8 APPENDIX

8.1 LLM instructions

Here are the instructions with the complete schema for both datasets. Under the attribute recommendation setting, the attributes for validation and testing are excluded from the schema and hidden from the LLM.

Extraction Instruction - Incidents

Your task is to analyze the given text of Incidents and extract the relevant information according to the specified JSON structure below.
Provide the extracted information in this JSON structure:

```
{
"Accident": {
        "0":{
                "Accident type": [],
                "Accident date": [],
                "Accident address": [],
                "Number of rounds fired": [],
                "Accident number": [],
                "Personnel arrived time": [],
        }
},
"Victim": {
        "0": {
                "Victim number": [],
                "Victim status": [],
                "Victim gender": [],
                "Victim age": [],
                "Victim based": [],
                "Hospital name": [],
                "Victim name": [],
                "Victim race": [],
                "Victim occupation": [],
                "Victim vehicle": [],
        },
        "1":{...},
        ...
},
"Suspect": {
        "0":{
                "Suspect gender": [],
                "Suspect number": [],
                "Suspect status": [],
                "Suspect age": [],
                "Suspect name": [],
                "Suspect description": [],
                "Suspect weapon": [],
                "Suspect vehicle": [],
                "Suspect occupation": [],
                "Suspect based": [],
                "Suspect race": [],
                "Prison name": [],
        },
        "1":{...},
        ...
}
}
```

Explanations about the JSON structure:
- Output only the JSON data.
- There can be three types of entities mentioned in the texts: Accident, Victim, and Suspect. There is at most one Accident in the text. There can be multiple Victims or Suspects in the text.
- For each entity, there are multiple attributes.
- For each attribute, its value is a list of exact substrings of the input text. You should consider every attribute listed in the above structure. If an attribute is not present or cannot be determined, do not include the attribute in the output.

Here are some examples:
{examples}
Here is the input text:
{text}
Here are some hints:
The above text is most likely to contain the attributes: $\{H^p\}$.

The above text is most likely to contain the following values: $\{V^p\}$.

Extraction Instruction - Weather

Your task is to analyze the given text of Incidents and extract the relevant information according to the specified JSON structure below.
Provide the extracted information in this JSON structure:

```
{
"Weather": {
        "0":{
                "Temperature": [],
                "Snow status": [],
                "Cloud type": [],
                "Sunset time": [],
                "Rain status": [],
                "Weather area": [],
                "Time": [],
                "Weather type": [],
                "Wind speed": [],
                "Weather compass direction": [],
                "Maximum temperature": [],
                "Weather frequency": [],
                "Wind status": [],
                "Minimum temperature": [],
                "Sunrise time": [],
                "Weather occurring chance": [],
                "Wind direction": [],
                "Location": [],
                "Cloud status": []
```

```
    },
    "1":{...},
    ...
    }
}
```

Explanations about the JSON structure:
- Output only the JSON data.
- There is one type of entity mentioned in the texts: Weather. There can be multiple Weather in the text.
- For each entity, there are multiple attributes.
- For each attribute, its value is a list of exact substrings of the input text. You should consider every attribute listed in the above structure. If an attribute is not present or cannot be determined, do not include the attribute in the output.

Here are some examples:
{examples}
Here is the input text:
{text}
Here are some hints:
The above text is most likely to contain the attributes: $\{H^p\}$.

The above text is most likely to contain the following values: $\{V^p\}$.

Discovery Instruction - Incidents

Your task is to analyze the given text and discover new information mentioned in the text.
Provide the discovered information in this JSON structure:

```
{
"Accident": {
    "0":{
        "⟨ new attribute 1⟩": [],
        "⟨new attribute 2⟩": []
    }
},
"Victim": {
    "0":{
        "⟨new attribute 1⟩": [],
        "⟨new attribute 2⟩": []
    },
    "1":{...},
    ...
},
"Suspect": {
    "0":{

        "⟨new attribute 1⟩": [],
        "⟨new attribute 2⟩": []
    },
    "1":{...},
    ...
},
}
```

Explanations about the JSON structure:
- Output only the JSON data.
- explanation
- For each entity, there may be multiple attributes.
- For each attribute, its value is a list of exact substrings of the input text.

Here are some examples:
{example text}
Explanations about the examples:
- Given an input text and the known information, you are encouraged to discover new information for each entity.
- The new information is qualified as a new attribute if and only if (1) it is relevant and important to the domain, (2) it is unique and not overlapped with the known information, and (3) it has proper granularity level compared with the known attributes, which means it is not too high-level or low-level concepts.
- The new attribute must have a meaningful name and actual values from the text.
- Output qualified new attributes as many as possible.

Here is the input text:
{text}
Here is the known information:
{extracted table}

Discovery Instruction - Weather

Your task is to analyze the given text and discover new information mentioned in the text.
Provide the discovered information in this JSON structure:

```
{
"Weather": {
    "0":{
        "⟨new attribute 1⟩": [],
        "⟨new attribute 2⟩": []
    },
    "1":{...},
    ...
    }
}
```

Explanations about the JSON structure:
- Output only the JSON data.
- explanation
- For each entity, there may be multiple attributes.
- For each attribute, its value is a list of exact substrings of the input text.

Here are some examples:
{discovery example}

Explanations about the examples:
- Given an input text and the known information, you are encouraged to discover new information for each entity.
- The new information is qualified as a new attribute if and only if (1) it is relevant and important to the domain, (2) it is unique and not overlapped with the known information, and (3) it has a proper granularity level compared with the known attributes, which means it is not too high-level or low-level concepts.
- The new attribute must have a meaningful name and actual values from the text.
- Output qualified new attributes as many as possible.

Here is the input text:
{text}
Here is the known information:
{extracted table}

Decriptive Prefix - Incidents

Here is a schema designed to document and manage detailed information and facts about criminal incidents or accidents, particularly those involving violence, such as shootings or other crimes. The attributes are organized into three main entities: the accident itself, the victims, and the suspects. For each entity, there are multiple attributes. The attributes are unique and in consistent styles. Here are the attributes in this schema:

Decriptive Prefix - Weather

Here is a schema designed to document and manage detailed information and facts about weather conditions from news reports, particularly those about location, time, temperature, wind, cloud, rain, and snow. The attributes are organized into one main entity: the Weather itself, and there are multiple attributes. The attributes are unique and in consistent styles. Here are the attributes in this schema:

Judge Instruction in Hybrid Integration Strategy

Your task is to analyze the given schema and judge whether the given concepts are semantically similar enough to be merged as one attribute. Here is the schema:
{schema_text}
Here are the concepts to be judged:
{elements}
Explanation about the task:
- The schema includes several unique attributes, which show the semantic granularity of attributes.
- If the above concepts (1) are semantically equivalent, or (2) one is semantically contained by the others, or (3) they represent a finer granularity compared with the existing attributes, they need to be merged into one single attribute. Then, output "One".
- Else, they should be at least two unique attributes, output "Two".
- Output only "Two" or "One".

8.2 Computation of Vendi Score

Vendi Score is the effective rank of the normalized similarity matrix,

$$vds(q) = \exp(-\mathrm{tr}(\frac{\mathcal{K}_{q\times q}}{|q|}\log\frac{\mathcal{K}_{q\times q}}{|q|})), \tag{26}$$

where $\mathcal{K}_{q\times q}$ is the similarity matrix of size $|q|\times|q|$ and $\mathrm{tr}(\cdot)$ means trace. Vendi Score is maximized as $|q|$ if $\mathcal{K}(i,j)=0, \forall i,j\in q, i\neq j$, meaning every two element is totally different, and it is minimized as 1 if $\mathcal{K}_{q\times q}$ contains all 1s. Since $q\subseteq Z\cup S=B$. We could precompute and cache $\mathcal{K}_{B\times B}$ and acquire submatix $\mathcal{K}_{q\times q}$ from it. Given $\mathcal{K}_{q\times q}$, the computational complexity of is in $O(|q|^3)$ [11].

8.3 Pruned Search for Suspicious sets

As established by Equation 18, if a set is identified as suspicious, then all of its subsets must also be suspicious. In other words, the monotonicity of $div(\cdot)$ means $div(p\cup\{z\})\geq div(p), \forall z\notin p$. Leveraging this property, our algorithm computes suspicious sets in ascending order of their size, enabling efficient pruning during the process, as in Algorithm 2.

Specifically, in each iteration of the outer loop, the algorithm searches all suspicious sets of size len+1. Br is a dictionary mapping a suspicious set p of size len to a set of attributes, $\mathrm{Br}[p]\subseteq Z\cup S$, which are possible branches to be added to p for constructing a larger suspicious set. The inner loop tests whether enlarging a suspicious set p by one element z' is also a suspicious set. This first involves a quick dictionary hashing check by replacing $\forall z\in p$ with z'. If it passes the check, implying all proper subsets of $p\cup\{z'\}$ are suspicious, and we compute the $vds(\cdot)$ for $p'=p\cup\{z'\}$ to decide the final answer.

8.4 Generating Attribute Proposal Definitions

Since each attribute proposal varies in occurrence frequency, number of mentions, and mention length across source texts, directly gathered raw contexts can be highly inconsistent in length and style. This variability introduces bias to the LLM and could result in overly long input that hinders reliable reasoning. To address this, we first normalize each proposal's context into a concise definition using the LLM itself.

This step produces a standardized context(z) for every attribute proposal$z$ that needs to be judged by the LLM, ensuring consistent style and length and improving fairness. Let $D^{u+}_{i_1}, D^{u+}_{i_2}, \ldots$ denote the pseudo-tables containing the attribute proposal z. The LLM instruction for generating a standard context is shown below.

Generating context(z) for attribute proposal z.

Your task is to summarize the context of an attribute and write a definition for it.

Algorithm 2 Pruned Bottom-up Search for P^*

Input: attribute proposals Z, known attributes S, threshold δ
Output: P^*

$P^* = \emptyset$, Br$\leftarrow$ {}, len$\leftarrow$ 1 ▷ Initialize dictionary.
for $p \in Z \cup S$ **do** Br$[p] = Z \cup S \setminus \{p\}$ ▷ Start by len=1.
end for
while |Br|$>0 \wedge$len$<|Z|+1$ **do** ▷ Outer loop.
  nextBr$\leftarrow$ {}
  for $p \in$ Br, $z' \in$ Br$[e]$ **do** ▷ Inner loop.
    valid $\leftarrow$ *True*, activeBr $\leftarrow$ Br$[p]$
    for $z \in p$ **do**: ▷ Check all subset of length $|p|$
      $p' = p \setminus \{z\} \cup \{z'\}$ ▷ Replace one element, $|p'| = |e|$
      if $p' \in$ Br **then**
        activeBr $\leftarrow$ activeBr $\cap$ Br$[p']$ ▷ Prune.
      else
        valid$\leftarrow$ *False*
        Break
      end if
    end for
    if valid = *True* **then** ▷ All proper subset of p is in P^*.
      $p' = p \cup \{z'\}$
      if $vds(p') < 1 + \delta$ **then**
        $P^* \leftarrow P^* \cup \{p'\}$
        nextBr$[p'] \leftarrow$ activeBr
      end if
    end if
  end for
  len$\leftarrow$len+1, Br$\leftarrow$ nextBr ▷ Search for the larger size
end while

Provide the definition in this JSON structure:
{ "<name>": "<definition>", }
Explanations about the JSON structure:
- Output only the JSON data, where the key is the name of the attribute and the value is its definition.
- { task description }
- The name should be concise and in the same style with the examples.
- The name should be of proper granularity. Note that if it is too general, it cannot accurately describe the range of values; if it is too specific, it cannot cover all the values.
- The definition should precisely describe the meaning and characteristics of the attribute, enabling the reliable extraction of the attribute from other texts.
- The definition should be less than 50 words.
{ example definition }
Here is the context of the attribute to be defined:
In the text $W^u_{i_1}$, the z value are: { values in $D^{u+}_{i_1}$ }
In the text $W^u_{i_2}$, the z value are: { values in $D^{u+}_{i_2}$ }
...

Here the "task description" is a brief description of the dataset, and "example definitions" is the definition for known attributes in the schema, if there are any.

8.5 Other Results

Figure 7 and Figure 8 are the results with Llama-backbone.

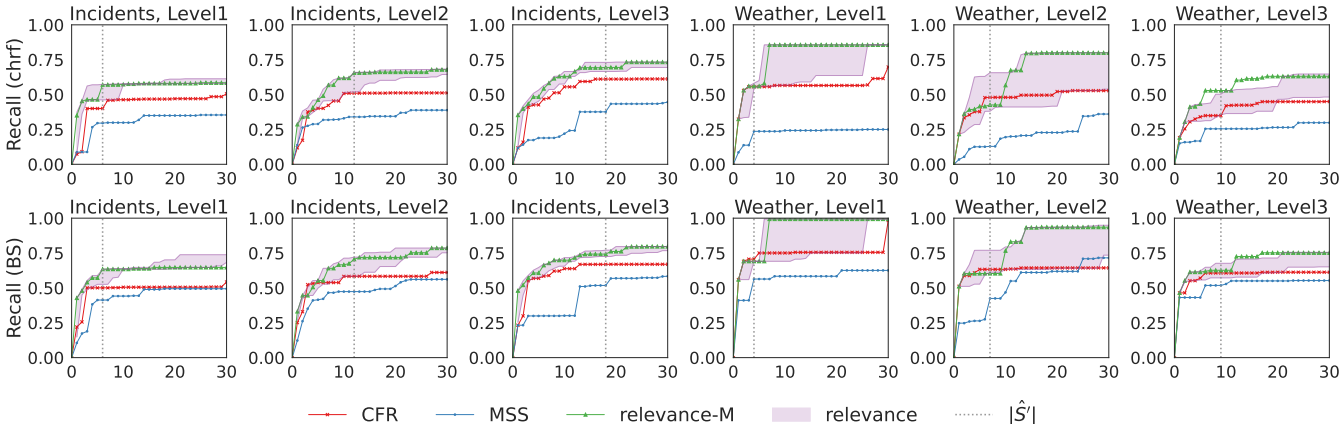

Figure 7: Recommendation recall with Llama.

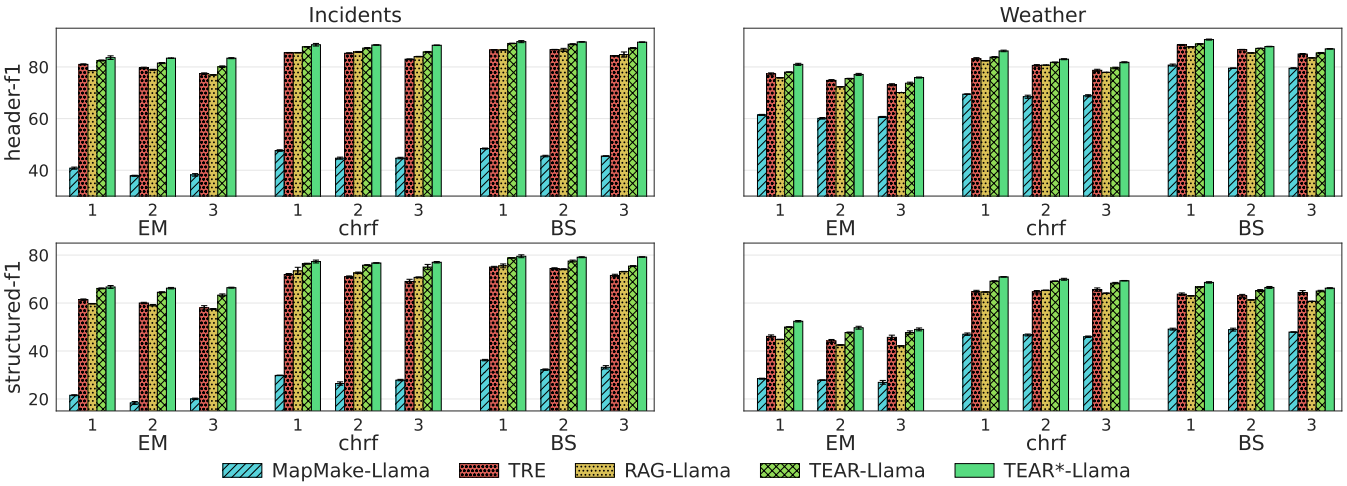

Figure 8: Table extraction with text-driven attributes, Llama backbone.