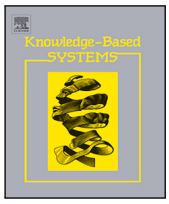
Knowledge-Based Systems 317 (2025) 113386

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys



Document-level event extraction from Italian crime news using minimal data

Giovanni Bonisoli [a], David Vilares [b,*], Federica Rollo [a], Laura Po [a]

[a] University of Modena and Reggio Emilia, "Enzo Ferrari" Engineering Department, Via Vivarelli, 10, Modena, 41125, Italy
[b] Universidade da Coruña, CITIC, Departamento de Ciencias de la Computación y Tecnologías de la Información, Campus de Elviña s/n, 15071, A Coruña, Spain

ARTICLE INFO

Dataset link: https://github.com/federicarollo/Italian-Crime-News, https://github.com/federicarollo/Document-Level-Event-Extraction

Keywords:
Event extraction
Large language models
In-context prompting
Few-shot learning
Prompt tuning
Crime news
Information extraction

ABSTRACT

Event extraction from unstructured text is a critical task in natural language processing, often requiring substantial annotated data. This study presents an approach to document-level event extraction applied to Italian crime news, utilizing large language models (LLMs) with minimal labeled data. Our method leverages zero-shot prompting and in-context learning to effectively extract relevant event information. We address three key challenges: (1) identifying text spans corresponding to event entities, (2) associating related spans dispersed throughout the text with the same entity, and (3) formatting the extracted data into a structured JSON. The findings are promising: LLMs achieve an F1-score of approximately 60% for detecting event-related text spans, demonstrating their potential even in resource-constrained settings. This work represents a significant advancement in utilizing LLMs for tasks traditionally dependent on extensive data, showing that meaningful results are achievable with minimal data annotation. Additionally, the proposed approach outperforms several baselines, confirming its robustness and adaptability to various event extraction scenarios.

1. Introduction

Document-level Event Extraction (DEE) aims at identifying information about an event within a lengthy text [1,2]. This usually refers to any significant occurrence, action, or situation involving specific entities (individuals, organizations, event-specific roles, abstractions like laws or policies, ...) and unfolding over a specific period of time. DEE has several application domains such as real-time detection of infectious disease outbreaks [3], crisis monitoring [4], identification of legal events to assist courts [5], analysis of historical texts [6], and ontology construction and biomedical triggers related to proteins and genes [7].

Event extraction in daily crime news involves sifting through unstructured data to identify and categorize events accurately [8]. Also, it aids in detection of emerging crime trends and threats, enabling risk reduction and community safety. Manual analysis of this vast data is not only time-consuming but also prone to inconsistencies. Thus, automated techniques can offer an advantage by improving the efficiency and consistency of analyzing crime news. However, developing these systems faces challenges, especially in acquiring labeled datasets, as the annotation process requires precise guidelines and specialized expertise. This complexity complicates developing high-quality corpora and complicates model training. Leveraging minimally informed methods for crime news analysis can alleviate these challenges, reducing dependence on annotated datasets while maintaining accuracy in event detection, categorization, and linking. Yet, little work has been done, despite its numerous practical applications, and existing methods mainly rely on Question Answering (QA) models and require labeled data [9,10]. Traditional NER methods often require extensive fine-tuning to achieve high performance. In contrast, our approach shows that LLMs can deliver comparable results with just a handful of in-context learning examples. This capability is particularly valuable in scenarios where access to large annotated datasets is limited. By reducing the reliance on extensive data preparation, our method offers a novel and practical solution that is both efficient and scalable in resource-constrained environments.

Fig. 1 shows an example of DEE from a crime news article, as framed in our work. As we can see, the task poses several challenges: (i) there can be multiple entities for the same label (e.g., victims and authors), (ii) entities can comprise several discontinuous segments of text, and (iii) not all labels are consistently present in news articles.

In our study: (i) we establish a methodology to enable LLMs to accurately extract event related data and provide it in a structured JSON format; (ii) we conduct an extensive evaluation to measure the robustness of various LLMs, exploring zero-shot and in-context learning (ICL), and the influence of presented examples; (iii) we release an effective automated approach to identify key event details – such as "who", "what", and "where" – from Italian crime news, which helps

* Corresponding author.
E-mail addresses: giovanni.bonisoli@unimore.it (G. Bonisoli), david.vilares@udc.es (D. Vilares), federica.rollo@unimore.it (F. Rollo), laura.po@unimore.it (L. Po).

https://doi.org/10.1016/j.knosys.2025.113386
Received 13 September 2024; Received in revised form 5 March 2025; Accepted 18 March 2025
Available online 1 April 2025

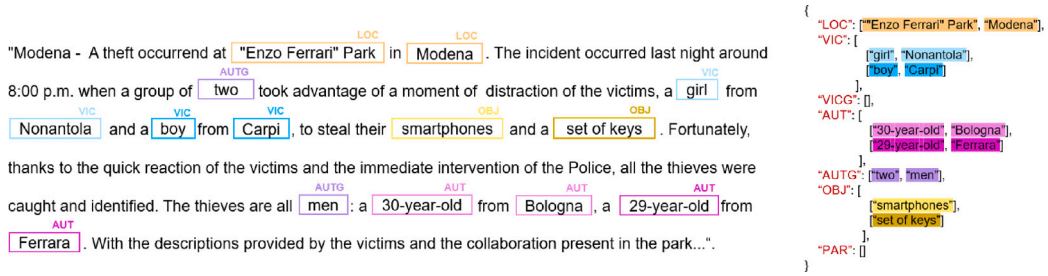


Fig. 1. A DEE example of an Italian crime news article (translated here for clarity). The left part depicts the visual representation of the event extraction, and the right part shows the JSON output we aim to generate in an end-to-end fashion. LOC (location) is the crime location, VIC (victim) is each individual victim, VICG (victim group) is the group of victims, AUT (author) is each individual perpetrator, AUTG (author group) is the group of perpetrators, OBJ (object) refers to the stolen objects, and PAR (party) is the injured party of the crime.

in detecting crime trends. By conducting this study, we aim to address the following research questions:

RQ1 How effective are LLMs in performing document-level event extraction from Italian crime news with low annotated data? Are there performance differences between LLMs?

RQ2 What is the impact of zero-shot prompting versus in-context learning on the accuracy and robustness of event extraction?

RQ3 How do the number, selection, and combination of examples for in-context learning affect the performance and stability of LLMs in event extraction?

RQ4 How are LLMs compared to traditional extractive QA and NER models?

RQ5 What are the specific challenges encountered in identifying and associating event-related text spans in Italian crime news, and how effectively do the LLMs address these challenges?

While we acknowledge the importance of generalization to other domains, our focus in this paper is to evaluate how well LLMs perform on very niche domains, such as crime news. This distinction allows us to prioritize depth over breadth in our analysis.

The remainder of this paper is organized as follows. Section 2 reviews existing methodologies for event extraction, with a focus on studies relevant to the Italian language and crime news. We also discuss the evolution of DEE techniques, including traditional rule-based systems and more recent approaches using LLMs. Our approach for DEE using LLMs is described in Section 3, including the problem formulation, dataset description, and our strategy for conditioning instruction-tuned models. We also describe the annotation schema used to structure the extracted event data. Section 4 presents the experimental setup, including the evaluation metrics and baseline models used for comparison. In Section 5, we discuss the results, comparing the performance of LLMs using zero-shot prompting and in-context learning on Italian crime news. We also explore the impact of example selection in in-context learning and discuss the challenges encountered in identifying and linking event-related spans. The final section summarizes the study's key findings and suggests directions for future research.

2. Related work

Different paradigms have been proposed over the years to address the challenges of DEE [11–13], ranging from traditional rule-based systems to more advanced machine learning models. In the following sections, we provide a detailed review of relevant studies that align closely with our research, focusing on three key areas: (i) event extraction specific to the Italian language, (ii) event extraction within the context of crime news, and (iii) data-driven techniques for DEE.

2.1. Event extraction on Italian documents

Few works have addressed event extraction from Italian documents, many of them facing limited availability of resources. One such effort is the creation of the Italian TimeBank by Caselli et al. [14], which includes a dataset of events, temporal expressions, and relations, along with a list of guidelines and specifications. Viani et al. [15] developed a system based on LSTMs to create morphosyntactically enriched embedding representations of Italian medical reports for event extraction. The system showed that it was possible to identify references to conditions (e.g., names of diseases), tests, and treatments. Relying on a similar approach, Caselli [16] tested a biLSTM-based model for sentence-level event detection and classification on the Italian news articles from the EVENTI corpus. However, their scope differed from ours, as it only aimed to detect verbs and assign them a category from a predefined list. Recently, a large corpus called EventNet-ITA [17] was introduced, featuring over 53k annotated sentences and 200 modeled frames for Italian event parsing. It also presented a supervised multi-label sequence labeling approach. We actually take a different approach, aiming to minimize the need for annotated data and make automated event extraction effective, using LLMs.

2.2. Event extraction for crime detection

The community has also explored applying natural language processing (NLP) techniques for event extraction from crime news. Dasgupta et al. [18] trained a convolutional network on annotated news articles to identify crime-related entities, such as the names of the accused and the victim, the nature of the crime, the geographic location, the law enforcement involved, and any action taken against the accused. A similar approach based on biLSTMs was tested by Bustamante et al. [19] on Peruvian crime-related news for the identification of authors, victims and locations. Mostafazadeh et al. [9] developed a system based on biLSTMs and a crime acts taxonomy for the identification of the crime category (e.g., homicide or kidnapping) and reason (e.g., nationality, gender, religion, ...) from English news articles of the Patch dataset. An ensemble approach was evaluated for sentence-level extraction of entities and actions from Polish criminal documents [20]. This strategy combined lexicon lookup, hand-crafted rules, statistics, and neural networks. Still, the performance was low, highlighting the task's complexity.

2.3. Crime information extraction as NER task

Named Entity Recognition (NER) has been extensively applied to extract structured information from text in the crime domain. For example, Al-Zaidy et al. [21] utilized NER techniques to identify individuals' names from unstructured texts to construct a network of people potentially involved in criminal activities. Subsequent studies developed rule-based NER systems for crime-related tasks, relying on

morphological and syntactic patterns to identify entities associated with criminal activities in newspaper articles [22,23].

More recent advancements have applied machine learning-powered NER techniques in the crime domain. Some studies integrated generic NER models into crime information extraction pipelines, subsequently filtering entities to retain those relevant to criminal activities [24–27].

However, Juez-Hernandez et al. [28] emphasized the significance of domain-specific NER models for tasks demanding high precision and secure data handling, noting their limited portability across domains. This perspective has been adopted in various works, where NER models are specifically trained from scratch for extracting crime-related entities. Some of these efforts rely on classic machine learning methods [20,29–32], while others leverage Transformer-based models, such as BERT and ELECTRA [33,34]. Other studies have shown that pre-trained NER models can achieve significant performance through domain adaptation, [35,36]. Notably, the works of Pérez-Diez et al. [37] and Alshammari et al. [38] are significant, as they adapted Spacy NER models to the medical and crime domains, respectively, achieving high performance with a few hundred training examples.

2.4. Evolution of traditional (supervised) DEE approaches

A traditional approach for DEE systems was based on pipelines, which involved a sequential process divided into distinct subtasks, executed one after the other in a rigid manner. For instance, Yang et al. [1] is a nice representative of this approach. First, they introduced a framework operating in the following stages: they performed sentence-level event extraction using a sequence tagging model. After this, they applied a DEE step that relies on the results from the SEE stage. The DEE step consisted of a key event detection model and a strategy to extract and complete event information at the document level. Zheng et al. [2] used a Transformer encoder to extract sentence and entity representations, followed by a decoding module that created an entity-based directed acyclic graph (EDAG) to sequentially extract multiple event records. Building on this, Xu et al. [39] replaced the Transformer encoder with a GCN-based encoder and used a memory-based tracker to model interactions between different events in the decoding phase. Subsequent approaches aimed to overcome the limitations of EDAG, such as an end-to-end model to predict events in parallel [40] and a framework that exploited sentence-level clues to explain each extracted event [41]. Huang and Jia [42] introduced the concept of sentence community, where each community corresponds to an event. The entities mentioned in a community are potentially participants in the events. The concept was revisited by Zhu et al. [43], whose approach identifies an entity maximal clique composed of pseudo-trigger words and expands other entities that have edges with all pseudo-trigger words in the clique. Recently, other DEE frameworks have leveraged pseudo-trigger extraction to build pruned complete graphs and derive event records directly from these graphs [44,45]. However, these methods require large amounts of annotated data for training and diverge from the direction needed for the domain addressed in this work.

2.5. DEE as a QA task

Several authors have addressed the problem of event extraction through extractive QA [6,46–48]. These approaches are typically based on fine-tuning Transformer-based models like BERT [49] on an annotated dataset. Yet, these datasets are scarce, especially for non-English languages. Also, many available datasets and models only perform single-span QA [50,51], i.e. the system extracts a single, continuous portion of text from a document as the answer. This is sometimes enough when working with short texts. However, when working with lengthy texts, the answer to a question might consist of multiple text portions scattered in the document. To address this, different multi-span QA systems and datasets have been published in the last years [52–54], yet they are mostly designed for English. For Italian, Bonisoli et al. [10] trained an extractive QA system based on SQUAD-it [55] for event extraction on the DICE dataset of Italian crime news, yet the system had no ability to connect spans belonging to the same entity.

2.6. DEE with LLMs

The use of LLMs for document event extraction has gained considerable attention in recent research [56]. In the past, initial attempts were made by Wang et al. [57], who used GPT-3 to annotate datasets, which were then used for training smaller models. Other works followed, exploring GPT's capability for labeling [58–60]. Ma et al. [61], however, criticized LLMs for their inferior performance and higher latency compared to fine-tuned SLMs. Comparing the LLM's performance to that of crowd-workers or non-experts has revealed that LLMs can even surpass human performance in text annotation tasks [62–64], especially in domain-specific scenarios, such as the financial domain [65]. Doosterlinck et al. [66] exploited few-shot in-context learning in adverse drug event extraction, evaluating the ability of OpenAI's LLMs to identify the core information from biomedical papers, i.e., patient information, drugs taken, dosages, and reactions experienced. LLMs were also used as expert annotators for event extraction [67], experimenting with both zero-shot and one-shot event extraction and revealing that the provision of examples improves the outcomes. A comparison on the ACE 2005 dataset [68] between GPT-4, PaLM, and GPT-3.5-Turbo revealed that GPT-4 achieved the highest performance, with an F1-score of 56.7% in one-shot scenarios.

Shiri et al. [69] explored the use of automated event extraction while addressing the issue of hallucination. The task is decomposed into two steps: Event Detection (ED) and Event Argument Extraction (EAE). Additionally, it improves performance by integrating schema-aware retrieval-augmented examples into prompts, extending techniques like Retrieval-Augmented Generation (RAG). Other works focus specifically on Document-level EAE exploring in-context learning [70] and Heuristic-Driven prompting [71].

Document-level event extraction can also be performed using LLMs in a conditional generation framework, where text generation is guided by specific conditions or prompts. For example, Li et al. [72] used an encoder–decoder LLM with an event passage input and an unfilled event template as the condition. Hsu et al. [73] improve this by using prompts that include an output template for semantic guidance and additional weakly-supervised information, like event descriptions and keywords. Peng et al. [74] further developed this approach with a multiple template choice model, incorporating extended event type mining to enhance event extraction. Blair et al. [75] also extended this methods to handle different densities of events within documents.

Although many of these studies focus on LLM performance with English corpora, some studies [57,76] have examined LLMs performance on non-English corpora, finding a significant decline in performance.

3. Methodology

In Section 3.1 we introduce the notation and concepts to formalize the problem. In Section 3.2 describe DICE, an Italian Crime News dataset with its annotation schema, which we will use as the starting point for this work. In Section 3.3 we discuss our efforts to condition instruction-tuned models with the appropriate information to ensure they follow the task guidelines and produce valid outputs.

3.1. Problem formulation

A key part of DEE involves identifying the components that take part in an event, such as participants, time, location, and other relevant attributes or circumstances. Given a text document $d \in D$ such that $d = \{w_1, w_2, \ldots, w_N\}$, where N is the total number of words and each w_i represents the ith word within that document, DEE is the process

Table 1
Schema labels. Header columns: ME = Multi-entity; MS = Multi-span.

Label	Definition	ME	MS
AUT	The perpetrator of a crime	True	True
AUTG	A group of perpetrators	False	True
VIC	The victim of a crime	True	False
VICG	A group of victims	False	True
PAR	An injured party in a crime	False	False
LOC	The location of a crime	False	True
OBJ	Robbed object(s)	True	True

of identifying and structuring relevant occurrences of entities within the text. In a specific context, the DEE is described by the annotation schema, that lists the labels $l \in L$, i.e. the essential components of the event; and the guidelines to annotate entities $E^l = \{E^l_1, E^l_2, \dots, E^l_M\}$ belonging to a label.

An entity E^l_j can be described by one or more non-consecutive *spans* within a document. Let s^l_j be a *span* of the text document, consisting of one or more consecutive words referring to E^l_j. We denote the *linkage set of a given entity* as a list of spans referring to a specific entity with a specific label $\mathcal{L}(E^l_j) = \{s^l_{j1}, s^l_{j2}, \dots, s^l_{jO}\}$. We denote the *set of spans of a given label* as $S^l = \bigcup_j \mathcal{L}(E^l_j)$. This set includes all spans referring to the same label l.

Some labels might consist of a single entity. This happens for the label l = LOC (location) in Fig. 1, which is referred twice through two different spans: 'Enzo Ferrari Park' and 'Modena'. For this label, there will be only one linkage set comprising these two spans, i.e. {'Enzo Ferrari Park', 'Modena'}. Conversely, other labels might consist of multiple entities, such as the label l = OBJ (robbed object), which includes two entities, 'smartphones' and 'set of keys'. Therefore, this label will have two linkage sets: {'smartphones'} and {'set of keys'}.

We want to mark that some entities will be single-span, like {'smartphones'} in Fig. 1, referring to one of the objects that was robbed. Others will be multi-span, such as {'two', 'men'}, which are mentioned in two different parts of the text to refer to the group of individuals who committed the crime.

3.2. DICE dataset and annotation schema

For our work, we rely on DICE, a Dataset of Italian Crime News articles extracted from the newspaper Gazzetta di Modena[1]; which publishes daily news of events of the Modena province, in Italy. It was created in [77] and is available online under the CC BY-NC-SA 4.0 license.[2]

To extract events from these articles, experts defined and agreed upon an annotation schema with sophisticated rules, as described by [10]. The annotation schema considers the journalistic style, a complex writing style requiring attention to several phenomena to extract main event information. These include information redundancy and inconsistency (when an entity is described in a general way, e.g. 'victims', or 'thieves'), varying levels of entity granularity (broad descriptions like 'man', specific ones like 'elderly male', and even finer details such as 'a 67-year-old man') and gradual specification with co-referential elements spanning different sentences.

DICE contains 10,395 crime news articles spanning 13 crime categories. Of these, 406 articles reporting single theft events were manually annotated as the Test Set according to the DICE schema, using the labels listed in Table 1. For our work, a group of domain experts studied the schema guidelines[3] and annotated 200 more theft news articles, bringing the total to 606 annotated articles.[4] After that, we decided to use the following splits for our approach:

1. Example Set (10 news articles): Chosen by a domain expert to represent the entire dataset, this new set has been annotated with expert consensus and includes news articles of varying lengths (ranging from 96 to 655 words), annotation patterns (ranging from 2 to 6 labels annotated out of the 7 labels of the schema), and the number of annotated elements (ranging from 4 to 13 annotated spans). These examples will be combined and fed into the prompt in different ways to explore in-context learning. This approach ensures that our examples provide a comprehensive reflection of the dataset's characteristics.
2. Validation Set (190 news articles): This newly annotated set serves as a platform for initial evaluation. It allows for a detailed examination of LLMs outputs to guide prompt refinement, tuning, and testing various in-context learning configurations using different numbers and combinations of examples.
3. Test Set (406 news articles): The Test Set, released by [10], provides an unbiased evaluation of LLMs' performance on data not used for prompt refinement, allowing demonstrating LLMs' generalization ability.

3.3. Prompt setup

Desired output format. We ask the models to output in JSON format due to its standard nature and adaptability, which allows for easy representation of hierarchical and variable-length data. The requested JSON structure should contain seven key–value pairs, one for each label of the DICE schema. Fig. 1 (right side) showed an example. Each pair has the label name as the key and a value that contains the annotations for the label. The value can be a string if it represents a single span for a single entity, a list of strings if it represents multiple spans for a single entity, or a list of lists of strings if it represents multiple linkage sets for multiple entities given a label (e.g., many victims in the crime news). For each crime news, a variable number of spans per label can be extracted, and sometimes there are no spans for a given label.

Prompt text. We first defined an initial prompt based on the annotation schema. This generic prompt was tested on the Validation Set to evaluate its effectiveness. During this preliminary testing, to improve specificity, we identified the need to include details such as a list of socio-demographic characteristics for victims and authors and practical examples of spans to be annotated or left unannotated. To exemplify this, we show the fragment of the prompt that specifies the details for the label VIC (victim): *VIC, a list of lists of strings. Each list of strings should contain the parts of the text that report information related to a single victim, such as the first name or initials and/or the relevant socio-demographic information, such as age, race, ethnicity, residence, resident/native, gender, occupation. Other characteristics or conditions or roles should not be included (e.g., 'victim', 'owner', 'blonde', 'husband', 'wife', 'son', etc.).*

Due to space reasons, we detail the full prompt (in Italian) in Appendix A.

Prompting single vs multiple labels. As an alternative, we conducted initial tests by creating prompts to annotate labels individually, rather than annotating all the labels of the event at once. However, this single-label approach was not satisfactory, showing a decrease in performances of around 10% in exact match and partial match. This experiments revealed that the combined approach improved the model's discernment. For example, different entities like author and victim were better distinguished when requested together.

[1] https://www.gazzettadimodena.it/
[2] https://github.com/federicarollo/Italian-Crime-News
[3] https://github.com/federicarollo/Italian-Crime-News/blob/main/Annotation%20Guidelines.pdf
[4] The annotation of the 200 news articles will be shared under the CC BY-NC-SA 4.0 license if the paper will be accepted for publication.

4. Experimental setup

4.1. Instruction-tuned models

We selected a few representative instruction-tuned LLMs. The first model is **Llama-2** [78], a family of models pre-trained on thirteen languages, including Italian. We used the two smallest instruction-tuned versions: Llama-2-7b-chat and Llama-2-13b-chat.[5] In what follows, we will call them LL2-7B and LL2-13B.

We also included the successor **Llama3** [79], which retains a similar architecture but offers better multilingual capabilities, enhanced with high-quality instruction tuning data for seven languages, including Italian. We selected Llama-3-8B-Instruct,[6] which we call LL3-8B.

The test also included **LLaMAntino-3-ANITA-8B-Inst-DPO-ITA**[7] [80], referenced from now on as LLT3-8B, which was derived by fine-tuning LL3-8B on Italian datasets.

The second model is **Mistral 7B** [81], a 7-billion parameter language model. In this work, we used two versions of Mistral, both fine-tuned for dialogue: Mistral-7B-Instruct-v0.1[8] and Mistral-7B-Instruct-v0.2.[9] We will abbreviate them to M7B1 and M7B2. The third model is **Mixtral 8x7B** [82], which inherits its architecture from Mistral, with some crucial differences, such as a larger context size (32K tokens) and the feedforward blocks replaced by Mixture-of-Expert layers. The model can handle five languages: English, French, Italian, German, and Spanish. In this work, we used the version fine-tuned for dialogue: Mixtral-8x7B–Instruct,[10] referred to as Mx7B throughout the rest of the paper. Although pre-trained on Italian data, these models are significantly influenced by English, the dominant language in the training data.

That said, we mostly adhered to the default generation hyperparameters from Meta Llama and the Huggingface Transformers libraries, with one exception: the maximum number of generated tokens was set to 1000 to ensure a reasonably large window capable of accommodating a significant number of extracted spans. Given our limited resources, Mx7B was tested with 4-bit quantization. We report the values of the key hyperparameters of the LLMs Appendix B.

4.2. Evaluation metrics

We focus on two aspects: (i) the models' capability to annotate and select spans for specific labels, and (ii) their capacity to group spans referring to the same entity, i.e., their ability to create linkage sets.

4.2.1. Evaluation of labeled spans

To evaluate ground-truth (S^l) against predicted spans ($\hat{S}^l$) for each label, we rely on exact match (EM) and partial match (PM) at span-level, following [54]. We use the microaverage, i.e. calculating metrics globally on all labels. A predicted span is considered an EM if it has the same content (exactly the same words) as a span in the ground-truth set.

4.2.2. Grouping of entities in linkage sets

Let $\mathfrak{L} = (\mathcal{L}^l_1, \ldots, \mathcal{L}^l_M)$ be the sequence of ground-truth linkage sets for a given label l, and $\hat{\mathfrak{L}} = (\hat{\mathcal{L}}^l_1, \ldots, \hat{\mathcal{L}}^l_P)$ the sequence of predicted linkage sets. We carry out an alignment procedure. First, for each pair of sets $(\mathcal{L}^l_i, \hat{\mathcal{L}}^l_j)$ from the *candidate pairs* $\mathfrak{L} \times \hat{\mathfrak{L}}$, we compute the Jaccard Index [83,84], a score in the range $[0, 1]$, where 0 means the sets are completely disjoint, while 1 means the sets are identical. Secondly, (2) we select the pair $(\mathcal{L}^l_i, \hat{\mathcal{L}}^l_j)$ with the highest Jaccard index and if the score is higher than the defined threshold, we add this pair to a list of *confirmed pairs*. Moreover, any pair from the list of candidate pairs that contains either $\mathcal{L}^l_i$ or $\hat{\mathcal{L}}^l_j$ is removed. We then repeat the algorithm at step (2), until there are no candidate pairs with a Jaccard index above the threshold, and thus the list of confirmed pairs cannot be further expanded. The confirmed pairs are the true positives. False negatives are the ground-truth linkage sets that are not part of any confirmed pair. False positives are the predicted linkage sets that are not part of any confirmed pair. Thus, precision, recall, and F1-Score for linkage sets evaluation can be computed.

4.2.3. Dealing with invalid JSON outputs

A model may produce an output that is an invalid JSON, i.e., an output with syntax errors (e.g., missing brackets) and/or structural issues (e.g., the presence of keys different from the seven keys we expect as specified in the prompt). In such cases, several corrections are attempted. If the initial or final curly brace is missing, it is added. If the output contains a partially malformed JSON, the correctly formatted key–value pairs are extracted and parsed to form a new JSON; the keys of any invalid key–value pairs are included with their values set to empty. If an invalid JSON is completely irreparable, the prediction is transformed into a JSON with the structure specified in the prompt, with all fields left empty. This transformation ensures that the number of predicted JSONs is consistent across all models, making their results comparable.

5. Results

5.1. Impact of example selection for crime news in-context learning

This test directly addresses **RQ2, RQ3, RQ5**. The models are tested through k-shot prompting on the extraction task using the Validation Set. All models are set up using the prompt described in Section 3.3, and with k ranging from 0 to 4. We are particularly interested in the impact of examples in in-context learning prompts. Although ICL typically involves providing prompts with examples, it is rare for studies to consider the impact that different examples can have on model performance. Therefore, for each k, different combinations of examples from the Example Set are tested to check the impact of the chosen examples on the performance. Given the vast number of potential combinations and permutations of the samples for k ranging from 0 to 4, it was impractical to evaluate all of them due to computational time and resource limitations. Therefore, we randomly sample subsets of k examples from the example set. This randomness is intentional and aligns with the standard practice in few-shot learning to avoid introducing selection bias. To this end, we conducted experiments with multiple combinations of k examples. For $k \in \{2, 3, 4\}$, the chosen combinations are permuted in various orders to assess also the impact of the order of examples.[11] In total, 205 combinations[12] are tested across the 5 models, resulting in 1025 runs executed on the Validation Set that counts 190 news.

[5] https://ai.meta.com/research/publications/llama-2-open-foundation-and-fine-tuned-chat-models/

[6] https://huggingface.co/meta-llama/Meta-Llama-3-8B-Instruct

[7] https://huggingface.co/swap-uniba/LLaMAntino-3-ANITA-8B-Inst-DPO-ITA

[8] https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.1

[9] https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.2

[10] https://huggingface.co/mistralai/Mixtral-8x7B-Instruct-v0.1

[11] We do not further discuss the order of the examples since the differences were generally small.

[12] 1 combination for 0-shot, 10 (1-shot), 56 (2-shots), 69 (3-shots), and 69 (4-shots).

Table 2
The percentage of outputs requiring each specific correction, calculated based on the total number of valid JSON outputs. The reported values are an average across all tested combinations.

Model	JSON syntax fix	None value removal	Schema correction	Dictionary adjustment	Field value fix
LL2-7B	2.9	0.0	4.7	0.5	65.6
LL2-13B	0.4	0.0	24.9	0.7	57.3
LL3-8B	0.1	0.0	0.5	0.0	4.0
LLT3-8B	0.4	0.0	1.1	0.2	7.6
M7B1	15.8	1.2	27.0	10.8	51.2
M7B2	25.5	0.1	7.9	0.1	33.1
Mx7B	52.3	0.0	3.9	0.2	19.5

5.1.1. Valid JSON output

As explained in Section 4.2.3, all LLM-generated outputs go through post-processing that involves a series of verification steps and adjustments aimed at improving JSON validity according to prompt requirements. When requirements are not satisfied, specific fixes are applied sequentially. Through careful analysis of model outputs, we established this sequence of checks and corrections:

1. **JSON Syntax Fix**. Attempts to properly format the answer as a JSON by correcting missing or extra brackets, removing unnecessary escape characters, and fixing list formatting issues. If the output of this correction is still an invalid JSON, it will be transformed into an empty JSON.
2. **None Value Removal**. Eliminates any None values present in the lists within the JSON.
3. **Schema Correction**. Ensures the JSON follows the expected schema by adding missing required fields (with empty lists as values) and removing any unexpected fields.
4. **Dictionary Adjustment**. Sometimes, LLMs format linkage sets as dictionaries instead of lists, with the extracted spans appearing as their values. Therefore, each dictionary is replaced with a simple list containing its values.
5. **Field Value Fix**. Adjusts list structures based on field requirements by flattening nested lists where only a simple list is needed (AUTG, VICG, PAR, and LOC) and wrapping single strings in sublists where nested lists are required (AUT, VIC and OBJ).

This sequential approach applies corrections one after another, though it cannot guarantee transformation of all invalid JSONs into valid ones.

Table 2 presents, for each correction, the percentage of outputs in which that correction contributed to the transformation of an invalid JSONs into a valid one.

The values clearly highlight that the models requiring the fewest corrections are LL3-8B and LLT3-8B, with corrections affecting less than 8% of their valid outputs. In contrast, models LL2-7B, LL2-13B, and M7B1 exhibit a high incidence of the Field Value Fix correction (over 50%), indicating a strong tendency to incorrectly structure the lists that serve as field values. M7B1 also shows the highest incidence of None Value removal, Schema Correction, and Dictionary adjustment. Additionally, M7B2 and Mx7B demonstrate a high need for the JSON Syntax Fix (just over 50% for Mx7B), along with a significant need for the Field Value Fix (just over 30% for M7B2).

Table 3 presents the percentage of JSON outputs generated by each LLM under zero-shot and few-shot scenarios that could not be transformed into valid JSON and were therefore converted to empty JSONs. For these few-shot scenarios, we report the average across all example combinations. It is evident that the LL2-7B model produces the highest percentage of invalid JSONs in the zero-shot setting. Furthermore, the M7B1 model consistently generates a significant proportion of invalid outputs (approximately 15%) across all k-shot scenarios. In contrast, the LL2-13B, LL3-8B, LLT3-8B, M7B2, and Mx7B models maintain a lower incidence of invalid JSONs, each staying below the 11% threshold, with M7B2 and Mx7B nearing almost no invalid JSONs.

Table 3
Percentage of invalid JSON outputs.

Model	k = 0	k = 1	k = 2	k = 3	k = 4
LL2-7B	54.7	10.3	2.5	3.4	5.5
LL2-13B	8.9	3.8	2.0	1.2	1.9
LL3-8B	**1.0**	3.2	2.1	1.3	0.9
LLT3-8B	8.4	10.2	4.0	1.8	1.3
M7B1	22.1	12.7	15.9	16.6	18.1
M7B2	7.37	4.5	2.2	1.7	1.2
Mx7B	**1.0**	**1.2**	**0.6**	**0.5**	**0.4**

It is important to note that in the zero-shot setup (where no example JSON is provided), LL3-8B and Mx7B generate a negligible number of invalid JSONs. Additionally, when comparing the two LLaMa models, we observe that the model size plays a crucial role in reducing the number of invalid outputs. However, this does not apply to LL3-8B, which, despite being smaller than the largest LLaMa-2 model, achieves an even lower rate of corrupted JSONs. Moreover, despite its adaptation to Italian, LLT3-8B achieves a worse percentage w.r.t. the multilingual model it is based on, LL3-8B, especially for k = 0 and k = 1. Furthermore, there is a significant improvement from the version 1 of Mistral to the newer version in terms of producing valid trees. The trends are also similar for the k-shot approaches, with a clear reduction or a small worsening in the number of corrupted JSONs from k = 0 to k = 1, and less noticeable improvements thereafter for most models, especially after k = 2, when in some cases the number of invalid trees increases, although not substantially in any instance.

A more detailed analysis of the invalid outputs identified one error pattern. The M7B2 model tends to misformat the AUT field when the author names in the news text are presented only as initials (e.g., “M.R” instead of “Mario Rossi”). No other notable error patterns were found.

5.1.2. Labeled spans

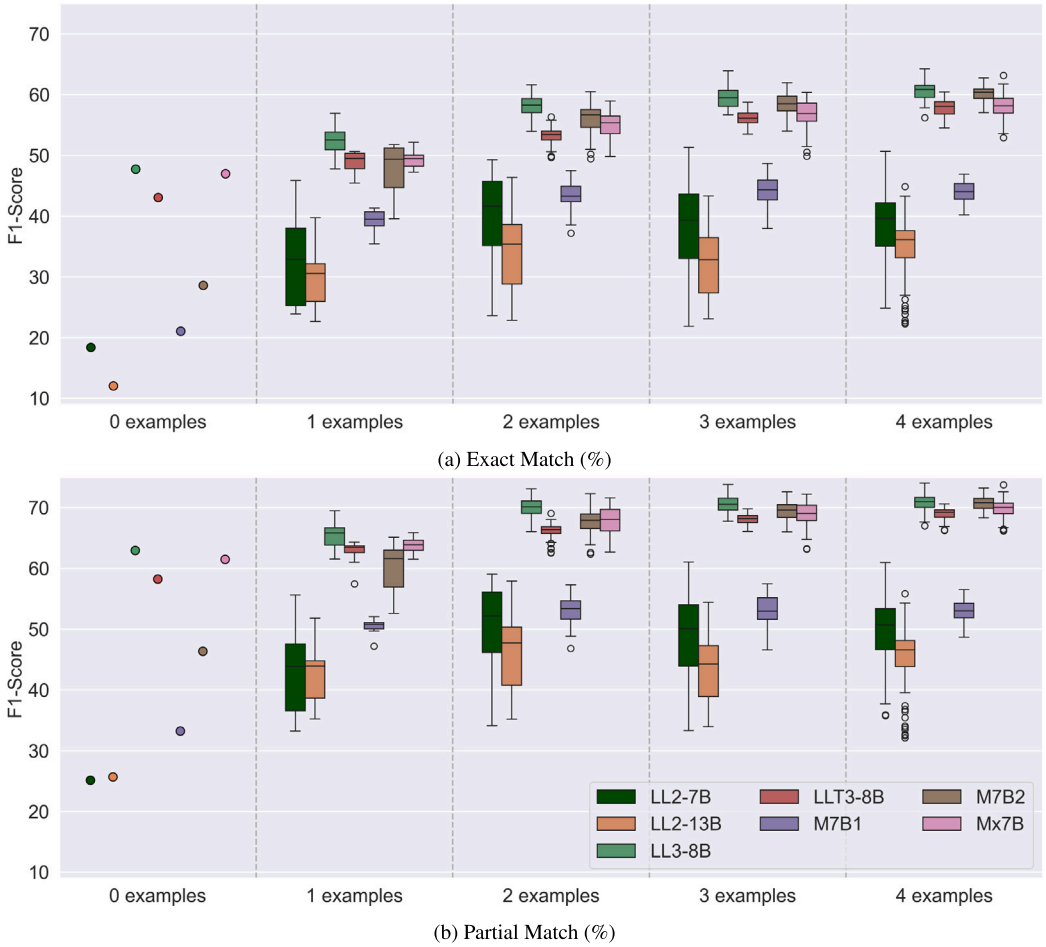
Fig. 2 illustrates EM and PM results on the Validation Set for all LLMs. Each group of boxplots corresponds to a *k*-shot configuration, with each individual boxplot showing the distribution of F1-Score values achieved by each LLM using different combinations of *k* examples. Results for both metrics, EM and PM, follow a similar trend across models, but some models are clearly more robust than others.

For instance, LL2-7B and LL2-13B display the most instability, with F1-scores averaging $36.1_{\pm 6.8}$ for EM and $47.1_{\pm 6.6}$ for PM across various few-shot configurations. In contrast, M7B2 achieves average F1-scores of $57.9_{\pm 3.5}$ for EM and $68.9_{\pm 2.9}$ for PM. This model shows greater variability in the 1-shot configuration compared to the 3-shot and 4-shot configurations, despite the increased in combinations in the latter cases. LL3-8B, LLT3-8B and Mx7B demonstrate the highest consistency across all few-shot configurations. Specifically, LL3-8B achieves an average F1-score of $59.1_{\pm 2.6}$ for EM and $70.2_{\pm 1.9}$ for PM, LLT3-8B achieves $55.5_{\pm 2.8}$ for EM and $67.5_{\pm 2.1}$ for PM, and Mx7B achieves $56.4_{\pm 2.9}$ for EM and $68.6_{\pm 2.3}$ for PM.

Providing prompts with a few labeled examples boosts the results. The performance of LLMs improves as the number of examples in the prompt increases, but approaching a plateau beyond two examples, for all tested models. Still, all models perform worse in zero-shot scenarios compared to few-shot scenarios. Specifically, the best-performing models, LL3-8B, LLT3-8B, M7B2 and Mx7B, achieve their highest performance with 4-shot configurations.

5.1.3. Linkage sets

As discussed in Section 4.2.2, the Jaccard similarity, ranging from 0 to 1, measures how similar a ground-truth linkage set is to a predicted linkage set. To identify confirmed pairs, a threshold for the Jaccard similarity must be established, where only candidate pairs with a similarity above this threshold are considered true positives. A detailed



(a) Exact Match (%)

(b) Partial Match (%)

Fig. 2. Span extraction performance on the Validation Set for all LLMs.

discussion about the evaluation of linkage sets, also on the variation of the threshold chosen, is reported in Appendix C.

Considering that precision, recall, and F1-score should also be evaluated in relation to AJ, we select a threshold of 0.3, which provides a good AJ. This threshold ensures that at least one out of three spans is shared for each confirmed pair. The results presented hereafter will refer to this threshold value.

Fig. 3 shows the performance of the LLMs across different k-shot configurations, with k ranging from 0 to 4. LL2-7B exhibits considerable variability in F1-score across different example set configurations. Noteworthy are the models LL3-8B, LLT3-8B, M7B2 and Mx7B, which perform the best, achieving F1-scores of up to 70%. In few-shot scenarios, the most effective models are LL3-8B, LLT3-8B, M7B2 and Mx7B.

Taking into account both assessment span selection and entity grouping, LL3-8B, LLT3-8B, M7B2 and Mx7B exhibit superior performance across both span selection and entity grouping. On the other hand, LL2-7B, LL2-13B and M7B1 exhibit lower performances, highlighting that performance does not necessarily improve with an increase in k.

5.2. Evaluation on the test set

This test directly addresses **RQ1, RQ2, RQ5**. We study whether the prompt and examples are robust against unseen crime news. Due to computational restrictions, we sample example configurations following a Gaussian distribution, ensuring a representative spread across performance levels. We randomly select models categorized from high, medium, and low performance based on the F1-score for PM. In the end, the chosen example configurations are 56 (including zero-shot): for $k = 1$ all the 10 combinations, while for $k \in \{2, 3, 4\}$ 15 combinations.

We evaluated consistency from two perspectives. First, we check that the averaged F1-score is consistent across the validation and Test Sets for all models. Second, for a given k, we rank the selected subsets of examples based on their F1-scores on both sets and observe if their positions remain highly stable. To elaborate, on the one hand, Fig. 4 reports the averaged F1-scores for different example combinations, comparing the performance on the Validation and Test Sets, and showing that trends remain consistent across both sets both for span and linkage sets F1-scores. On the other hand, we compared the rankings for each subset of examples for a given k in the Validation and Test Sets, and they appear similar. The assessment was done using Spearman's rank correlation coefficient [85], which measures the strength and direction of the monotonic relationship between two ranked variables. Table 4 shows the Spearman's correlation values for the seven LLMs and different k values. Regarding PM, all models except M7B1 exhibit a high the coefficient, above 0.80, indicating a very strong monotonic relationship between the rankings on the two sets. For M7B1 with $k \in \{2, 3, 4\}$, the Spearman coefficient is above 0.60, indicating a strong monotonic relationship. For linkage sets, LL2-7B, LL2-13B and M7B1 exhibit variable coefficient values which do not go beyond 0.63. LL3-8B, LLT3-8B, M7B2 and Mx7B show consistently higher values across all k values, with LL3-8B achieving a very strong correlation for $k \in \{1, 2, 3\}$, LLT3-8B for $k \in \{2, 3\}$, and Mx7B for $k \in \{1, 2\}$.

Our experiments demonstrated that the models' performance remained stable across unseen crime news, as validated by consistent F1-scores and high Spearman correlation values between the validation and test sets (Table 4). This stability suggests that the models

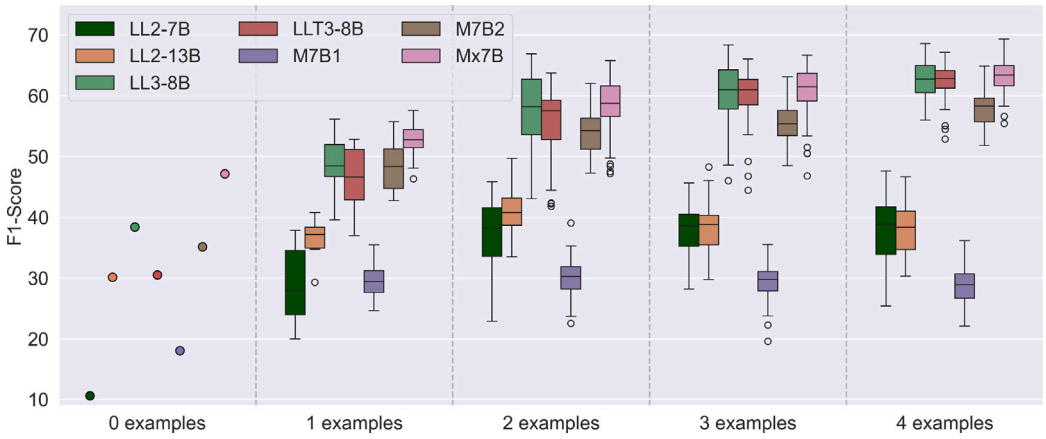
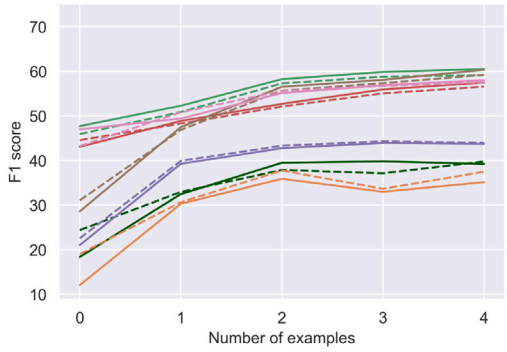
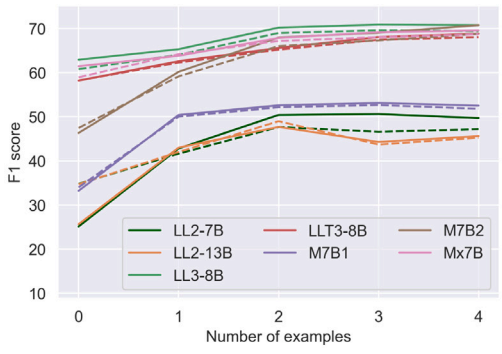


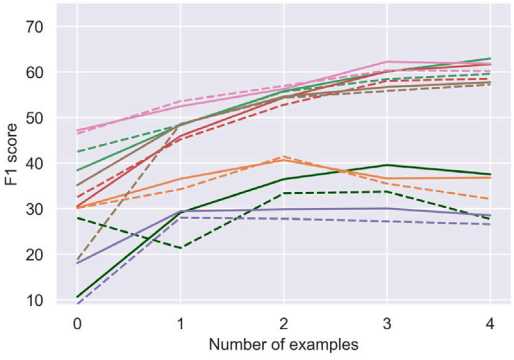
Fig. 3. Performance of the models on linkage sets with threshold 0.3.



(a) Exact Match



(b) Partial Match



(c) Linkage Sets

Fig. 4. Performance trends of different LLMs on the Validation Set (solid line) and Test Set (dashed line).

output crime-news representations effectively, enhancing their generalizability beyond the training data. The results indicate that, despite being trained on a limited number of annotated articles, the models can effectively extract event-related information from previously unseen documents, highlighting their potential applicability to other subdomains of crime news. Recent studies have shown that LLMs can generalize well across domains, even in few-shot learning setups [81] and that models can also act as adaptive annotators, achieving competitive results with minimal supervision [86]. These findings align with our results, suggesting that LLMs could potentially extend their capabilities to other event extraction tasks beyond the crime domain, such as legal proceedings or accident reports.

5.3. Comparison to extractive QA and NER models

This testing directly addresses **RQ4, RQ5**. The final test involved comparing the selected seven LLMs, configured for optimal performance, with other event extraction methods previously applied to the same DICE dataset as described in [10]. This comparison, that in [10] was conducted on a very limited set of news articles (30), is now proposed on the entire Test Set (406 news). The two multi-span QA

Table 4
Spearman correlation between the rankings on the validation set and the test set.

Model	k = 1		k = 2		k = 3		k = 4	
	PM	LS	PM	LS	PM	LS	PM	LS
LL2-7B	**.89**	.38	**.92**	.58	**.90**	.11	.76	.27
LL2-13B	**.95**	.09	**.92**	.62	**.97**	.62	.78	.53
LL3-8B	**.89**	**.90**	**.88**	**.93**	.73	**.82**	.68	.71
LLT3-8B	.64	**.93**	**.87**	**.96**	.75	**.86**	.61	.76
M7B1	.24	.63	**.86**	.44	.78	.26	.70	.39
M7B2	**.99**	.77	**.85**	.61	.70	.70	.67	.49
Mx7B	**.85**	**.94**	**.87**	**.84**	**.84**	.61	.95	.66

Table 5
Comparison between Multi-Span QA, NER models, and LLMs on the Test Set in terms of precision (P), recall (R) and F1-score. LS refers to linkage sets.

Model	EM (%)			PM (%)			LS (%)		
	P	R	F1	P	R	F1	P	R	F1
BERT	35.3	23.6	28.3	38.5	26.0	31.0	–	–	–
ELECTRA	35.6	24.1	28.7	42.1	30.1	35.1	–	–	–
$SpacyNER_{10}$	53.3	44.6	48.6	57.5	46.5	51.4	–	–	–
$SpacyNER_{200}$	**59.6**	61.0	60.3	66.8	65.8	66.3	–	–	–
$LL2\text{-}7B_{k=3}$	46.2	48.7	47.4	55.9	58.1	56.9	38.9	37.4	38.1
$LL2\text{-}13B_{k=2}$	45.8	49.8	47.7	55.6	63.1	59.1	49.5	54.2	51.7
$LL3\text{-}8B_{k=3}$	58.0	**63.2**	**60.5**	66.5	73.4	69.8	**59.2**	64.1	61.5
$LLT3\text{-}8B_{k=4}$	54.1	61.7	57.6	64.4	75.3	69.4	52.2	**75.0**	**61.6**
$M7B1_{k=3}$	48.7	46.7	47.7	56.7	54.1	55.4	40.4	22.2	28.6
$M7B2_{k=3}$	56.8	58.2	57.5	65.4	68.1	66.7	53.9	58.7	56.2
$Mx7B_{k=4}$	58.1	61.1	59.5	**66.9**	**76.1**	**71.2**	58.4	61.5	59.9

models are: an Italian instance of BERT[13] [49] and an Italian instance of ELECTRA[14] [87]. Both models were fine-tuned on a translated version of the MultiSpanQA dataset [54]. To perform event extraction, for each entity, a question in natural language was specified to be given as input to the models along with the news text (e.g., *What was stolen?* for OBJ). It is important to note that the AUT and AUTG labels were evaluated together using the same question (*Who is the thief or criminal?*), since the answer can include both single author and group of authors, similarly to VIC and VICG.

Due to the extensive use of NER systems for extracting crime-related information, two Italian NER models were included in the comparison. Both models were adapted from a pre-trained NER model included in the Italian Spacy pipeline `it_core_news_md`.[15] The first was fine-tuned using the Example Set (10 news articles) and we call it $SpacyNER_{10}$, while the second underwent fine-tuning on both the Example Set and the Validation Set (200 news articles), here named $SpacyNER_{200}$.

In Table 5, the comparison between QA models, NER models and LLMs is presented. This evaluation focuses solely on labeled spans, as QA and NER models do not provide linkage sets. The bold font was used to highlight the best value in each column. LLMs are represented in the table with their optimal configurations (calculated on the Validation Set). The comparison underscores that QA models exhibit notably poor performance, falling behind the adapted NER models by at least 16 points. $SpacyNER_{200}$ significantly improves results w.r.t. $SpacyNER_{10}$, demonstrating that this type of model requires a substantial amount of data for fine-tuning.

Considering the LLMs, M7B2 achieves results comparable to $SpacyNER_{200}$, while LL3-8B, LLM3-8B, and Mx7B remain comparable in EM and improve their scores in PM. LL2-7B, LL2-13B, and M7B1, are less remarkable, as they remain at least 5 points behind M7B2. We report the macro-average scores in Appendix D for completeness.

[13] https://huggingface.co/mrm8488/bert-italian-finedtuned-squadv1-it-alfa
[14] https://huggingface.co/anakin87/electra-italian-xxl-cased-squad-it
[15] https://spacy.io/models/it

Table 6
Statistics and comparison of partial match F1-scores among the different labels on Test Set.

	AUT	AUTG	VIC	VICG	PAR	LOC	OBJ
	Label statistics						
# spans	513	162	180	23	175	686	664
% doc with spans	50	24	27	4	43	92	85
Model	F1-score (%)						
BERT	23.8		51.7	56.6	15.8	19.4	
ELECTRA	23.0		51.3	53.9	30.1	27.7	
$SpacyNER_{10}$	36.2	70.3	67.4	95.4	57.1	56.6	10.5
$SpacyNER_{200}$	71.0	78.7	68.3	94.6	61.6	**52.6**	**49.8**
$LL2\text{-}7B_{k=3}$	57.5	51.4	55.1	91.8	38.7	**54.8**	**56.1**
$LL2\text{-}13B_{k=2}$	64.2	69.6	60.6	92.7	25.4	47.8	63.7
$LL3\text{-}8B_{k=3}$	67.3	70.5	70.3	93.1	59.5	**69.3**	**64.6**
$LLT3\text{-}8B_{k=4}$	71.3	70.3	72.1	95.4	60.8	**65.7**	**61.8**
$M7B1_{k=3}$	41.6	68.6	64.5	88.8	55.9	47.8	42.0
$M7B2_{k=3}$	68.9	66.1	57.4	92.8	71.6	**53.0**	**66.5**
$Mx7B_{k=4}$	74.8	69.0	68.4	95.4	68.7	**60.3**	**69.3**

Table 7
Comparison of inference times of the models with and without quantization. The time was evaluated on the test set using the best configuration of the example set determined on the validation set for each model.

Model	Average inference time per news article (seconds)		
	Without quantization	8-bit quantization	4-bit quantization
LL2-7B	6.7	5.7	6.0
LL2-13B	8.8	11.5	6.5
LL3-8B	2.9	2.9	4.0
LLT3-8B	3.9	3.7	10.0
M7B1	6.4	14.8	4.7
M7B2	6.2	6.2	5.1
Mx7B[a]	n.a.	n.a.	10.0

[a] Mx7B model was tested exclusively with 4-bit quantization due to computational limitations.

Table 6 presents statistics on the different labels in the Test Set, along with the models' performance on these labels in terms of partial match. It is clear that LOC and OBJ have the highest number of spans to annotate, appearing in over 80% of the documents. In contrast, the third label with the most spans, AUT, is found in only 50% of the documents. It is therefore clear that the LOC and OBJ labels are critical for a model to perform well, as the scores for the other labels are mainly inflated by the lack of spans to predict. It is particularly evident with the least frequent label, VICG, where all NER models and LLMs achieve an F1-score above 90% or slightly below, as in the case of M7B1.

Four LLMs outperform $SpacyNER_{200}$ on these two labels, with LL3-8B, LLT3-8B, and Mx7B standing out by exceeding 60% on both LOC and OBJ labels.

5.4. Inference time evaluation

We analyzed the inference times of the seven LLMs used in our experiments. Table 7 reports the average inference time per article. On average, processing a single news article takes between 2.9 and 8.8 s depending on the model (see first column of Table 7). Although this time frame allows for large-scale processing of crime news, we explored quantization strategies to further optimize efficiency. We applied 8-bit quantization using LLM.int8() [88] and 4-bit quantization with QLoRA [89], leveraging the bitsandbytes Python library.[16] Although faster inference times were expected, some models exhibit opposite behavior. For instance, LL2-13B and M7B1 report longer inference times with 8-bit quantization. In contrast, LLT3-8B experiences a substantial

[16] https://huggingface.co/docs/bitsandbytes/index

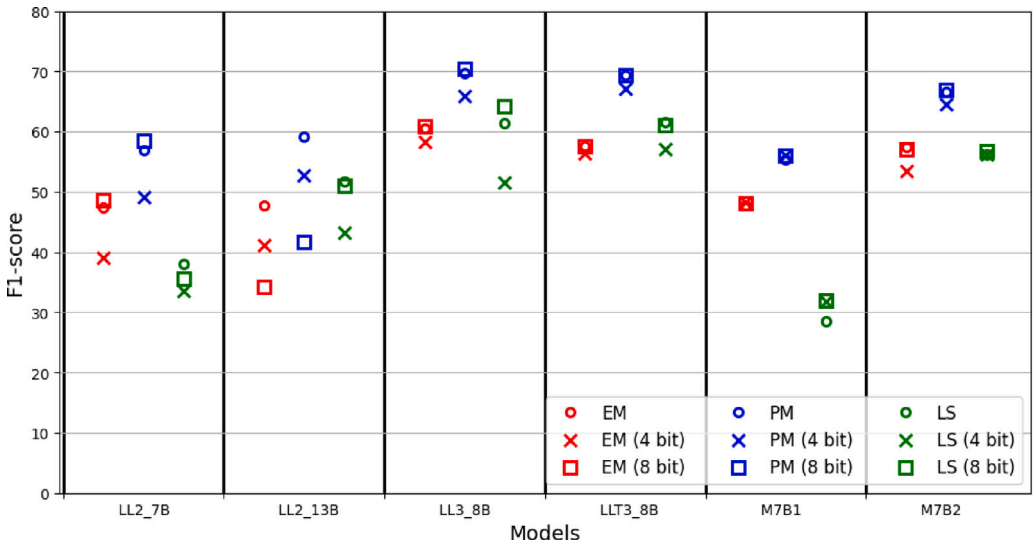


Fig. 5. Comparison of F1-scores on EM, PM and LS when using no quantization (circles), 8-bit (squares) and 4-bit quantization (crosses). For each model, we selected the optimal few-shot configuration and evaluated it using the corresponding quantized version.

performance drop with 4-bit quantization. No notable variations are observed in the other LLMs. These results indicate that quantization does not always lead to faster inference times, as has also been confirmed in the literature [90,91].

Although quantization can improve efficiency, it may come at the cost of reduced model performance due to lower numerical precision. This trade-off is evident in our experimental results for some quantized models.

Fig. 5 presents the F1-scores for exact match, partial match, and linkage set. Results obtained with 4-bit quantization are marked with crosses, those with 8-bit quantization are marked with squares, while circles represent the results without quantization. Overall, F1-scores do not drastically decline with quantization for most models, where only a minor decrease of a few percentage points is observed with 4-bit quantization. For the LL2-7B model, 4-bit quantization leads to a performance decrease. The model most affected by quantization is LL2-13B, which experiences the greatest deterioration in F1-scores.

5.5. Limitations

Computational constraints. Our experiments relied on a limited number of computational resources, specifically four NVIDIA A100 40 GB GPUs, which restricted the number and size of models evaluated. Although we aimed to provide insights into the evolution of models across families (e.g., LLaMa-2 7B versus 13B) and architectures (e.g., Mixtral), this constraint prevented the inclusion of additional models or configurations.

Furthermore, while we explored quantization as a strategy to reduce inference costs, our results indicate that performance and efficiency gains vary across models and are not always consistent. For instance, LL2-7B maintained comparable performance with 8-bit quantization, but degraded with 4-bit quantization. LL2-13B, on the other hand, experienced the most significant performance drop. Additionally, as noted in the literature, the dequantization process can introduce runtime overhead, in some cases offsetting the expected speedup.

These findings highlight a broader challenge — whether the observed performance gains of LLMs, even when optimized through quantization, justify the computational cost compared to smaller, less resource-intensive models or traditional NER techniques. Future work could investigate more adaptive quantization strategies or hybrid approaches to better balance performance and efficiency [91–94].

Selection of models to evaluate on the test set. Related to the previous point, we conducted extensive experiments on our Validation Set to verify how the selection and number of examples influence the models' performance. To the best of our knowledge, this aspect has not been widely explored in the context of LLMs. To ensure the robustness of our results, we also examined how these findings extrapolate and generalize to new, unseen crime news. However, due to limited computational resources, we cannot replicate all experiments with all combinations on the much larger Test Set. Instead, we sampled a few models from the Validation Set according to a normal distribution. We believe this approach is relevant and follows good practices, as it highlights the importance of reporting results on both Validation and Test Sets to ensure generalization, aligning with established research standards in NLP [95].

Domain-specific focus and generalization potential. Our work focuses on a very specific domain: DEE on Italian crime news articles, where each article discusses a single event, i.e. thefts. This narrow scope allowed for a detailed analysis and provided an ideal testbed for our methodology. Although this choice ensures a controlled setting for evaluation, expanding the approach to other domains or multilingual settings could further enhance its applicability. For example, some news articles describe multiple events that our current approach could be adapted to better accommodate. Despite this, the strong correlation in rankings across different k-shot settings suggests that the models can generalize well to new examples, particularly when example selection is optimized. Future research should explore cross-domain evaluations to better understand the limits of generalizability. Testing the approach on datasets covering diverse event types (e.g., natural disasters, political events) could provide further insights into the models' adaptability to varying linguistic structures and event distributions.

In-context learning limitations. The reliance on in-context learning introduces inherent challenges, including sensitivity to prompt design and example selection. Although we conducted extensive validation experiments to understand these effects, the approach may not consistently perform well across unseen datasets or significantly different data distributions. Moreover, in-context learning can be less robust than domain-specific training methods when handling noisy, incomplete, or ambiguous data, which is common in real-world scenarios. In our paper, we rely on manual select of examples done by a domain expert. Some strategies aim to select examples that are semantically similar to the test sample as demonstrations [96–99], while other works explore the combination of diversity and similarity [100,101].

However, the optimal strategy for selecting demonstration examples may be task-dependent, and it remains a common practice to select examples randomly.

6. Conclusion

We studied LLMs' ability to extract events from Italian crime news using minimal annotated information, through zero-shot prompting and few-shot in-context learning. The paper explored various aspects of event extraction relevant for the domain at hand, including identifying event entities, linking related information, and formatting data for downstream applications. We tested various LLMs and observed interesting trends. All models benefited from seeing examples, but their effectiveness showed clear diminishing returns after only two examples. Still, in a fixed few-shot setting, the specific examples mattered. Thus, although selecting optimal samples was beyond the scope of this paper, it is a topic we believe is worth studying as future work. Overall, Mistral 7B v2 and Mixtral 7B were the most robust models among the tested ones, while Llama-2 7B exhibited the least robustness. This superiority was evident in both their ability to select the correct textual spans for events and to group related entities. Furthermore, the tests conducted in this paper have allowed us to address each of the research questions outlined in Section 1:

RQ1. *How effective are LLMs in performing document-level event extraction from Italian crime news with low annotated data? Are there performance differences between LLMs?* LLMs, particularly through zero-shot prompting and in-context learning, achieved meaningful results with minimal labeled data. While zero-shot outputs were competitive, in-context learning provided better results, demonstrating that LLMs are effective for this task even with limited data. LL3-8B, LLT3-8B, M7B2 and Mx7B are consistently among the best performers. These models achieve the highest F1-scores and demonstrate the greatest robustness across different few-shot configurations. Models like LL2-7B and LL2-13B display higher variability and instability in their performance. They tend to have lower average F1-scores and greater fluctuations across different few-shot example configurations compared to the four best LLMs.

RQ2. *What is the impact of zero-shot prompting versus in-context learning on the accuracy and robustness of event extraction?* All models perform worse in zero-shot scenarios compared to few-shot scenarios, highlighting the importance of having at least some annotated data. The study highlighted that in-context learning's performance varied significantly based on the examples shown, indicating that carefully selected examples can enhance the accuracy and robustness of the results.

RQ3. *How do the number, selection, and combination of examples for in-context learning affect the performance and stability of LLMs in event extraction?* The performance of in-context learning models varied significantly with different sets of examples. This suggests that the quality and relevance of examples are critical for optimizing the model's performance. The evaluation showed stable trends across Validation and Test Sets, and the Spearman's rank correlation was used to assess the stability of rankings across different example subsets, confirming the impact of example selection on model stability.

RQ4. *How are LLMs compared to traditional extractive QA and NER models?* LLMs outperform traditional extractive QA models like Italian BERT and ELECTRA when evaluated on the same dataset. Specifically, models such as LL3-8B, LLT3-8B, M7B2 and Mx7B achieve significantly higher F1-scores compared to these QA models. Among the NER models, SpacyNER_{200} delivers results comparable to the top-performing LLMs. However, it is important to note that achieving this required domain adaptation using 200 news articles. In contrast, the LLMs achieved high scores without any training, relying solely on ICL. This highlights their greater effectiveness in scenarios with minimal annotated data.

RQ5. *What are the specific challenges encountered in identifying and associating event-related text spans in Italian crime news, and how effectively do the LLMs address these challenges?* The primary challenges included identifying text spans corresponding to event entities, associating related spans to the same entity, and formatting the extracted data into a structured JSON format. Despite these challenges, LLMs with zero-shot and in-context learning effectively addressed them, achieving an F1-score of around 60% for exact span detection. LL2-7B produces the highest percentage of invalid JSONs in zero-shot settings, while LL3-8B, LLT3-8B, M7B2 and Mx7B maintain a lower incidence of invalid outputs across different k-shot scenarios.

Future work will focus on assessing the generalization of the methodology across various domains and developing a technique for extracting event-related data from multi-event documents.

CRediT authorship contribution statement

Giovanni Bonisoli: Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **David Vilares:** Writing – review & editing, Validation, Supervision, Methodology, Conceptualization. **Federica Rollo:** Writing – review & editing, Visualization, Validation, Methodology, Formal analysis, Data curation, Conceptualization. **Laura Po:** Writing – review & editing, Validation, Supervision, Methodology, Conceptualization.

Declaration of generative AI and AI-assisted technologies in the writing process

During the preparation of this work, the authors used ChatGPT in order to improve the readability and language of the manuscript. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the content of the published article.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work has received support by Grant GAP (PID2022-139308OA-I00) funded by MCIN/AEI/10.13039/501100011033/ and by ERDF, EU; by Grant SCANNER-UDC (PID2020-113230RB-C21) funded by MICIU/AEI/10.13039/501100011033/; by Xunta de Galicia (ED431C 2024/02); and by Centro de Investigación de Galicia "CITIC", funded by the Xunta de Galicia through the collaboration agreement between the Consellería de Cultura, Educación, Formación Profesional e Universidades and the Galician universities for the reinforcement of the research centers of the Galician University System (CIGUS); by Mobility of higher education students and staff supported by internal policy funds (2022-1-IT02-KA131-HED-000064316) funded by EU through the Erasmus+ Programme and by UNIMORE; by Ministry for Digital Transformation and Civil Service and 'Next-GenerationEU'/PRTR under Grant TSI-100925-2023-1.

Appendix A. Prompts

The prompt we used is shown in Fig. A.6.

Appendix B. LLM inference hyperparameters

Table B.8 reports the list of hyperparameters used for the inference phase to generate responses. The first column lists the parameters used with the LLaMA models: LL2-7B, LL2-13B, LL3-8B, and LLT3-8B. The second column pertains to the Mistral models: M7B1, M7B2, and Mx7B.

```
Quando ricevi in input il testo di una notizia di furto, devi restituire un JSON
strutturato con i seguenti campi:

• AUT, una lista di liste di stringhe. Ogni lista di stringhe deve contenere le parti
del testo riportanti le informazioni relative a un singolo autore, quali il nome
proprio o le iniziali e/o età, razza, etnia, residenza, abitante/nativo, sesso,
occupazione, status giuridico (es. "incensurato", "pregiudicato", "gravato da
precedenti"). Non vanno incluse altre caratteristiche o condizioni o ruoli (es.
"malvivente", "ladro", "biondo", "marito", "moglie", "figlio", "ignoti" ecc).

• AUTG, una lista di stringhe. Qui vanno inserite le parti di testo contenenti, se
presenti, riferimenti all'intero gruppo di autori, nel caso gli autori del furto siano
più di uno. Le informazioni da estrarre sono quelle socio-demografiche di riferimento
quali età, razza, etnia, residenza, abitante/nativo, sesso, occupazione. Non includere
termini generici ("autori", "ladri", "malviventi", "ignoti", ecc).

• OBJ, una lista di liste di stringhe. Ogni lista deve contente la parte di testo che
menziona un oggetto rubato e quella che, se presente, ne specifica la quantità. Non
vanno inclusi oggetti o immobili danneggiati, né termini generici come "refurtiva",
"bottino", "oggetti", "possedimenti", ecc.

• VIC, una una lista di liste di stringhe. Ogni lista di stringhe deve contenere le
parti del testo riportanti le informazioni relative a una singola vittima, quali il
nome proprio o le iniziali e/o le informazioni socio-demografiche di riferimento, quali
età, razza, etnia, residenza, abitante/nativo, sesso, occupazione. Non vanno incluse
altre caratteristiche o condizioni o ruoli (es. "vittima", "proprietario", "biondo",
"marito", "moglie", "figlio", ecc).

• VICG, una lista di stringhe. Qui vanno inserite le parti di testo contenenti, se
presenti, riferimenti all'intero gruppo di autori, nel caso le vittime del furto siano
più di una. Le informazioni da estrarre sono quelle socio-demografiche di riferimento
quali età, razza, etnia, residenza, abitante/nativo, sesso, occupazione. Non includere
termini generici ("vittime", "proprietari", ecc).

• PAR, lista di stringhe riferite a una parte lesa dal furto che non sia una persona
fisica, ma un'attività commerciale (negozi, aziende, supermercati, ecc), un ente
pubblico (comune, provincia, scuola, ecc) o un'associazione. La lista deve contenere la
ragione sociale e la tipologia di questa entità. Vanno esclusi termini generici come
"attività commerciale" o "impresa".

• LOC, una lista di stringhe contentente le parti di testo riferite al luogo del
delitto: città, zona cittadina (periferia o centro), via, numero civico, tipo di
struttura ("abitazione", "casa", "appartamento", "condominio"). Nel caso il furto venga
commesso a danno di un'attività commerciale, va incluso anche il nome o tipo di tale
attività che poi sarà presente anche nel campo "PAR".
```

Fig. A.6. Final version of the prompt.

Table B.8
Hyperparameter list for LLaMA and Mistral models.

Hyperparameter	LLaMa models	Mistral models
Max_new_token	1000	1000
Top_p	0.9	1.0
Top_k	50	50
Temperature	0.6	1.0

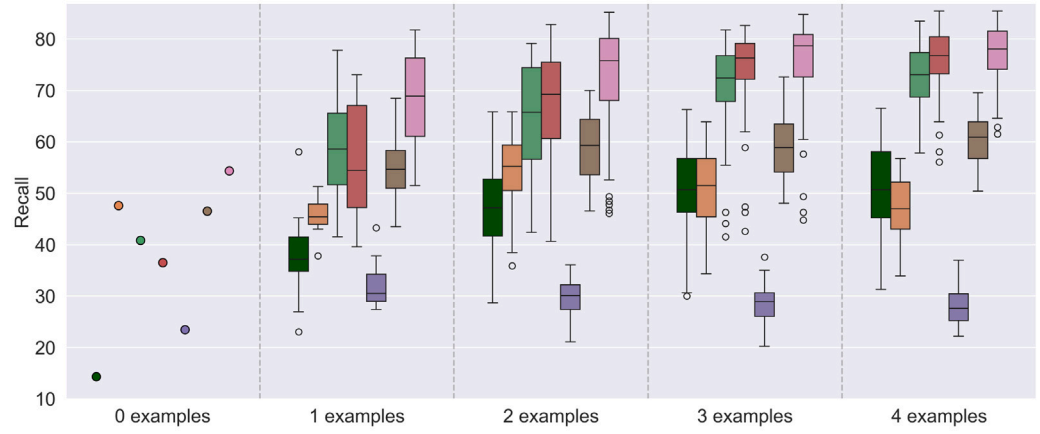
Appendix C. Grouping of entities in linkage sets

Table C.9 explores the impact of varying the threshold on the accuracy and quality of the linkage sets. As the threshold increases, the number of confirmed pairs (i.e., true positives) decreases, while false positives (predicted linkage sets not part of any confirmed pair) and false negatives (ground-truth linkage sets not part of any confirmed pair) increase. This leads to very low precision, recall, and F1-score values even for a threshold of 0.5. No results are reported for values above 0.5, as they would further reduce precision and recall. As expected, with such high and restrictive threshold, the Average Jaccard Similarity (AJ) is very high. On the other hand, with very low thresholds, the number of true positives is high (reaching the maximum possible with no threshold), and both false negatives and false positives decrease accordingly. As a result, precision, recall, and F1-score achieve good values at thresholds of 0.1 and 0.3.

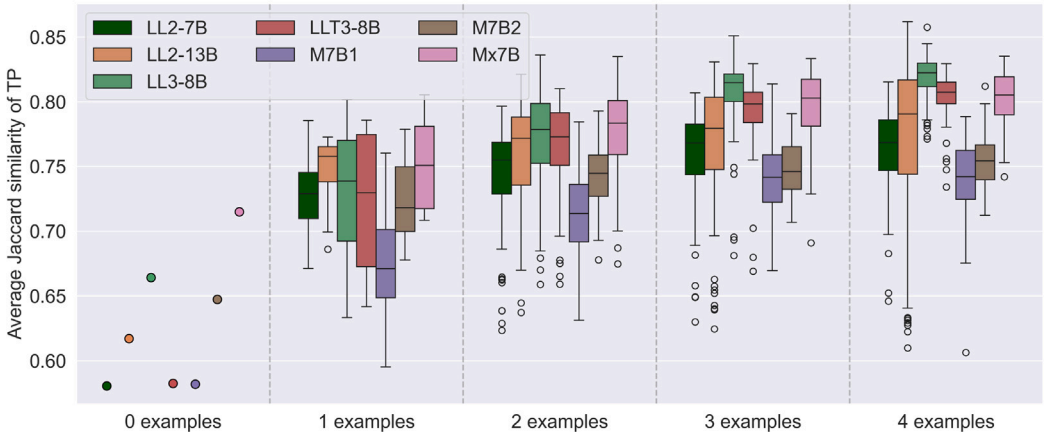
For a threshold of 0.3, Fig. C.7 displays the percentage of ground-truth paired linkage sets (recall) and the average Jaccard index for these paired sets. Each group of boxplots corresponds to a k-shot configuration, with each individual boxplot showing the distribution of recall and Jaccard similarity values achieved by each LLM using different combinations of k examples. The LLMs display significant variability across different few-shot configurations, underscoring the impact of example combinations on the effectiveness of grouping linkage sets. In Fig. C.7(a), it is evident that the most effective models in few-shot scenarios are LL3-8B, LLM3-8B and Mx7B. For $k \geq 2$, the average number of paired linkage sets remains relatively stable. In Fig. C.7(b), it is clear that for all models, the average Jaccard Similarity increases as the number of provided examples grows. This suggests that providing more examples enhances the models' ability to accurately group spans referring to the same entity.

Appendix D. Macro-average performance

Table D.10 shows the results obtained on the Test Set using the macro-average across the different labels. Comparing these results with the micro-average scores, we can notice that trends remain consistent.



(a)



(b)

Fig. C.7. Recall (a) and Average Jaccard Similarity of true positives (b) for linkage sets with threshold 0.3.

Table D.10
Macro-average performance on the Test Set, each with its best configuration.

Model	EM (%)			PM (%)			LS (%)		
	P	R	F1	P	R	F1	P	R	F1
BERT	35.5	28.1	30.8	38.6	30.5	33.5	–	–	–
ELECTRA	36.2	28.0	31.1	42.4	33.7	37.2	–	–	–
$SpacyNER_{10}$	56.8	50.3	52.8	59.1	51.8	54.8	–	–	–
$SpacyNER_{200}$	63.1	63.8	63.2	69.0	67.9	68.2	–	–	–
$LL2\text{-}7B_{k=3}$	50.9	50.7	50.2	58.1	58.5	57.9	33.7	35.3	33.6
$LL2\text{-}13B_{k=2}$	51.4	51.5	51.1	59.1	62.8	60.6	42.2	44.9	43.3
$LL3\text{-}8B_{k=3}$	**63.6**	**64.1**	**63.4**	**69.6**	72.5	70.6	**58.3**	47.0	51.1
$LLT3\text{-}8B_{k=4}$	60.7	64.0	61.9	68.4	75.1	71.0	49.9	**69.3**	57.8
$M7B1_{k=3}$	54.3	51.3	52.5	60.2	57.3	58.4	34.9	19.6	24.6
$M7B2_{k=3}$	60.1	60.5	60.2	67.2	69.3	68.1	49.7	58.7	52.5
$Mx7B_{k=4}$	61.8	63.7	62.6	69.1	**76.3**	**72.3**	55.3	64.7	**58.7**

Data availability

The DICE dataset is publicly available at https://github.com/federicarollo/Italian-Crime-News. The Example Set, Validation Set, Test Set, the prompt used and the annotations generated by the seven LLMs are available at https://github.com/federicarollo/Document-Level-Event-Extraction.

References

[1] Hang Yang, Yubo Chen, Kang Liu, Yang Xiao, Jun Zhao, DCFEE: A document-level Chinese financial event extraction system based on automatically labeled training data, in: Fei Liu, Thamar Solorio (Eds.), Proceedings of ACL 2018, System Demonstrations, Association for Computational Linguistics, Melbourne, Australia, 2018, pp. 50–55.

[2] Shun Zheng, Wei Cao, Wei Xu, Jiang Bian, Doc2EDAG: An end-to-end document-level framework for Chinese financial event extraction, in: Kentaro Inui, Jing Jiang, Vincent Ng, Xiaojun Wan (Eds.), Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP, Association for Computational Linguistics, Hong Kong, China, 2019, pp. 337–346.

[3] Ralph Grishman, Silja Huttunen, Roman Yangarber, Real-time event extraction for infectious disease outbreaks, in: Proceedings of Human Language Technology Conference, HLT, 2002, pp. 366–369.

Table C.9
Evaluation of LLMs on the Validation Set for grouping entities into linkage sets as a function of threshold variation. The results are in terms of precision (P), recall (R), f1-score (F1) and Average Jaccard Index of true positives (AJ), expressed in percentage. The metrics are reported for each number of shots, which is denoted by k. In the case of $k \geq 1$, the average is reported across all example combinations.

Model	Threshold	k = 0				k = 1				k = 2				k = 3				k = 4			
		P	R	F1	AJ	P	R	F1	AJ	P	R	F1	AJ	P	R	F1	AJ	P	R	F1	AJ
LL2-7B	n.d.	18	30	22	30	42	65	50	44	50	74	58	50	48	77	58	51	46	77	57	52
	0.1	11	18	13	50	28	44	34	66	37	55	44	67	36	58	43	69	35	58	43	70
	0.3	8	14	11	58	24	38	29	73	31	47	37	74	31	51	38	76	31	51	38	76
	0.5	3	5	4	95	15	23	18	93	19	30	23	94	20	33	24	95	20	33	24	95
LL2-13B	n.d.	37	80	50	41	48	72	57	50	48	78	59	55	46	75	57	54	47	69	56	54
	0.1	29	63	40	52	36	53	42	67	38	61	46	70	35	57	44	70	36	52	43	72
	0.3	22	48	30	62	31	46	37	74	33	54	41	76	31	51	38	76	32	47	38	77
	0.5	10	21	13	85	20	29	23	92	22	36	27	93	21	35	26	92	22	32	26	93
LL3-8B	n.d.	51	58	54	53	49	69	57	65	59	72	64	72	58	75	65	77	59	76	66	79
	0.1	50	57	53	53	49	69	57	65	59	71	64	72	58	75	65	77	59	76	66	79
	0.3	36	41	38	66	41	59	48	73	53	65	58	77	55	70	61	80	56	72	63	82
	0.5	21	24	22	84	26	38	31	89	37	45	40	92	40	52	45	93	42	55	48	94
LLT3-8B	n.d.	41	57	47	45	47	66	55	64	53	74	62	71	55	79	65	75	56	80	66	77
	0.1	39	55	46	46	46	66	54	65	53	74	62	71	55	79	65	75	56	80	66	77
	0.3	26	36	30	58	39	56	46	72	48	67	57	76	51	74	60	79	53	76	62	80
	0.5	11	16	13	80	24	36	29	89	32	46	38	92	36	53	43	93	39	56	46	93
M7B1	n.d.	35	57	44	28	48	56	51	42	52	49	50	46	50	45	47	49	47	45	46	49
	0.1	22	35	27	46	34	40	37	58	38	36	37	63	37	33	35	66	34	33	33	67
	0.3	15	23	18	58	27	32	29	67	31	29	30	71	31	28	29	74	30	28	29	74
	0.5	6	9	7	86	14	17	15	90	18	17	18	92	19	17	18	93	18	17	18	94
M7B2	n.d.	44	73	56	46	59	75	66	56	66	77	71	59	68	76	71	60	71	76	73	62
	0.1	38	62	47	54	51	65	57	65	57	67	62	68	59	67	63	69	62	67	64	70
	0.3	28	47	35	65	43	55	49	72	50	59	54	74	53	59	55	75	56	60	58	75
	0.5	14	22	17	89	27	34	30	91	32	38	35	91	35	39	37	91	38	41	37	91
Mx7B	n.d.	59	77	67	54	55	86	66	62	60	87	71	66	61	89	72	69	63	76	90	70
	0.1	52	67	58	62	49	76	59	70	54	79	63	73	55	80	65	76	57	81	66	78
	0.3	42	54	47	72	43	68	52	75	50	73	58	77	52	76	61	80	54	77	63	80
	0.5	26	33	29	89	29	46	35	91	34	51	41	92	37	55	44	93	40	57	47	93

[4] Hristo Tanev, Jakub Piskorski, Martin Atkinson, Real-time news event extraction for global crisis monitoring, in: Natural Language and Information Systems: 13th International Conference on Applications of Natural Language To Information Systems, NLDB 2008 London, UK, June 24-27, 2008 Proceedings 13, Springer, 2008, pp. 207–218.

[5] Shirong Shen, Guilin Qi, Zhen Li, Sheng Bi, Lusheng Wang, Hierarchical Chinese legal event extraction via pedal attention mechanism, in: Donia Scott, Nuria Bel, Chengqing Zong (Eds.), Proceedings of the 28th International Conference on Computational Linguistics, International Committee on Computational Linguistics, Barcelona, Spain (Online), 2020, pp. 100–113.

[6] Nadav Borenstein, Natália da Silva Perez, Isabelle Augenstein, Multilingual event extraction from historical newspaper adverts, in: Anna Rogers, Jordan Boyd-Graber, Naoaki Okazaki (Eds.), Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Toronto, Canada, 2023, pp. 10304–10325.

[7] Diya Li, Lifu Huang, Heng Ji, Jiawei Han, Biomedical event extraction based on knowledge-driven tree-LSTM, in: Jill Burstein, Christy Doran, 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), Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 1421–1430.

[8] Federica Rollo, Laura Po, Crime event localization and deduplication, in: Jeff Z. Pan, Valentina A.M. Tamma, Claudia d'Amato, Krzysztof Janowicz, Bo Fu, Axel Polleres, Oshani Seneviratne, Lalana Kagal (Eds.), The Semantic Web - ISWC 2020 - 19th International Semantic Web Conference, Athens, Greece, November 2-6, 2020, Proceedings, Part II, in: Lecture Notes in Computer Science, vol. 12507, Springer, 2020, pp. 361–377.

[9] Aida Mostafazadeh Davani, Leigh Yeh, Mohammad Atari, Brendan Kennedy, Gwenyth Portillo Wightman, Elaine Gonzalez, Natalie Delong, Rhea Bhatia, Arineh Mirinjian, Xiang Ren, Morteza Dehghani, Reporting the unreported: Event extraction for analyzing the local representation of hate crimes, in: Kentaro Inui, Jing Jiang, Vincent Ng, Xiaojun Wan (Eds.), Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP, Association for Computational Linguistics, Hong Kong, China, 2019, pp. 5753–5757.

[10] Giovanni Bonisoli, Maria Pia Di Buono, Laura Po, Federica Rollo, DICE: a dataset of Italian crime event news, in: Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '23, Association for Computing Machinery, New York, NY, USA, 2023, pp. 2985–2995.

[11] Giacomo Frisoni, Gianluca Moro, Antonella Carbonaro, A survey on event extraction for natural language understanding: Riding the biomedical literature wave, IEEE Access 9 (2021) 160721–160757, Cited by: 24; All Open Access, Gold Open Access, Green Open Access.

[12] Wei Xiang, Bang Wang, A survey of event extraction from text, IEEE Access 7 (2019) 173111–173137, Cited by: 102; All Open Access, Gold Open Access.

[13] Yang Yang, Zhilei Wu, Yuexiang Yang, Shuangshuang Lian, Fengjie Guo, Zhiwei Wang, A survey of information extraction based on deep learning, Appl. Sci. (Switz.) 12 (19) (2022) Cited by: 15; All Open Access, Gold Open Access.

[14] Tommaso Caselli, Valentina Bartalesi Lenzi, Rachele Sprugnoli, Emanuele Pianta, Irina Prodanof, Annotating events, temporal expressions and relations in Italian: the it-timeml experience for the ita-TimeBank, in: Nancy Ide, Adam Meyers, Sameer Pradhan, Katrin Tomanek (Eds.), Proceedings of the 5th Linguistic Annotation Workshop, Association for Computational Linguistics, Portland, Oregon, USA, 2011, pp. 143–151.

[15] Natalia Viani, Timothy A. Miller, Dmitriy Dligach, Steven Bethard, Carlo Napolitano, Silvia G. Priori, Riccardo Bellazzi, Lucia Sacchi, Guergana K. Savova, Recurrent neural network architectures for event extraction from Italian medical reports, in: Annette ten Teije, Christian Popow, John H. Holmes, Lucia Sacchi (Eds.), Artificial Intelligence in Medicine, Springer International Publishing, Cham, 2017, pp. 198–202.

[16] Tommaso Caselli, Italian event detection goes deep learning, in: Elena Cabrio, Alessandro Mazzei, Fabio Tamburini (Eds.), Proceedings of the Fifth Italian Conference on Computational Linguistics (CLiC-It 2018), Torino, Italy, December 10-12, 2018, in: CEUR Workshop Proceedings, vol. 2253, CEUR-WS.org, 2018.

[17] Marco Rovera, EventNet-ITA: Italian frame parsing for events, in: Yuri Bizzoni, Stefania Degaetano-Ortlieb, Anna Kazantseva, Stan Szpakowicz (Eds.), Proceedings of the 8th Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature, LaTeCH-CLfL 2024, Association for Computational Linguistics, St. Julians, Malta, 2024, pp. 77–90.

[18] Tirthankar Dasgupta, Rupsa Saha, Lipika Dey, Abir Naskar, Automatic extraction of causal relations from text using linguistically informed deep neural networks, in: Kazunori Komatani, Diane Litman, Kai Yu, Alex Papangelis, Lawrence Cavedon, Mikio Nakano (Eds.), Proceedings of the 19th Annual SIGDial Meeting on Discourse and Dialogue, Association for Computational Linguistics, Melbourne, Australia, 2018, pp. 306–316.

[19] Gina Bustamante, Arturo Oncevay, CSI peru news: finding the culprit, victim and location in news articles, in: Amittai Axelrod, Diyi Yang, Rossana Cunha,

Samira Shaikh, Zeerak Waseem (Eds.), Proceedings of the 2019 Workshop on Widening NLP, Association for Computational Linguistics, Florence, Italy, 2019, pp. 174–176.

[20] Paweł Skórzewski, Mikołaj Pieniowski, Grazyna Demenko, Named entity recognition to detect criminal texts on the web, in: Nicoletta Calzolari, Frédéric Béchet, Philippe Blache, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Jan Odijk, Stelios Piperidis (Eds.), Proceedings of the Thirteenth Language Resources and Evaluation Conference, European Language Resources Association, Marseille, France, 2022, pp. 6223–6231.

[21] Rabeah Al-Zaidy, Benjamin C.M. Fung, Amr M. Youssef, Francis Fortin, Mining criminal networks from unstructured text documents, Digit. Investig. 8 (3) (2012) 147–160.

[22] Mona Asharef, Nazlia Omar, Mohammed Albared, Z MINHUI, W WEIMING, Z JINGJING, Z ALI, Wu Fei, Z DONGSONG, FU YU, Arabic named entity recognition in crime documents, J. Theor. Appl. Inf. Technol. 44 (2012).

[23] Nadhira Annisa Dwi Dharviyanti, Nori Wilantika, Rule-based NER for crime information extraction through online news site, in: 2024 International Conference on Information Technology Research and Innovation, ICITRI, 2024, pp. 99–104.

[24] Rexy Arulanandam, Bastin Tony Roy Savarimuthu, Maryam A. Purvis, Extracting crime information from online newspaper articles, in: Proceedings of the Second Australasian Web Conference - Volume 155, AWC '14, Australian Computer Society, Inc., AUS, 2014, pp. 31–38.

[25] Vasu Sharma, Rajat Kulshreshtha, Puneet Singh, Nishant Agrawal, Akshay Kumar, Analyzing newspaper crime reports for identification of safe transit paths, in: Diana Inkpen, Smaranda Muresan, Shibamouli Lahiri, Karen Mazidi, Alisa Zhila (Eds.), Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop, Association for Computational Linguistics, Denver, Colorado, 2015, pp. 17–24.

[26] Quintin Goraseb, Nathar Shah, Using conditional random field in named entity recognition for crime location identification, Int. J. Mech. Eng. Robot. Res. (2020) 252–257.

[27] Md. Mamun Hossain, Zarin Rafah Chowdhury, S. M. Rezwanul Haque Akib, Md. Sabbir Ahmed, Md. Moazzem. Hossain, Abu Saleh Musa Miah, Crime text classification and drug modeling from Bengali news articles : A transformer network-based deep learning approach, in: 2023 26th International Conference on Computer and Information Technology, ICCIT, 2023, pp. 1–6.

[28] Rodrigo Juez-Hernandez, Lara Quijano-Sánchez, Federico Liberatore, Jesús Gómez, AGORA: An intelligent system for the anonymization, information extraction and automatic mapping of sensitive documents, Appl. Soft Comput. 145 (2023) 110540.

[29] Hafedh Shabat, Nazlia Omar, Khmael Rahem, Named entity recognition in crime using machine learning approach, in: Azizah Jaafar, Nazlena Mohamad Ali, Shahrul Azman Mohd Noah, Alan F. Smeaton, Peter Bruza, Zainab Abu Bakar, Nursuriati Jamil, Tengku Mohd Tengku Sembok (Eds.), Information Retrieval Technology, Springer International Publishing, Cham, 2014, pp. 280–288.

[30] Hafedh Shabat, Nazlia Omar, Named entity recognition in crime news documents using classifiers combination, Middle- East J. Sci. Res. (2015) 1215–1221.

[31] Siti Azirah Asmai, Muhammad Sharilazlan Salleh, Halizah Basiron, Sabrina Ahmad, An enhanced malay named entity recognition using combination approach for crime textual data analysis, Int. J. Adv. Comput. Sci. Appl. 9 (9) (2018).

[32] Marijn Schraagen, Matthieu Brinkhuis, Floris Bex, Evaluation of named entity recognition in dutch online criminal complaints, Comput. Linguist. Neth. J. 7 (2017) 3–16.

[33] Fillipe Barros Rodrigues, William Ferreira Giozza, Robson de Oliveira Albuquerque, Luis Javier García Villalba, Natural language processing applied to forensics information extraction with transformers and graph visualization, IEEE Trans. Comput. Soc. Syst. 11 (4) (2024) 4727–4743.

[34] Siripen Pongpaichet, Boonyapat Sukosit, Chitchaya Duangtanawat, Jiramed Jamjongdamrongkit, Chancheep Mahacharoensuk, Kantapong Matangkarat, Pattadon Singhajan, Thanapon Noraset, Suppawong Tuarob, CAMELON: A system for crime metadata extraction and spatiotemporal visualization from online news articles, IEEE Access 12 (2024) 22778–22802.

[35] Varsha Naik, Rajeswari Kannan, Sanket Agarwal, Aryan Sable, Himanshu Chaudhari, An effective search algorithm for analyzing and extracting Indian legal judgments using NER and document summarization, in: 2023 7th International Conference on Computing, Communication, Control and Automation, ICCUBEA, 2023, pp. 1–6.

[36] Varsha Naik, Purvang Patel, Rajeswari Kannan, Legal entity extraction: An experimental study of NER approach for legal documents, Int. J. Adv. Comput. Sci. Appl. 14 (3) (2023).

[37] Irene Pérez-Díez, Raúl Pérez-Moraga, Adolfo López-Cerdán, Jose-Maria Salinas-Serrano, María de la Iglesia-Vayá, De-identifying Spanish medical texts - named entity recognition applied to radiology reports, J. Biomed. Semant. 12 (1) (2021) 6.

[38] Tuwailaa Alshammari, Extracting data from unstructured crime text to represent in structured occurrence nets using natural language processing, in: Michael Köhler-Bussmeier, Daniel Moldt, Heiko Rölke (Eds.), Proceedings of the International Workshop on Petri Nets and Software Engineering 2024 Co-Located with the 45th International Conference on Application and Theory of Petri Nets and Concurrency (PETRI NETS 2024), June 24 - 25, 2024, Geneva, Switzerland, in: CEUR Workshop Proceedings, vol. 3730, CEUR-WS.org, 2024, pp. 270–282.

[39] Runxin Xu, Tianyu Liu, Lei Li, Baobao Chang, Document-level event extraction via heterogeneous graph-based interaction model with a tracker, in: Chengqing Zong, Fei Xia, Wenjie Li, 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), Association for Computational Linguistics, Online, 2021, pp. 3533–3546.

[40] Hang Yang, Dianbo Sui, Yubo Chen, Kang Liu, Jun Zhao, Taifeng Wang, Document-level event extraction via parallel prediction networks, in: Chengqing Zong, Fei Xia, Wenjie Li, 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), Association for Computational Linguistics, Online, 2021, pp. 6298–6308.

[41] Shudong Lu, Gang Zhao, Si Li, Jun Guo, Explainable document-level event extraction via back-tracing to sentence-level event clues, Knowl.-Based Syst. 248 (2022) 108715.

[42] Yusheng Huang, Weijia Jia, Exploring sentence community for document-level event extraction, in: Marie-Francine Moens, Xuanjing Huang, Lucia Specia, Scott Wen-tau Yih (Eds.), Findings of the Association for Computational Linguistics: EMNLP 2021, Association for Computational Linguistics, Punta Cana, Dominican Republic, 2021, pp. 340–351.

[43] Tong Zhu, Xiaoye Qu, Wenliang Chen, Zhefeng Wang, Baoxing Huai, Nicholas Yuan, Min Zhang, Efficient document-level event extraction via pseudo-trigger-aware pruned complete graph, in: Lud De Raedt (Ed.), Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22, International Joint Conferences on Artificial Intelligence Organization, 2022, pp. 4552–4558, Main Track.

[44] Fumin Chen, Xu Wang, Xiaohui Liu, Dezhong Peng, A trigger-free method enhanced by coreference information for document-level event extraction, in: 2023 International Joint Conference on Neural Networks, IJCNN, 2023, pp. 1–8.

[45] Guanqiu Qin, Nankai Lin, Menglan Shen, Qifeng Bai, Dong Zhou, Aimin Yang, Global information enhancement and subgraph-level weakly contrastive learning for lightweight weakly supervised document-level event extraction, Expert Syst. Appl. 240 (2024) 122516.

[46] Jian Liu, Yubo Chen, Kang Liu, Wei Bi, Xiaojiang Liu, Event extraction as machine reading comprehension, in: Bonnie Webber, Trevor Cohn, Yulan He, Yang Liu (Eds.), Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP, Association for Computational Linguistics, Online, 2020, pp. 1641–1651.

[47] Liu Liu, Ming Liu, Shanshan Liu, Kun Ding, Event extraction as machine reading comprehension with question-context bridging, Knowl.-Based Syst. 299 (2024) 112041.

[48] Xing David Wang, Leon Weber, Ulf Leser, Biomedical event extraction as multi-turn question answering, in: Eben Holderness, Antonio Jimeno Yepes, Alberto Lavelli, Anne-Lyse Minard, James Pustejovsky, Fabio Rinaldi (Eds.), Proceedings of the 11th International Workshop on Health Text Mining and Information Analysis, Association for Computational Linguistics, Online, 2020, pp. 88–96.

[49] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, in: Jill Burstein, Christy Doran, 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), Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 4171–4186.

[50] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, Percy Liang, SQuAD: 100,000+ questions for machine comprehension of text, in: Jian Su, Kevin Duh, Xavier Carreras (Eds.), Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Austin, Texas, 2016, pp. 2383–2392.

[51] Pranav Rajpurkar, Robin Jia, Percy Liang, Know what you don't know: Unanswerable questions for SQuAD, in: Iryna Gurevych, Yusuke Miyao (Eds.), Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Association for Computational Linguistics, Melbourne, Australia, 2018, pp. 784–789.

[52] Elad Segal, Avia Efrat, Mor Shoham, Amir Globerson, Jonathan Berant, A simple and effective model for answering multi-span questions, in: Bonnie Webber, Trevor Cohn, Yulan He, Yang Liu (Eds.), Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP, Association for Computational Linguistics, Online, 2020, pp. 3074–3080.

[53] Ming Zhu, Aman Ahuja, Da-Cheng Juan, Wei Wei, Chandan K. Reddy, Question answering with long multiple-span answers, in: Trevor Cohn, Yulan He, Yang Liu (Eds.), Findings of the Association for Computational Linguistics: EMNLP 2020, Association for Computational Linguistics, Online, 2020, pp. 3840–3849.

[54] Haonan Li, Martin Tomko, Maria Vasardani, Timothy Baldwin, MultiSpanQA: A dataset for multi-span question answering, in: Marine Carpuat, Marie-Catherine de Marneffe, Ivan Vladimir Meza Ruiz (Eds.), Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, Seattle, United States, 2022, pp. 1250–1260.

[55] Danilo Croce, Alexandra Zelenanska, Roberto Basili, Neural learning for question answering in italian, in: AI* IA 2018–Advances in Artificial Intelligence: XVIIth International Conference of the Italian Association for Artificial Intelligence, Trento, Italy, November 20–23, 2018, Proceedings 17, Springer, 2018, pp. 389–402.

[56] Maja Pavlovic, Massimo Poesio, The effectiveness of LLMs as annotators: A comparative overview and empirical analysis of direct representation, in: Gavin Abercrombie, Valerio Basile, Davide Bernadi, Shiran Dudy, Simona Frenda, Lucy Havens, Sara Tonelli (Eds.), Proceedings of the 3rd Workshop on Perspectivist Approaches To NLP (NLPerspectives) @ LREC-COLING 2024, ELRA and ICCL, Torino, Italia, 2024, pp. 100–110.

[57] Shuohang Wang, Yang Liu, Yichong Xu, Chenguang Zhu, Michael Zeng, Want to reduce labeling cost? GPT-3 can help, in: Marie-Francine Moens, Xuanjing Huang, Lucia Specia, Scott Wen-tau Yih (Eds.), Findings of the Association for Computational Linguistics: EMNLP 2021, Association for Computational Linguistics, Punta Cana, Dominican Republic, 2021, pp. 4195–4205.

[58] Fan Huang, Haewoon Kwak, Jisun An, Is ChatGPT better than human annotators? Potential and limitations of ChatGPT in explaining implicit hate speech, in: Ying Ding, Jie Tang, Juan F. Sequeda, Lora Aroyo, Carlos Castillo, Geert-Jan Houben (Eds.), Companion Proceedings of the ACM Web Conference 2023, WWW 2023, Austin, TX, USA, 30 April 2023 - 4 May 2023, ACM, 2023, pp. 294–297.

[59] Yiming Zhu, Peixian Zhang, Ehsan ul Haq, Pan Hui, Gareth Tyson, Can ChatGPT reproduce human-generated labels? A study of social computing tasks, 2023, CoRR, abs/2304.10145.

[60] Xingwei He, Zhenghao Lin, Yeyun Gong, A-Long Jin, Hang Zhang, Chen Lin, Jian Jiao, Siu Ming Yiu, Nan Duan, Weizhu Chen, AnnoLLM: Making large language models to be better crowdsourced annotators, 2023, CoRR, abs/2303.16854.

[61] Yubo Ma, Yixin Cao, Yong Hong, Aixin Sun, Large language model is not a good few-shot information extractor, but a good reranker for hard samples!, in: Houda Bouamor, Juan Pino, Kalika Bali (Eds.), Findings of the Association for Computational Linguistics: EMNLP 2023, Association for Computational Linguistics, Singapore, 2023, pp. 10572–10601.

[62] Fabrizio Gilardi, Meysam Alizadeh, Maël Kubli, ChatGPT outperforms crowd workers for text-annotation tasks, Proc. Natl. Acad. Sci. USA 120 (30) (2023) Cited by: 9; All Open Access, Green Open Access, Hybrid Gold Open Access.

[63] Petter Törnberg, ChatGPT-4 outperforms experts and crowd workers in annotating political Twitter messages with zero-shot learning, 2023, ArXiv, abs/2304.06588.

[64] Jianxun Chen, Peng Chen, Xuxu Wu, Generating Chinese event extraction method based on ChatGPT and prompt learning, Appl. Sci. (Switz.) 13 (17) (2023).

[65] Toyin D. Aguda, Suchetha Siddagangappa, Elena Kochkina, Simerjot Kaur, Dongsheng Wang, Charese Smiley, Large language models as financial data annotators: A study on effectiveness and efficiency, in: Nicoletta Calzolari, Min-Yen Kan, Veronique Hoste, Alessandro Lenci, Sakriani Sakti, Nianwen Xue (Eds.), Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation, LREC-COLING 2024, ELRA and ICCL, Torino, Italia, 2024, pp. 10124–10145.

[66] Karel D'Oosterlinck, François Remy, Johannes Deleu, Thomas Demeester, Chris Develder, Klim Zaporojets, Aneiss Ghodsi, Simon Ellershaw, Jack Collins, Christopher Potts, BioDEX: Large-scale biomedical adverse drug event extraction for real-world pharmacovigilance, in: Houda Bouamor, Juan Pino, Kalika Bali (Eds.), Findings of the Association for Computational Linguistics: EMNLP 2023, Association for Computational Linguistics, Singapore, 2023, pp. 13425–13454.

[67] Ruirui Chen, Chengwei Qin, Weifeng Jiang, Dongkyu Choi, Is a large language model a good annotator for event extraction? in: Michael J. Wooldridge, Jennifer G. Dy, Sriraam Natarajan (Eds.), Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada, AAAI Press, 2024, pp. 17772–17780.

[68] Christopher Walker, Stephanie Strassel, Julie Medero, Kazuaki Maeda, ACE 2005 multilingual training corpus, 2006.

[69] Fatemeh Shiri, Farhad Moghimifar, Reza Haffari, Yuan-Fang Li, Van Nguyen, John Yoo, Decompose, enrich, and extract! schema-aware event extraction using llms., in: 2024 27th International Conference on Information Fusion, FUSION, 2024, pp. 1–8.

[70] Hanzhang Zhou, Junlang Qian, Zijian Feng, Lu Hui, Zixiao Zhu, Kezhi Mao, LLMs learn task heuristics from demonstrations: A heuristic-driven prompting strategy for document-level event argument extraction, in: Lun-Wei Ku, Andre Martins, Vivek Srikumar (Eds.), Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Bangkok, Thailand, 2024, pp. 11972–11990.

[71] Xinliang Frederick Zhang, Carter Blum, Temma Choji, Shalin Shah, Alakananda Vempala, ULTRA: Unleash LLMs' potential for event argument extraction through hierarchical modeling and pair-wise self-refinement, in: Lun-Wei Ku, Andre Martins, Vivek Srikumar (Eds.), Findings of the Association for Computational Linguistics: ACL 2024, Association for Computational Linguistics, Bangkok, Thailand, 2024, pp. 8172–8185.

[72] Sha Li, Heng Ji, 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, Yichao Zhou (Eds.), Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, Online, 2021, pp. 894–908.

[73] I-Hung Hsu, Kuan-Hao Huang, Elizabeth Boschee, Scott Miller, Prem Natarajan, Kai-Wei Chang, Nanyun Peng, DEGREE: A data-efficient generation-based event extraction model, in: Marine Carpuat, Marie-Catherine de Marneffe, Ivan Vladimir Meza Ruiz (Eds.), Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, Seattle, United States, 2022, pp. 1890–1908.

[74] Jiaren Peng, Wenzhong Yang, Fuyuan Wei, Liang He, Prompt for extraction: Multiple templates choice model for event extraction, Knowl.-Based Syst. 289 (2024) 111544.

[75] Philip Blair, Kfir Bar, DEGREE^2: Efficient extraction of multiple events using language models, in: Joel Tetreault, Thien Huu Nguyen, Hemank Lamba, Amanda Hughes (Eds.), Proceedings of the Workshop on the Future of Event Detection, FuturED, Association for Computational Linguistics, Miami, Florida, USA, 2024, pp. 25–31.

[76] Bosheng Ding, Chengwei Qin, Linlin Liu, Yew Ken Chia, Boyang Li, Shafiq Joty, Lidong Bing, Is GPT-3 a good data annotator? in: Anna Rogers, Jordan L. Boyd-Graber, 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, Association for Computational Linguistics, 2023, pp. 11173–11195.

[77] Federica Rollo, Laura Po, Crime event localization and deduplication, in: Jeff Z. Pan, Valentina Tamma, Claudia d'Amato, Krzysztof Janowicz, Bo Fu, Axel Polleres, Oshani Seneviratne, Lalana Kagal (Eds.), The Semantic Web – ISWC 2020, Springer International Publishing, Cham, 2020, pp. 361–377.

[78] 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, Thomas Scialom, Llama 2: Open foundation and fine-tuned chat models, 2023, CoRR, abs/2307.09288.

[79] Aaron Grattafiori, et al., The llama 3 herd of models, 2024.

[80] Marco Polignano, Pierpaolo Basile, Giovanni Semeraro, Advanced natural-based interaction for the ITAlian language: LLaMAntino-3-ANITA, 2024.

[81] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, William El Sayed, Mistral 7B, 2023, CoRR, abs/2310.06825.

[82] Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, William El Sayed, Mixtral of experts, 2024, CoRR, abs/2401.04088.

[83] Paul Jaccard, Etude de la distribution florale dans une portion des alpes et du jura, Bull. Soc. Vaudoise Des Sci. Nat. 37 (1901) 547–579.

[84] Paul Jaccard, The distribution of the flora in the alpine zone.1, New Phytol. 11 (2) (1912) 37–50.

[85] Jerrold H. Zar, Spearman rank correlation, Encycl. Biostat. 7 (2005).

[86] Maja Pavlovic, Massimo Poesio, The effectiveness of LLMs as annotators: A comparative overview and empirical analysis of direct representation, 2024, CoRR, abs/2405.01299.

[87] Kevin Clark, Minh-Thang Luong, Quoc V. Le, Christopher D. Manning, ELECTRA: pre-training text encoders as discriminators rather than generators, in: 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020, OpenReview.net, 2020.

[88] Tim Dettmers, Mike Lewis, Younes Belkada, Luke Zettlemoyer, LLM.int8: 8-bit matrix multiplication for transformers at scale, in: Proceedings of the 36th International Conference on Neural Information Processing Systems, NIPS '22, Curran Associates Inc., Red Hook, NY, USA, 2022.

[89] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, Luke Zettlemoyer, Qlora: Efficient finetuning of quantized LLMs, in: A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, S. Levine (Eds.), Advances in Neural Information Processing Systems, Vol. 36, Curran Associates, Inc., 2023, pp. 10088–10115.

[90] Renren Jin, Jiangcun Du, Wuwei Huang, Wei Liu, Jian Luan, Bin Wang, Deyi Xiong, A comprehensive evaluation of quantization strategies for large language models, in: Lun-Wei Ku, Andre Martins, Vivek Srikumar (Eds.), Findings of the Association for Computational Linguistics: ACL 2024, Association for Computational Linguistics, Bangkok, Thailand, 2024, pp. 12186–12215.

[91] Yujun Lin, Haotian Tang, Shang Yang, Zhekai Zhang, Guangxuan Xiao, Chuang Gan, Song Han, QServe: W4A8KV4 quantization and system co-design for efficient LLM serving, 2024.

[92] Yilong Zhao, Chien-Yu Lin, Kan Zhu, Zihao Ye, Lequn Chen, Size Zheng, Luis Ceze, Arvind Krishnamurthy, Tianqi Chen, Baris Kasikci, Atom: Low-bit quantization for efficient and accurate LLM serving, in: P. Gibbons, G. Pekhimenko, C. De Sa (Eds.), Proceedings of Machine Learning and Systems, Vol. 6, 2024, pp. 196–209.

[93] Shang Yang, Junxian Guo, Haotian Tang, Qinghao Hu, Guangxuan Xiao, Jiaming Tang, Yujun Lin, Zhijian Liu, Yao Lu, Song Han, LServe: Efficient long-sequence LLM serving with unified sparse attention, 2025.

[94] Dongyoung Lee, Seungkyu Choi, Ik Joon Chang, Qrazor: Reliable and effortless 4-bit LLM quantization by significant data razoring, 2025.

[95] Ana Lucic, Maurits Bleeker, Samarth Bhargav, Jessica Forde, Koustuv Sinha, Jesse Dodge, Sasha Luccioni, Robert Stojnic, Towards reproducible machine learning research in natural language processing, in: Luciana Benotti, Naoaki Okazaki, Yves Scherrer, Marcos Zampieri (Eds.), Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts, Association for Computational Linguistics, Dublin, Ireland, 2022, pp. 7–11.

[96] Yiming Zhang, Shi Feng, Chenhao Tan, Active example selection for in-context learning, 2022, pp. 9134–9148.

[97] Branislav Pecher, Ivan Srba, Maria Bielikova, Joaquin Vanschoren, Automatic combination of sample selection strategies for few-shot learning, 2024.

[98] Chengwei Qin, Aston Zhang, Chen Chen, Anirudh Dagar, Wenming Ye, In-context learning with iterative demonstration selection, 2024.

[99] Xiaonan Li, Xipeng Qiu, Finding support examples for in-context learning, 2023, pp. 6219–6235.

[100] Xi Ye, Srinivasan Iyer, Asli Celikyilmaz, Veselin Stoyanov, Greg Durrett, Ramakanth Pasunuru, Complementary explanations for effective in-context learning, in: Anna Rogers, Jordan Boyd-Graber, Naoaki Okazaki (Eds.), Findings of the Association for Computational Linguistics: ACL 2023, Association for Computational Linguistics, Toronto, Canada, 2023, pp. 4469–4484.

[101] Hongjin Su, Jungo Kasai, Chen Henry Wu, Weijia Shi, Tianlu Wang, Jiayi Xin, Rui Zhang, Mari Ostendorf, Luke Zettlemoyer, Noah A. Smith, Tao Yu, Selective annotation makes language models better few-shot learners, 2023.