Gen AI vs. NLP for Legal Entity Extraction

Porchourng Chuor, Ashwin Ittoo
HEC ULiege, Université de Liège, Liège, Belgium
e-mail: porchourng.chuor@uliege.be, ashwin.ittoo@uliege.be

Policies and regulations, such as those from the EU, are subject to regular updates and amendments. This evolution leads to an increased demand for data in the form of reporting requirements from the concerned parties, resulting in higher regulatory reporting burdens. Urgent needs to streamline regulatory burdens to harmonize reporting requirement. For policy-making institutions, such as the European Commission (EC), the first step towards this goal is to standardize the regulatory reporting metadata of EU legislation. Typically, reporting obligations specify the information that a reporting entity must disclose to a regulatory entity by a given date. Examples are in Figure 1. As can be seen, reporting obligations expressed in natural language tend to have a complex sentential structure. The two reporting obligations presented are very different and do not follow any predefined templates. Different entities, such as the information to be reported and the regulatory entity, can play the same syntactic role. This lack of structure makes it difficult for humans to read and interpret. More importantly, it hinders the development of methods for automating the reporting process. Therefore, in this work, we present a framework for transforming reporting obligations into a structured format. A key step to achieve this objective is the extraction of relevant entities from reporting obligations. This is the problem we aim to address. In our use case, financial reporting regulation, we focus on the following entities: *Addresser*: Who reports what to whom; *Action*: What action that is performed; *ActionResult*: What is done; *Addressee*: To whom the action is directed; *Date*: When the action occurs or deadline. A popular corpus of regulatory reporting obligations is Eur-Lex[1], which is maintained by the EC. However, the majority of such corpora are not labeled, precluding the use of supervised approaches. In our study, we investigate two methods to address this issue. First, we employ a rule-based syntactic dependency parsing approach. Second, we rely on few-shot prompting with a pre-trained Large Language Model (LLM) [2], which has shown much promise in information extraction (IE) [1]. Several LLMs have been proposed recently. However, an important research question, which is yet to be addressed, which LLM achieves better performance in IE regulatory reporting. Another question is whether an LLM can outperform a rule-based syntactic parsing method. In this paper, we address these research questions. Specifically, in our syntactic parsing approach, we first generated the syntactic dependencies in the reporting obligations. Next, we used syntactic roles to identify entities. For example, a noun sequence as the subject of a main verb is identified as the *Addresser* (e.g. *competent authority of the Member States* in Figure.1). Other entities, such as the *Addressee, ActionResult*, and *Action*, are

[1] https://eur-lex.europa.eu/homepage.html

detected in a similar manner. Concerning our LLM approach, we benchmarked the performance of two LLMs, namely, Llama-3-8B-Instruct [2] (we refer to this model as Llama3) and ChatGPT-4o [3], the latest version of the GPT model. We adopted a few-shot prompting strategy composed of three components: ***{context, entity explanation, exemplars}***. Our experiments were conducted on a subset of the Eur-Lex corpus. The best performance achieved by the syntactic parsing approach was an F1-score of 0.62 for detecting the *Action* entity from the reporting obligations, while its worst performance was 0.07 for the *Date* entity. Furthermore, this approach faced several challenges, including errors in syntactic parse trees and complex dependency structures that made the parsing process brittle and difficult to generalize. Different sentence structures required significant customization of the tree traversal process. Conversely, the performance of both LLMs, namely Llama3 and ChatGPT-4o, was significantly better. Llama3 achieved its highest F1-score of 0.98 for detecting the *Action* entity, while its lowest performance was an F1-score of 0.48 for the *ActionResult* entity. ChatGPT-4o achieved its best and worst performances on these same entities, with F1-scores of 0.84 and 0.27, respectively. An interesting observation is that, overall, Llama3 outperforms ChatGPT-4o, despite being a smaller model. Our key contributions are as follows: i) We investigated the performance of a standard NLP approach based on syntactic rules against LLMs for IE from reporting obligations; ii) As LLMs, we focused on Llama3 and ChatGPT-4o, demonstrating that Llama3 outperforms ChatGPT-4o despite its smaller size; iii) we provide a thorough analysis of the results from both the LLMs and the syntactic parsing approach.

Example 1: Individual suspected adverse reaction reports and follow-ups submitted to the Eudravigilance database by marketing authorisation holders shall be transmitted electronically upon receipt to the competent authority of the Member State where the reaction occurred.
Example 2: Within 30 days of receipt of the assessment report, the marketing authorisation holder and the members of the Pharmacovigilance Risk Assessment Committee may submit comments to the Agency and to the rapporteur.

Figure 1: Here are examples of regulatory reporting, where each color represents a different entity: red for the *Addresser entity*, green for the *Action entity*, blue for the *ActionResult entity*, purple for the *Addressee entity*, and orange for the *Date entity*.

References

[1] Monica Agrawal et al. Large language models are few-shot clinical information extractors. In Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 1998–2022, December 2022.

[2] Tom Brown et al. Language Models are Few-Shot Learners. In Advances in Neural Information Processing Systems, volume 33, pages 1877–1901, 2020.

[2] https://huggingface.co/meta-llama/Meta-Llama-3-8B-Instruct
[3] We used the OpenAI API.