Structuration des sorties des grands modèles de langue: limitations et perspectives

Description

Les grands modèles de langue sont de plus en plus utilisés pour générer des informations structurées suivant un schéma pré-défini. Cela peut être dans le cadre d'une seule recherche d'information ou d'interactions complexes avec des agents autonomes. Le plus souvent, de telles données sont représentées au format JSON et les schémas dans le standard JSON Schema. Des techniques telles que le *Constrained Decoding* ont été introduites pour permettre une telle génération. Or, selon des travaux récents tels que [2] et [3], le résultat obtenu n'est pas toujours fidèle au schéma spécifié par l'utilisateur et une telle technique entraîne souvent un *overhead* non-négligeable du fait de la prise en compte du schéma lors de la génération.

Objectifs

L'objectif principal est d'étudier, de manière plus fine, les limitations des techniques de Constrained Decoding pour la génération de données JSON conformes au schéma fourni en entrée de la génération. Cette étude s'appuiera sur les récents benchmarks proposés dans [2] et [3] mais qui n'étudient pas la corrélation entre la forme des schémas en entrée et le résultat de la génération. Or, notre expérience dans l'analyse de générateurs pour JSON Schema nous pousse à suspecter de limitations liées à la présence de certains opérateurs ou conditions [1].

Le plan de travail suggéré est comme suit:

1. Familiarisation avec le langage JSON Schema [1]
2. Etude approfondie des benchmarks [2] et [3] de générations par des LLMS et des approches sous-jacentes (Guidance et Outlines)
3. Etude de la corrélation entre la la forme des schémas et les résultats obtenus. Une piste envisagée est d'utiliser les métriques de complexité des schémas proposées dans [3]
4. Rédaction d'un rapport synthétique présentant les résultat de l'analyse
5. Proposer des pistes d'améliorations (selon le temps)

Prérequis et attendus

- Python, Intérêt pour l'IA générative
- JSON Schema, Guidance [2], Outlines[3], Function Calling [4], LLMS open-source (Mistral, Llama 3)

[1] `https://json-schema.org`
[2] `https://www.microsoft.com/en-us/research/project/guidance-control-lm-output/`
[3] `https://github.com/outlines-dev/outlines`
[4] `https://platform.openai.com/docs/guides/function-calling`

Contact

Mohamed-Amine Baazizi (mohamed-amine.baazizi@lip6.fr), Nour ElHouda Ben Ali (nour-el-houda.ben-ali@dauphine.eu)

References

[1] Lyes Attouche, Mohamed Amine Baazizi, Dario Colazzo, Giorgio Ghelli, Carlo Sartiani, and Stefanie Scherzinger. Witness generation for JSON schema. *Proc. VLDB Endow.*, 15(13):4002–4014, 2022.

[2] Saibo Geng, Hudson Cooper, Michał Moskal, Samuel Jenkins, Julian Berman, Nathan Ranchin, Robert West, Eric Horvitz, and Harsha Nori. Jsonschemabench: A rigorous benchmark of structured outputs for language models, 2025.

[3] Darren Yow-Bang Wang, Zhengyuan Shen, Soumya Smruti Mishra, Zhichao Xu, Yifei Teng, and Haibo Ding. Slot: Structuring the output of large language models, 2025.