Literature Review: XCS224U

October 2023

Samuel Malu

[malusamuel.m@gmail.com](mailto:malusamuel.m@gmail.com)

Project Topic: Using Large Language Models for News Incident/Event Information Extraction

1. Introduction

Information Extraction (IE) is a natural language processing (NLP) task that aims to extract structured or semi-structured information from unstructured text. Large Language Models (LLMs) have proven to be effective in a range of Natural Language Understanding tasks. This document explores existing literature on the NLP task of IE usingLLMs, and for some papers, we take a deeper look at findings regarding the Event Extraction IE task which is the selected topic of the final project.

1. Problem statement

It is sometimes desirable to extract information from unstructured text such as news articles, social media posts and documents. Named Entity recognition (NER), Event extraction (EE), and Entity-Relation extraction (RE) are some of the main IE tasks that can be applied to unstructured tasks (Wei et al., 2023)

Typically, models are trained to perform to perform specialized and narrow IE task, when new patterns are discovered, more training is required. However, training/fine-tuning of new patterns is made redundant with LLMs. Additionally, LLMs might have emergent capabilities that enable us to extract and infer nuanced information that would otherwise not have been possible with traditional IE models.

However, some IE tasks such as EE are hard, and LLMs still perform worse than some other approaches (Ma et al. 2023; Ling et al. 2023). Are there approaches we can take to significantly improve performance of LLM with IE tasks?

1. Summary of literature
   1. Paper 1: Zero-Shot Information Extraction via Chatting with ChatGPT

**Datasets:** NYT11-HR, conllpp, DuEE1.0

In this paper Wei et al. illustrate a new framework for IE using ChatGPT. The framework entails using a multi-turn QA strategy, i.e., composing multiple Question Answering (QA) interactions with ChatGPT to break down the IE task to simple sub-tasks. Two major IE tasks are explored: EE and RE.

The framework works in two stages, the first step involves a single turn of QA to perform and initial classification task. For EE, this involves returning a list of inferred event types from the input context; for RE, this involves returning a list of potential relationship types.

Results show that vanilla zero-shot ChatGPT performs poorly in IE task, but the performance significantly improves with a multi-stage (in this case 2- stage) QA strategy. Performance further improves in a few-shot setting when example QA inputs are randomly selected and included in the query context.

* 1. Paper 2: Exploring the Feasibility of ChatGPT for Event Extraction

**Datasets:** Automatic Context Extraction (ACE 2005)

In this paper, Gao et al. explore event extraction using ChatGPT. Particularly, they explore the area of **Event Detection**. Following the ACE 2005 dataset definition, Event **Detection (ED)** is differentiated from the broader EE task in that it does not include **event argument extraction** (e.g., parties/entities involved) – definition used as defined in the ACE 2005 dataset.

This paper found that ChatGPT can extract events from unstructured text with simple scenarios. Providing ChatGPT with the set of possible event types/class as well as providing examples (few-shot) was critical to good results. However, ChatGPT struggled to output accurate results for long-tail (rare) and complex scenarios. Additionally, it had a hard time consistently returning structured data (JSON), however, it is worth to note that the structured format was only provided through few shot examples and a template was not explicitly provided alongside the event type definitions.

* 1. Paper 3: Information Extraction with Large Language Models - Parsing Unstructured Data with GPT-3

Dataset: Hacker News job postings

In this blog post, Maier explores IE by parsing unstructured text with GPT-3 Da-Vinci model. Particularly, they instruct GPT3 to parse job posting text to a JSON formatted output with some predefined properties (e.g., job title, company name, location etc.) in a zero-shot setting.

The literature illustrates that it is trivial to use prompt engineering to instruct LLMs to extract structured data, even in a zero-shot setting. It is however highlighted that since the illustrated task is purely an extraction operation with not much nuance (job postings are mostly non-ambiguous), then it is desirable to dial down the model’s ‘creativity’, e.g., setting the **temperature** configuration to zero (or a low value).

It is worth to note an interesting method used in this literature where text embedding for job postings is queried from OpenAI targeting some model. These embedding can then be used to preform similarity dependent operations such as sorting by some criteria. To make these embeddings performant, the author reduced the dimensions of these embeddings, which were still able to produce reliable results.

* 1. Paper 4: Improving Open Information Extraction with Large Language Models: A Study On Demonstration Uncertainty

Datasets: OIE2016, ReOIE, CaRB

Ling et al. Explore ways to improve the task of Open Information Extraction – basically relationship extraction- using LLMs, by proposing a 4-step approach to enhance in-context learning.

For example, using ChatGPT within the same context windows, the steps are as follows:

* **Define the task:** Prompt the model on an initial task definition including what structure to format the results, additionally include a few demonstrations.
* **Correcting the errors:** Prompt the model to extract several arbitrary relationship triplets (using some ‘train’ dataset), letting ChatGPT know when it failed and providing it with the correct response as a correction mechanism.
* **Demonstrations selection:** Perform a last prompt with the target sentence and a set of structurally similar illustrations (from the ‘train’ set). Similar sentences can be identified by comparing sentence embeddings obtained from any LLM.
* **Response ensembling:** As part of the final prompt, instruct ChatGPT to return all the possible relationship triplet combinations including those with low confidence. Intuitively, these combinations will include not just results derived from the target sentences, but also those derived from the demonstrations (and may include duplicates)

An uncertainty score is calculated that is inversely proportional to the count if the unique combination in the results. A filtering function is then employed to remove results that are above some uncertainty threshold.

This 4-step in-context few-shot learning approach is seen to have significant improvement to the zero-shot approach. Even though supervised fined-tuned models still performed better than the few-shot LLM approach, the gap is low, and the literature speculates that this gap might reduce when LLM with larger parameters are used.

* 1. Large Language Model Is Not a Good Few-shot Information Extractor, but a Good Reranker for Hard Samples!

Datasets: CONLL 2003, OntoNotes, FewNERD, TA-CRED, TACREV, ACE05, MAVEN, ERE

Ma et al. argue that LLM fundamentally perform poorly in few-shot settings for IE operations; this includes in all three IE categories: NER, RE and ED. This is when compared with fine-tuned Small Pre-trained Language models (SLMs) such as RoBerTa. It is shown that IE tasks such as event detection suffer limitations of in-context since nuance is lost the number of event type increases (lower number of event types showed improved performance)

The literature however highlights that LLMs are good at reranking results with low confidences scores, of hard IE problems; the reason being that LLMs have inherence knowledge and some emergent properties that can better infer nuances.

1. Discussion
   1. Prompting

LLMs can return structured data depending on the instructions it is provided. While Gao et al. noted inconsistencies with ChatGPT sometimes returning invalid formats, this might be attributed by insufficient templating of the desirable structure/schema in the context. All the reviewed literature agrees that the quality of prompts has a significant effect on the quality of the results. It is show that few-shot approaches almost always perform better.

* 1. Prompt Sequencing

There are 3 main prompt sequencing approaches illustrated in the reviewed literature for IE tasks.

The first one involves breaking the prompt to multiple sub-tasks (Wei et al. 2023), e.g., for EE, this would first involve prompting for the event **detection/classification task** then followed by the **event argument extraction** task.

The second approach involves making multiple interactive prompts in the same ‘context window’ where we teach, correct, and demonstrate to the LLM our desired results (Ling et al.)

The third approach simply includes a single step where all context, including the schema structure and demonstrates are sent with one prompt.

* 1. LLMs vs Fine-tuned Small pretrained LMs (SLMs)

All the reviewed literature concurs that fine-tuned SLMs perform much better for all IE tasks, though it is acknowledged that LLMs are still powerful due to their emergent capabilities as well as inherently holding ‘external knowledge’ that can augment the provided context; they are not simply dumb IEs.

1. Future work

Does IEs improve with the size of LLMs?

It is not clear what trend of performance improvement there is for IE tasks as the size of LLMs has increased (linear, exponential etc.), and how does this compare with the sizes of smaller finetuned language models.

Multi-prompting of EE

Would be good to have a comprehensive comparison of the two prompting approaches for EE i.e. between two-stage prompting (classification then argument extraction) and single-stage prompting in both zero-shot and few-shot in context learning settings. A two-stage approach allows for the separation of event argument schemas, while a single-stage approach would require a flattened schema: how would these differences impact performance?

Limited Event Type Classes of EE tasks

For datasets with many event types/classes, performance of EE tasks is seen to be lower that dataset with fewer event types (Ma et al. 2023). Is it possible that LLMs might perform better at EE tasks that have a narrower event type variability, but that could still benefit from the nuanced capabilities of LLMs?

References