 2023.

Zeqi Tan, Shen Huang, Zixia Jia, Jiong Cai, Yinghui Li, Weiming Lu, Yueting Zhuang, Kewei Tu, Pengjun Xie, and Fei Huang. DAMO-NLP at semeval-2023 task 2: A unified retrieval-augmented system for multilingual named entity recognition. In Atul Kr. Ojha, A. Seza Dogruöz, Giovanni Da San Martino, Harish Tayyar Madabushi, Ritesh Kumar, and Elisa Sartori (eds.), *Proceedings of the The 17th International Workshop on Semantic Evaluation, SemEval@ACL 2023, Toronto, Canada, 13-14 July 2023*, pp. 2014–2028. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.semeval-1.277. URL https://doi.org/10.18653/v1/2023.semeval-1.277.

Simone Tedeschi and Roberto Navigli. MultiNERD: A multilingual, multi-genre and fine-grained dataset for named entity recognition (and disambiguation). In Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz (eds.), *Findings of the Association for Computational Linguistics: NAACL 2022*, pp. 801–812, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-naacl.60. URL https://aclanthology.org/2022.findings-naacl.60.

Erik F. Tjong Kim Sang and Fien De Meulder. 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*, pp. 142–147, 2003. URL https://aclanthology.org/W03-0419.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models. *CoRR*, abs/2302.13971, 2023a. doi: 10.48550/arXiv.2302.13971. URL https://doi.org/10.48550/arXiv.2302.13971.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288, 2023b. doi: 10.48550/arXiv.2307.09288. URL https://doi.org/10.48550/arXiv.2307.09288.

Christopher Walker, Stephanie Strassel, Julie Medero, and Kazuaki Maeda. Ace 2005 multilingual training corpus. *Linguistic Data Consortium, Philadelphia*, 57:45, 2006. URL https://catalog.ldc.upenn.edu/LDC2006T06.

Ben Wang and Aran Komatsuzaki. Gpt-j-6b: A 6 billion parameter autoregressive language model, 2021.

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. Instructuie: Multi-task instruction tuning for unified information extraction. *CoRR*, abs/2304.08085, 2023a. doi: 10.48550/arXiv.2304.08085. URL https://doi.org/10.48550/arXiv.2304.08085.

Xingyao Wang, Sha Li, and Heng Ji. Code4Struct: Code generation for few-shot event structure prediction. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3640–3663, Toronto, Canada, July 2023b. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.202. URL https://aclanthology.org/2023.acl-long.202.

Xinyu Wang, Yong Jiang, Nguyen Bach, Tao Wang, Zhongqiang Huang, Fei Huang, and Kewei Tu. Improving named entity recognition by external context retrieving and cooperative learning. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (eds.), *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 1800–1812, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.142. URL https://aclanthology.org/2021.acl-long.142.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Atharva Naik, Arjun Ashok, Arut Selvan Dhanasekaran, Anjana Arunkumar, David Stap, Eshaan Pathak, Giannis Karamanolakis, Haizhi Gary Lai, Ishan Purohit, Ishani Mondal, Jacob Anderson, Kirby Kuznia, Krima Doshi, Kuntal Kumar Pal, Maitreya Patel, Mehrad Moradshahi, Mihir Parmar, Mirali Purohit, Neeraj Varshney, Phani Rohitha Kaza, Pulkit Verma, Ravsehaj Singh Puri, Rushang Karia, Savan Doshi, Shailaja Keyur Sampat, Siddhartha Mishra, Sujan Reddy A, Sumanta Patro, Tanay Dixit, and Xudong Shen. Super-naturalinstructions: Generalization via declarative instructions on 1600+ NLP tasks. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pp. 5085–5109. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.emnlp-main.340. URL https://doi.org/10.18653/v1/2022.emnlp-main.340.

Chih-Hsuan Wei, Yifan Peng, Robert Leaman, Allan Peter Davis, Carolyn J Mattingly, Jiao Li, Thomas C Wiegers, and Zhiyong Lu. Assessing the state of the art in biomedical relation extraction: overview of the biocreative v chemical-disease relation (cdr) task. *Database*, 2016, 2016.

Renzo M. Rivera Zabala, Paloma Martinez, and Isabel Segura-Bedmar. A hybrid bi-lstm-crf model to recognition of disabilities from biomedical texts. In *Proceedings of the Third Workshop on Evaluation of Human Language Technologies for Iberian Languages*, 2018. URL https://ceur-ws.org/Vol-2150/DIANN_paper5.pdf.

Mozhi Zhang, Hang Yan, Yaqian Zhou, and Xipeng Qiu. Promptner: A prompting method for few-shot named entity recognition via k nearest neighbor search. *CoRR*, abs/2305.12217, 2023a. doi: 10.48550/arXiv.2305.12217. URL https://doi.org/10.48550/arXiv.2305.12217.

Sheng Zhang, Hao Cheng, Jianfeng Gao, and Hoifung Poon. Optimizing bi-encoder for named entity recognition via contrastive learning. In *The Eleventh International Conference on Learning Representations*, 2023b. URL https://openreview.net/forum?id=9EAQVEINuum.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona T. Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. OPT: open pre-trained transformer language models. *CoRR*, abs/2205.01068, 2022. doi: 10.48550/arXiv.2205.01068. URL https://doi.org/10.48550/arXiv.2205.01068.

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D. Manning. Position-aware attention and supervised data improve slot filling. In Martha Palmer, Rebecca Hwa, and Sebastian Riedel (eds.), *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 35–45, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1004. URL https://aclanthology.org/D17-1004.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena. *CoRR*, abs/2306.05685, 2023. doi: 10.48550/arXiv.2306.05685. URL https://doi.org/10.48550/arXiv.2306.05685.

Wenxuan Zhou, Sheng Zhang, Yu Gu, Muhao Chen, and Hoifung Poon. Universalner: Targeted distillation from large language models for open named entity recognition. *CoRR*, abs/2308.03279, 2023. doi: 10.48550/arXiv.2308.03279. URL https://doi.org/10.48550/arXiv.2308.03279.

```python
@dataclass
class Launcher(Template):
      """Refers to a vehicle designed primarily to transport payloads from the Earth's
      surface to space. Launchers can carry various payloads, including satellites,
      crewed spacecraft, and cargo, into various orbits or even beyond Earth's orbit.
      They are usually multi-stage vehicles that use rocket engines for propulsion."""

      mention: str
      """
      The name of the launcher vehicle.
      Such as: "Sturn V", "Atlas V", "Soyuz", "Ariane 5"
      """
      space_company: str # The company that operates the launcher. Such as: "Blue origin", "ESA", "Boeing"
      crew: List[str]
      """Names of the crew members boarding the Launcher.
      Such as: "Neil Armstrong", "Michael Collins", "Buzz Aldrin"
      """

@dataclass
class Mission(Template):
      """Any planned or accomplished journey beyond Earth's atmosphere with specific objectives,
      either crewed or uncrewed. It includes missions to satellites, the International
      Space Station (ISS), other celestial bodies, and deep space."""

      mention: str
      """
      The name of the mission.
      Such as: "Apollo 11", "Artemis", "Mercury"
      """
      date: str # The start date of the mission
      departure: str # The place from which the vehicle will be launched. Such as: "Florida", "Houston"
      destination: str # The place or planet to which the launcher will be sent. Such as "Moon", "low-orbit"


# This is the text to analyze
text = (
      "The Ares 3 mission to Mars is scheduled for 2032. The Starship rocket build by SpaceX will take off"
      "from Boca Chica, carrying the astronauts Max Rutherford, Elena Soto, and Jake Martinez."
)

# The annotation instances that take place in the text above are listed here
result = [
      Mission(mention='Ares 3', date='2032', departure='Boca Chica', destination='Mars'),
      Launcher(mention='Starship', space_company='SpaceX', crew=['Max Rutherford', 'Elena Soto', 'Jake Martinez'])
]
```

Figure 5: Example of generalization to custom tasks defined by the user.

# A EXAMPLES

In addition to NER, EE and EAE for which examples are shown in Figures 2 and 3 respectively, we also feed the model with data from RE and SF. The formulation of RE is similar to the NER but with two argument attributes. However, the SF task is more complex as shown in Figure 6. With this task, we added several layers of complexity to the input: extended definitions for each possible attribute (slot), optional arguments, and, fine-grained definitions of types such as Names, Values, or Strings. We also added constraints into the prompt to condition the model to just output the information of our desired query instead of every single template on the text. In the future, we would like to add more complex tasks to the training and evaluation to improve the capabilities and flexibility of the model. For more examples refer to the GitHub repository.

## A.1 EXAMPLE OF GENERALIZATION TO NEW CUSTOM TASKS

Our model allows the user to define custom annotation schemas using Python code. We provide an example where we define two new types of entities: Launcher and Mission. As shown in Figure 5, Launcher and Mission are not simple entities, they correspond to what we call *Template*, a class similar to *Entity* but with additional arguments, like the SF task. For example, the space_company or the crew of the launcher are some of the additional arguments we added to the schema. As shown in the example, the model's output (everything after `result = [`) satisfies the type constraints defined in the guidelines, attributes defined as strings are filled with strings and, the arguments defined as lists (like *crew*) are filled with lists. The model can correctly analyze the given sentence with our newly created annotation schema.

```python
# The following lines describe the task definition
@dataclass
class PersonTemplate(Template):
    """Person templates encodes the information about the given query
    Person entity."""

    query: str  # The Person entity query
    alternate_names: Optional[List[Name]] = None
    """Names used to refer to the query person that are distinct from the
    'official' name. Including: aliases, stage names, abbreviations ..."""
    date_of_birth: Optional[Value] = None
    """The date on which the query person was born."""
    age: Optional[Value] = None
    """A reported age of the query person."""
    city_of_birth: Optional[Name] = None
    """The geopolitical entity at the municipality level (city, town, or
    village) in which the query person was born"""
    date_of_death: Optional[Value] = None
    """The date of the query person's death."""

                    (Collapsed 36 more slots)
```

```python
# This is the text to analyze
text = "Mongolian Prime Minister M. Enkhbold met
with Liu Hongcai , vice minister of the
International Department of the Chinese Communist
Party Central Committee on Monday here ."

# The list called result contains the templates
# instances for the following entity queries:
#    - M. Enkhbold: PersonTemplate
#
result = [
    PersonTemplate(
        query="M. Enkhbold",
        countries_of_residence=[Name("Mongolian")],
        title=[String("Prime Minister")],
    ),
]
```

Figure 6: Example of the TACRED dataset converted to Slot Filling task represented as code.

## B Performance in Seen vs Unseen Labels: Further Analysis

Table 6: List of labels in the zero-shot datasets that overlap with the ones in the training datasets (seen) and the labels that do not overlap with the ones in the training datasets (unseen)

| Dataset | Seen Labels | Unseen Labels |
|---|---|---|
| BroadTwitter | Location, Organization, Person | - |
| $\text{CASIE}_{\text{EE}}$ | - | DatabreachAttack, PhisingAttack, RansomAttack, VulnerabilityDiscover, VulnerabilityPatch |
| AI | Product, Country, Person, Organization, Location, Miscellaneous | Field, Task, Algorithm, Researcher, Metric, University, ProgrammingLanguage, Conference |
| Literature | Event, Person, Location, Organization, Country, Miscellaneous | Book, Writer, Award, Poem, Magazine, LiteraryGenre |
| Music | Event, Country, Location, Organization, Person, Miscellaneous | MusicGenre, Song, Band, Album, MusicalArtist, MusicalInstrument, Award |
| Politics | Person, Organization, Location, Election, Event, Country, Miscellaneous | Politician, PoliticalParty |
| Science | Person, Organization, Country, Location, ChemicalElement, ChemicalCompound, Event, Miscellaneous | Scientist, University, Discipline, Enzyme, Protein, AstronomicalObject, AcademicJournal, Theory, Award |
| E3C | ClinicalEntity | - |
| FabNER | Biomedical | Material, ManufacturingProcess, MachineEquipment, Application, EngineeringFeatures, MechanicalProperties, ProcessCharacterization, ProcessParameters, EnablingTechnology, ConceptPrinciples, ManufacturingStandards |
| HarveyNER | - | Point, Area, Road, River |
| Movie | Year | Actor, Character, Director, Genre, Plot, Rating, RatingsAverage, Review, Song, Tittle, Trailer |
| Restaurants | Location, Price, Hours | Rating, Amenity, RestaurantName, Dish, Cuisine |
| MultiNERD | Person, Location, Organization, Biological, Disease, Event, Time, Vehicle | Animal, Celestial, Food, Instrument, Media, Plant, Mythological |
| $\text{WikiEvents}_{\text{NER}}$ | CommercialProduct, Facility, GPE, Location, MedicalHealthIssue, Money, Organization, Person, JobTitle, Numeric, Vehicle, Weapon | Abstract, BodyPart, Information, SideOfConflict |
| $\text{WikiEvents}_{\text{EE}}$ | ConflictEvent, ContactEvent, GenericCrimeEvent, JusticeEvent, MedicalEvent, MovementTransportEvent, PersonnelEvent, TransactionEvent | ArtifactExistanceEvent, CognitiveEvent, ControlEvent, DisasterEvent, LifeEvent |

Table 6 categorizes the labels for each zero-shot dataset into those that overlap with the training dataset and those that are completely unseen. We adhere to a strict approach in this classification. For instance, although the label COUNTRY does not appear in the training datasets, similar labels such as GEOPOLITICAL entity do. Therefore, we consider that the model has been exposed to this label during training.

While some labels in the zero-shot datasets overlap with those in the training dataset, the annotation guidelines for each label may vary significantly between datasets. Table 7 presents the micro-F1

Table 7: Micro F1 score for the seen and unseen labels in the zero-shot datasets.

| | Baseline | | GoLLIE | | GoLLIE 13B | | GoLLIE 34B | |
|---|---|---|---|---|---|---|---|---|
| **Dataset** | **Seen** | **Unseen** | **Seen** | **Unseen** | **Seen** | **Unseen** | **Seen** | **Unseen** |
| BroadTwitter | 39.0±0.6 | - | 49.5±0.8 | - | 51.4±1.8 | - | 50.3±2.1 | - |
| CASIE$_{EE}$ | - | 33.9±6.5 | - | 59.3±2.3 | - | 62.2±0.9 | - | 65.5±1.8 |
| AI | 43.5±1.4 | 21.1±0.3 | 57.8±1.2 | 60.0±1.2 | 57.8±0.8 | 55.8±4.7 | 57.7±2.8 | 64.2±1.3 |
| Literature | 34.6±0.2 | 43.6±1.5 | 54.6±3.6 | 67.4±3.0 | 52.4±0.2 | 64.6±0.5 | 52.7±2.1 | 63.7±2.8 |
| Music | 46.8±1.0 | 62.2±1.6 | 53.7±0.2 | 74.9±0.3 | 52.8±3.9 | 72.7±3.5 | 54.0±3.8 | 76.3±1.2 |
| Politics | 45.9±1.1 | 4.6±2.6 | 64.0±0.2 | 31.9±4.7 | 62.0±2.2 | 22.4±14.6 | 64.4±1.5 | 45.8±9.3 |
| Science | 38.7±0.8 | 34.7±3.0 | 52.7±1.7 | 58.8±1.5 | 52.7±1.0 | 60.4±0.9 | 52.5±0.4 | 60.5±0.7 |
| E3C | 59.8±0.3 | - | 59.0±0.7 | - | 59.0±0.9 | - | 60.0±0.4 | - |
| FabNER | 0.0±0.0 | 6.2±0.4 | 22.6±2.3 | 24.9±0.6 | 23.9±4.4 | 25.5±0.6 | 20.7±2.9 | 26.5±0.6 |
| HarveyNER | - | 23.2±0.4 | - | 37.3±1.8 | - | 41.3±0.9 | - | 38.9±0.5 |
| Movie | 31.5±0.7 | 46.1±1.5 | 58.7±2.3 | 63.8±0.5 | 47.3±3.1 | 65.3±1.0 | 42.7±2.3 | 66.1±1.4 |
| Restaurants | 18.0±1.1 | 38.7±2.8 | 33.2±2.7 | 49.9±1.5 | 38.0±3.6 | 57.1±0.2 | 46.0±4.2 | 57.2±0.9 |
| MultiNERD | 58.0±1.1 | 39.5±1.4 | 81.2±0.5 | 44.6±0.9 | 82.4±0.4 | 47.7±0.7 | 82.3±0.5 | 49.1±0.5 |
| WikiEvents$_{NER}$ | 77.2±5.1 | 0.0±0.0 | 81.5±0.7 | 0.0±0.0 | 80.9±0.8 | 0.0±0.0 | 82.1±0.5 | 3.5±2.6 |
| WikiEvents$_{EE}$ | 43.3±0.3 | 57.2±1.5 | 41.7±0.1 | 45.0±1.5 | 43.9±0.8 | 48.8±1.7 | 45.0±1.1 | 50.4±0.9 |
| Average | 41.2±0.4 | 31.6±0.6 | 54.6±0.3 | 47.5±0.6 | 54.2±0.4 | 48.0±1.3 | 54.7±0.9 | 51.4±1.0 |

scores for both seen and unseen labels across each zero-shot dataset. Generally, the models perform better on seen labels compared to unseen ones. However, there are instances where the reverse is true. This occurs when a dataset contains labels that, although overlapping with those in the training dataset, have vastly different annotation guidelines. As discussed in Section 5.3, the model has strong preconceptions for some labels, adversely affecting zero-shot performance. GoLLIE, trained to adhere to specific annotation guidelines, demonstrates greater robustness against these label preconceptions than the baseline model. Consequently, it achieves better results for both seen and unseen labels. GoLLIE can successfully handle both, seen and unseen labels from datasets that were not used during training. This ability underscores GoLLIE's superior generalization capabilities, largely attributable to its capability of leveraging annotation guidelines.

## C Model hallucinations

Table 8: Number impossible to parse outputs and number predicted labels that are hallucinations. F1 scores on the dataset are shown for reference.

| | GoLLIE | | |
|---|---|---|---|
| **Dataset** | **Impossible to Parse** | **Hallucinations** | **F1 Score** |
| BroadTwitter | 0±0 / 2002 | 0±0 / 1664 | 49.5±0.8 |
| CASIE$_{EE}$ | 1±0 / 199 | 6±1 / 1548 | 59.3±2.3 |
| CASIE$_{EAE}$ | 1±1 / 199 | 3±2 / 2804 | 50.0±1.1 |
| AI | 0±0 / 431 | 1±1 / 1292 | 59.1±1.1 |
| Literature | 0±0 / 416 | 0±0 / 2059 | 62.7±3.2 |
| Music | 0±0 / 465 | 6±2 / 3080 | 67.8±0.2 |
| Politics | 0±0 / 651 | 3±2 / 4142 | 57.2±1.0 |
| Science | 0±0 / 543 | 7±1 / 2700 | 55.5±1.6 |
| E3C | 0±0 / 851 | 1±0 / 688 | 59.0±0.7 |
| FabNER | 1±0 / 2064 | 13±3 / 4474 | 24.8±0.6 |
| HarveyNER | 0±0 / 1303 | 1±1 / 708 | 37.3±1.8 |
| Movie | 0±0 / 2443 | 1±0 / 3919 | 63.0±0.6 |
| Restaurants | 0±0 / 1521 | 3±0 / 1451 | 43.4±0.8 |
| MultiNERD | 49±11 / 32908 | 51±8 / 67142 | 76.0±0.7 |
| WikiEvents$_{NER}$ | 0±0 / 573 | 1±1 / 2666 | 80.7±0.7 |
| WikiEvents$_{EE}$ | 0±0 / 573 | 3±1 / 630 | 43.0±0.6 |
| WikiEvents$_{EAE}$ | 2±1 / 321 | 0±0 / 363 | 51.9±0.4 |

In this section, we evaluate the hallucinations generated by the model. We examine two different phenomena. First, we consider instances where the output is so corrupted that it is *impossible to parse*. In such cases, we treat the output as an empty list. Second, we look at instances where the model outputs a label *Hallucination*, that is, a label not defined among the input classes. In these instances, we remove the label from the output. As demonstrated in Table 8, for all the zero-shot datasets, both phenomena occur in less than 1% of the predictions. This demonstrates that GoLLIE is highly resistant to hallucinations and closely adheres to the classes defined in the input.

# D Extended training details

## D.1 Loss calculation

We have used the standard Next Token Prediction (NTP) loss to train our models. However, several regularizations that we applied to the models made the loss computed over the guideline tokens much higher than the actual output tokens. This is because we randomly shuffle the guidelines order, mask names, or, drop classes, which makes it impossible to predict what goes next. To avoid the loss of the guideline tokens overshadowing the actual output tokens loss, we decided to only compute the loss over the output tokens. This way, we can also avoid some overfitting of the guidelines. This resulted in faster training and better results overall.

## D.2 Dataset Detailts

Table 9 shows the number of examples for each training and zero-shot dataset. OntoNotes is generated semi-automatically and is orders of magnitude larger than the other ones. Therefore, for each training epoch, we sample 30.000 random examples from the training set. The models were trained for 3 epochs with an effective batch size of 32 and a learning rate of 3e-4 with a cosine scheduler. Therefore, we perform 15,485 training steps.

Regarding the splits, we use the standard train, dev test splits for every dataset. In the case of ACE, we follow the split provided by Lin et al. (2020). In the case of CASIE, we took the first 200 instances as validation and the last 2000 as test.

Table 9: Number of examples for each training and zero-shot dataset

| **Dataset** | **Train** | **Dev** | **Test** |
|---|---|---|---|
| $ACE05_{NER}$ | 19217 | 676 | 901 |
| $ACE05_{RE}$ | 19217 | 901 | 676 |
| $ACE05_{EE}$ | 19217 | 676 | 901 |
| $ACE05_{EAE}$ | 3843 | 397 | 368 |
| $ACE05_{RC}$ | 5691 | - | - |
| $ACE05_{VER}$ | 19217 | - | - |
| BC5CDR | 4561 | 4582 | 4798 |
| CoNLL 2003 | 14041 | 3250 | 3453 |
| DIANN | 3976 | 793 | 1309 |
| NCBIDisease | 5433 | 924 | 941 |
| Ontonotes 5 | *30000* | 15680 | 12217 |
| RAMS | 7329 | 924 | 871 |
| TACRED | 10027 | 3896 | 2311 |
| WNUT 2017 | 3394 | 1009 | 1287 |
| Total | 165163 | 33708 | 30033 |
| BroadTwitter | - | - | 2002 |
| $CASIE_{EE}$ | - | - | 199 |
| $CASIE_{EAE}$ | - | - | 199 |
| AI | - | - | 431 |
| Literature | - | - | 416 |
| Music | - | - | 465 |
| Politics | - | - | 651 |
| Science | - | - | 543 |
| E3C | - | - | 851 |
| FabNER | - | - | 2064 |
| HarveyNER | - | - | 1303 |
| Movie | - | - | 2443 |
| Restaurants | - | - | 1521 |
| MultiNERD | - | - | 32908 |
| $WikiEvents_{NER}$ | - | - | 573 |
| $WikiEvents_{EE}$ | - | - | 573 |
| $WikiEvents_{EAE}$ | - | - | 321 |
| Total | - | - | 47463 |

Table 10: Details about the training resources required for each model.

| Model | Hardware | FLOPs | Time (h) | $CO_2$eq (kg) |
|---|---|---|---|---|
| Baseline | 1xA100 | $4.5e^{18}$ | 17.3 | 0.61 |
| GoLLIE | 1xA100 | $11.9e^{18}$ | 44.5 | 1.57 |
| 13B | 1xA100 | $22.7e^{18}$ | 79.5 | 2.80 |
| 34B | 2xA100 | $55.8e^{18}$ | 94.6 | 6.67 |

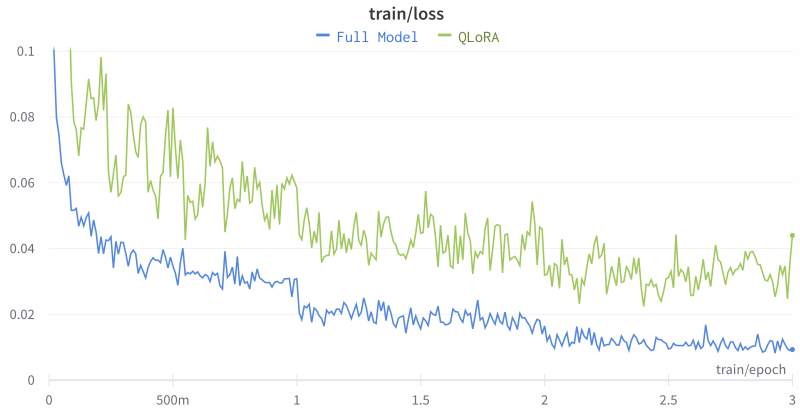
### D.3 Carbon footprint

Fine-tuning LLMs is not as expensive as pre-training these models. Still, we believe that it is important to measure and document the costs that our experiments have on our planet. We provide the resources required for a single run of our experiments in Table 10. All the experiments were done on our private infrastructure. For the carbon footprint estimation, we estimated the values considering a 400W consumption per GPU with a 0.141 kg/kWh carbon intensity[†].

### D.4 LoRA vs Full Model Fine-Tuning

We conducted preliminary experiments to compare the performance of QLoRA (Hu et al., 2022; Dettmers et al., 2023) with that of training all the parameters in the model. These preliminary experiments were conducted using the LLaMA2 7B model Touvron et al. (2023b) and an early version of the code. However, the experimental setup for both models was identical. Both approaches were prompted with guidelines. First, we compared the training loss of both approaches. Figure 7 shows that when fine-tuning all the parameters, the loss decreases much more rapidly than when training only the LoRA layers. It also achieves a lower loss at the end of training. However, when evaluating the model at the end of the first and third epochs, we observed that training the full model performs very poorly, as shown in Table 11. We hypothesize that when training all the parameters, the model overfits quickly (indicated by the lower training loss) and memorizes the training data. On the other hand, training only the LoRA layers, which represent around 0.5% of the total model weights, introduces a bottleneck that prevents the model from memorizing the training dataset. It is also noteworthy that the QLoRA approach was trained using just one Nvidia A100 80GB GPU thanks to the 4-bit quantization of the frozen model Dettmers et al. (2023). Training the full model required a minimum of four Nvidia A100 80GB GPUs to fit the model into memory. We used DeepSpeed [‡] to distribute the model across the four GPUs for training. Due to the high cost of training, we did not perform an extensive hyperparameter search for the full model.

Figure 7: Training loss of fine-tuning the full model vs training LoRA layers only.




[†]Statistic taken from `https://app.electricitymaps.com/map`

[‡]`github.com/microsoft/DeepSpeed`

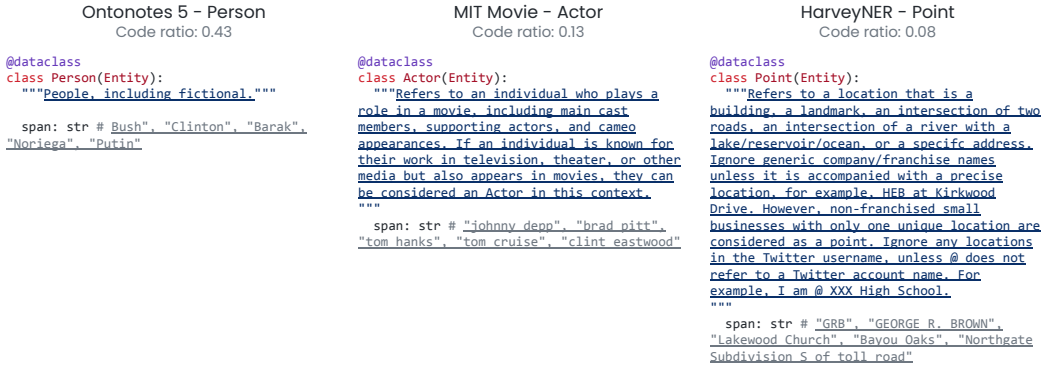
Table 11: F1 scores achieved when training the full model vs only training the LoRA Layers at the end of the first and third epoch.

| Training | Epoch | Precision | LR | HarveyNer | FabNER | Restaurant | Movie | CASIE$_{EE}$ | CoNLL03 |
|---|---|---|---|---|---|---|---|---|---|
| Full | 1 | BF16 | 1e−4 | 0.00 | 0.00 | 0.25 | 4.74 | 0.00 | 85.57 |
| Full | 3 | BF16 | 1e−4 | 3.45 | 0.21 | **46.7** | 16.72 | 0.42 | 84.83 |
| QLoRA | 1 | 4Bit + BF16 | 2e−3 | 34.98 | **20.78** | 45.01 | **51.14** | 55.83 | 91.41 |
| QLoRA | 3 | 4Bit + BF16 | 2e−3 | **35.34** | 16.21 | 39.07 | 44.18 | **57.93** | **93.14** |

## E HANDLING DATASETS WITH HUNDREDS OF LABELS AND CODE-STYLE PROMPT OVERHEAD

In our research, we focus on datasets with fewer than 20 labels. However, some datasets, such as FIGER Ling & Weld (2012), include hundreds of fine-grained labels. Including guidelines in datasets with hundreds of labels can make inputs excessively long, exceeding the context size of current Large Language Models (LLMs). This is a known constraint in LLMs, and recently, significant research effort has been directed towards algorithms that efficiently increase the context window size Press et al. (2022). We anticipate that future LLMs will have a context window large enough to accommodate not only more labels but also more detailed guidelines. For the time being, this problem can be mitigated by batching the labels into multiple input examples. Instead of prompting the model with, for example, 100 labels in a single input, it is possible to prompt the model with 10 inputs, each incorporating 10 labels, and then combine all the outputs into a single response. In any case, handling datasets with a large number of labels remains a limitation of GoLLIE.

Figure 8: Percentage of characters from the input required to represent the code-style prompt for different labels. For detailed guidelines, the code is a small fraction of the input.




Our approach uses Python-based code-style prompts, this requires including tokens in the input to represent the code structures. Figure 8 illustrates various labels formatted in our code-style input alongside their respective guidelines. For very generic guidelines, such as those for the PERSON entity in OntoNotes 5, the code structure accounts for almost half of the input's characters. However, for detailed guidelines, like those for the POINT entity in HarveyNER, the code structure constitutes only a small portion of the input. While there is an overhead of tokens to represent the code structure, when dealing with datasets with a very large number of labels, the primary limitation is fitting the guideline definitions into the model's input, rather than accommodating the Python code structure.

## F HUMAN EFFORT TO BUILD THE PROMPTS

GoLLIE requires formatting the input in a Python-based code representation. We achieve this by filling pre-defined templates for each task (NER, EE, EAE, RE, SF). We will make this predefined set of templates publicly available along with our code. Implementing new datasets only requires defining a list of labels and the guidelines for each label. We reuse the annotation guidelines provided by the dataset authors. Therefore, for most datasets, this process is straightforward and requires very little human effort. For datasets with very large and complex guidelines, such as TACRED, manual summarization of the guidelines was necessary. In any case, the involvement of a domain expert is not required. Additionally, since the inputs are automatically generated using templates, implementing new datasets does not require knowledge of Python coding.

Some datasets did not have publicly available guidelines, either due to semi-automatic generation or because the authors chose not to disclose them. For these specific datasets, human experts were needed to construct the guidelines from examples in the development split. We plan to release our generated guidelines to support future research. Human domain experts are necessary to adapt GoLLIE to new tasks where guidelines or annotations are unavailable. However, this requirement is common to any other Information Extraction (IE) model.

## G DATA-CONTAMINATION STATEMENT

We believe data contamination is a relevant problem that affects the NLP evaluations nowadays, becoming more prevalent with LLMs (Dodge et al., 2021; Magar & Schwartz, 2022; Sainz et al., 2023b;a). Detecting whether a dataset was inside an LLM pre-trained corpora is challenging even with the pre-training data itself. In this paper, unfortunately, we do not have access to the pre-training data used to train Code-LLaMA the backbone LLM of our model. This issue is particularly worrying for us because one big source of contamination is probably GitHub and other code repositories which are also used to upload evaluation benchmarks Dodge et al. (2021). As Code-LLaMA is trained on code, there is a chance for this particular data leakage. However, all of our comparisons were made against our baseline, which has the same backbone LLM as GoLLIE. Even if the results were impacted, **the improvements of our model over the baseline would not be affected by data contamination as both share the same pre-training**. We hope that in the future more transparency will allow to perform safer evaluations.