Kairntech à EvalLLM 2024

Hugo Lafayette, Kévin Deturck, Olivier Terrier

Kairntech à EvalLLM 2024

Hugo Lafayette[1], Kévin Deturck[1] Olivier Terrier[1]
(1) Kairntech, 29 chemin du Vieux Chêne, 38240 Meylan, France
hugo.lafayette@kairntech.com, kevin.deturck@kairntech.com, olivier.terrier@kairntech.com

RÉSUMÉ

Trois systèmes pour extraire treize types d'entités nommées ont été développés en se basant uniquement sur un guide d'annotation et cinq documents d'entraînement, sans donnée supplémentaire. Les approches mises en œuvre sont généralistes pour l'annotation d'entités nommées dans un contexte « few-shot ». Les essais sur GPT-4o et Mixtral-8x22B ont montré que ces LLM pouvaient partiellement suivre le guide d'annotation fourni. Cependant, ces systèmes rencontrent des limitations techniques avec des guides complexes. Une configuration classique d'apprentissage automatique, BiLSTM-CRF, a aussi montré son efficacité pour apprendre à partir des exemples fournis en ayant l'avantage d'être particulièrement frugal. Dans cet article, nous présentons d'abord les réflexions préliminaires et les choix généraux effectués pour cadrer le travail, puis nous décrivons chacun des trois runs soumis au challenge avant de conclure par un bilan et des perspectives sur les résultats obtenus.

ABSTRACT

Kairntech at EvalLLM 2024.
Three systems to extract thirteen named entity types based solely on an annotation guide and five training documents, with no additional data, have been developed. The approaches implemented are generalist for annotating named entities in a few-shot context. Tests on GPT-4o and Mixtral-8x22B showed that these LLMs could partially follow the guidelines provided. However, these systems encounter technical limitations with complex guides. A classic machine learning configuration, BiLSTM-CRF, has also shown its effectiveness in learning from the examples provided, with the advantage of being particularly frugal. In this article, we first present the preliminary thoughts and general choices made to frame the work, then we describe each of the three runs submitted to the challenge before concluding with an assessment and outlook on the results obtained.

MOTS-CLÉS : grand modèle de langue, few-shot, REN, apprentissage automatique
KEYWORDS : LLM, few-shot, NER, machine learning

1 Remarques préliminaires

1.1 Approche globale au challenge

EvalLLM 2024 en quelques mots

Le challenge EvalLLM 2024 porte sur l'identification automatique dans des bulletins d'information et des articles de blogs en français de treize types d'entité nommée pour le domaine du renseignement : « PERSON », « UNKNOWN », « FUNCTION », « ORGANISATION », « MILITARY_UNIT », « GROUP », « LOCATION », « SITE », « RESOURCE », « EQUIPMENT », « EVENT », « TIME », « ID ». L'identification doit se faire en contexte *few-shot*, c'est-à-dire avec un guide d'annotation et quelques documents annotés.

Le challenge nous semble en ligne avec nos axes de recherche et de prospection actuels, notamment l'étude des scénarios *few-shot* (Song et al., 2012) dans lesquels peu de données annotées sont disponibles. Par ailleurs, l'évaluation de l'apport pratique des LLM dans différents cas d'usages « industriels » nous intéresse particulièrement.

Une approche polyvalente et reproductible

Conscients qu'il s'agit d'une compétition, nous avons cherché à obtenir des annotations fiables et de qualité. Nous avons cependant privilégié une approche généraliste, simple et reproductible à d'autres cas d'usage plutôt que de chercher la performance sur ce cas d'usage précis. Ce challenge est pour nous l'occasion d'évaluer dans quelle mesure les LLM peuvent permettre d'assister la création de jeux de données dans un contexte *few-shot* (un guide d'annotation, quelques exemples), plutôt que de produire un système complexe et performant mais inadapté dans un autre contexte, même similaire.

Gestion de la discontinuité

Nous avons choisi d'ignorer les cas de discontinuité car l'effort à produire nous semblait trop important au regard des gains attendus. En plus, les exemples fournis sont trop peu nombreux pour bien comprendre les différents enjeux liés à cette problématique. Nous avons choisi de nous concentrer sur une approche « naïve » qui n'annote que l'entité englobante. Nous pensons qu'une solution possible est un post-traitement dédié à cette tâche, sans doute en exploitant une analyse morpho-syntaxique du fragment de texte concerné (via Spacy[1] par exemple).

1.2 Exploitation du guide d'annotation

En contexte *few-shot*, le guide d'annotation est la ressource que nous avons essentiellement utilisée pour les runs LLM à travers du *prompting* (Arora et al., 2022). Notre démarche a été de fournir dans le prompt les différentes indications contenues dans le guide, notamment les définitions des catégories d'annotation et les exemples. La problématique générale a été de passer du document fourni en PDF-image à un prompt efficace pour un LLM.

Extraction du contenu

L'exploitation du guide d'annotation fourni était difficile car tout le contenu, même textuel, était sous la forme d'images tandis que nous avions besoin de texte pour l'utilisation dans les prompts. Nous avons procédé à la conversion du document PDF-image dans un format texte avec l'outil

[1] https://spacy.io/

Adobe Acrobat[2]. Restaient cependant les exemples rares donc précieux sous la forme de schémas, pas directement exploitables, que nous avons dû manuellement transcrire en texte dans les prompts. Le guide d'annotation contenait aussi des images d'arbres de décision, un format potentiellement pratique pour un humain mais pas exploitable directement dans un prompt. La conversion de ces schémas en texte pour le prompt nous a semblé périlleuse, nécessitant potentiellement plusieurs niveaux d'imbrications (*Tree-of-Thoughts* ?). C'est pourquoi, dans le temps imparti, nous avons préféré nous focaliser sur les instructions nativement textuelles.

Prompting

Après l'extraction de contenu depuis le guide fourni, nous avons fait face à la difficulté d'en faire un prompt qui soit propice à maximiser les performances des LLM. Notamment, le guide est avant tout rédigé à destination d'humains et donc pas nécessairement adéquat pour produire un prompt efficace. Les indications sont exprimées avec des phrases complexes, parfois techniques. L'ensemble est souvent morcelé, avec des indications ou des exemples concernant une certaine étiquette parfois situés à plusieurs endroits dans le document et des renvois à d'autres paragraphes. Nous décrivons notre construction du prompt dans la section consacrée au run 1. C'est une première approche ouvrant la perspective d'approfondir les bonnes pratiques en matière de *prompting*, notamment sur le caractère explicite des instructions pour un LLM (Zamfirescu-Pereira et al., 2023).

1.3 Constitution d'un corpus de référence

Pour nous permettre d'évaluer comparativement et d'améliorer la qualité de nos trois systèmes, nous avons choisi de constituer un petit corpus de référence *ad hoc*, équivalent à un corpus de développement dans un processus classique d'apprentissage automatique.

Choix des exemples

Face à la nécessité de procéder rapidement et par annotation manuelle, l'objectif n'était pas de constituer un corpus d'évaluation significatif en taille mais d'avoir un moyen d'évaluer nos systèmes au fil du développement. Aussi, nous avons fait en sorte d'avoir une certaine représentation de toutes les catégories d'annotation visées. Nous avons finalement sélectionné un ensemble de onze phrases représentatives faisant partie des 24 documents proposés par l'organisation de l'atelier. L'objectif dès lors était de les annoter pour en faire un petit corpus de référence.

Annotation

Les trois membres de l'équipe ont travaillé sur l'annotation de façon à maximiser l'assurance d'avoir des annotations de qualité, c'est-à-dire en accord avec le guide d'annotation fourni. Tous les trois ont travaillé ensemble en se reportant systématiquement aux indications du guide d'annotation en cas de doute. Ce travail nous a permis de constater que la tâche d'annotation était loin d'être évidente, même pour trois humains travaillant communément. Ce fut parfois l'occasion de discussions autour du guide d'annotation pour déterminer la bonne annotation. Nous avons finalement traité l'ensemble des onze documents en à peu près 25 minutes à trois.

[2] https://www.adobe.com/fr/acrobat.html

2 Description de l'approche LLM GPT-4o (run 1)

L'objectif de ce run est l'exploitation du guide d'annotation et des exemples fournis via la construction d'un prompt destiné à l'extraction des entités nommées en une seule passe avec un LLM, en l'occurrence GPT-4o[3]. Nous avons choisi la variante « o » de GPT-4 car nos essais préliminaires ont montré que les résultats étaient meilleurs avec un coût de traitement moindre, à la fois sur les aspects du temps de traitement, de l'impact énergétique et du prix.

2.1 Pré-traitement et *prompting*

Chaque document fourni a d'abord été découpé en *chunks*, souvent équivalents à une phrase ou un paragraphe, en utilisant la librairie Blingfire[4]. Nous avons constaté que d'utiliser cette plus petite unité de traitement permettait d'obtenir une bien meilleure qualité d'annotation que si nous restions sur un traitement par document. En revanche, chaque traitement étant plus petit cela multiplie d'autant leur nombre et les appels au moteur GPT-4o. Or chaque appel peut être particulièrement coûteux puisque chacun à requiert le traitement par le LLM du prompt entier.

Pour chacun des *chunks*, nous utilisons un prompt dont la composition est décrite ci-dessous.

1. Une instruction générale (« system message ») liée au rôle attendu du LLM, ici un expert en extraction d'entités nommées, suivie par un résumé de la tâche, notamment la liste des treize étiquettes attendues.
2. L'intégralité du guide d'annotation fourni avec deux exceptions : les exemples ont été rassemblés (cf. point 3) et les images d'arbre de décision n'ont pas été utilisées. Nous avons gardé la formulation choisie, même si quelques changements mineurs ont été réalisés, notamment la suppression des renvois à des paragraphes. Nous avons également exclu la partie sur les entités discontinues car nous avons constaté que cela conférait de légers gains en qualité.
3. Une succession d'exemples de paragraphes avec la sortie souhaitée dans le format attendu. Ces exemples sont à la fois issus directement du guide d'annotation et des documents d'exemple fournis. L'ensemble constitue un corpus d'une centaine d'exemples (des paragraphes).
4. Le *chunk* à traiter.

L'ensemble du prompt fourni au LLM était finalement assez long, de l'ordre de 6 000 mots en fonction de la taille du *chunk* ou paragraphe à traiter. Cette approche est gourmande en temps de calcul car pour chacun des 216 *chunks* de texte à traiter, nous fournissons le guide d'annotation complet, soit environ 6 000 mots. Cela implique au total environ 1 300 000 mots à traiter (soit 2 000 000 de « tokens » pour le LLM selon nos estimations). Avec GPT-4o, le temps de traitement des 24 documents est de l'ordre de 8 min environ et un coût d'environ 10 $. Nous estimons que l'entraînement produit entre 7 000 t et 15 000 t de CO2e et l'inférence environ 6 kg de CO2e.

[3] https://openai.com/index/hello-gpt-4o/
[4] https://github.com/microsoft/BlingFire

2.2 Annotation et post-traitement

Pour chaque paragraphe, nous récupérons le résultat produit par le LLM. Ce résultat brut est ensuite analysé pour retrouver les fragments de texte annotés et pouvoir créer les spans afférents. Nous donnons ci-dessous un exemple de paragraphe suivi de la sortie brute (*en italique*).

« Les pertes de la coalition en Afghanistan (mais pas celles de l'armée nationale afghane) peuvent ainsi se réduire mais pas autant qu'espéré et sans que le nombre d'incidents diminue par ailleurs. »

ORGANIZATION:afghane

MILITARY_UNIT:l'armée nationale afghane|la coalition

LOCATION:en Afghanistan

EVENT:Les pertes|incidents|réduire|diminue

Le traitement ici s'applique à retrouver les frontières des entités nommées dans le texte d'origine pour le transformer dans le format attendu du challenge. Le traitement est approximatif : un fragment de texte apparaissant plusieurs fois dans le paragraphe étudié sera traité comme annoté pour chacune de ses occurrences. En pratique cependant, ce cas nous semble relativement rare. Une fois les paragraphes correctement annotés, nous agrégeons simplement les résultats (avec une gestion d'*offset*) de l'ensemble des paragraphes du document pour obtenir le résultat final.

3 Description de l'approche LLM Mixtral-8x22B (run 3)

Le run 3, comme le run 1, consiste en l'exploitation du guide d'annotation et des exemples fournis par du *prompting* de LLM pour l'extraction des entités nommées en une seule passe. La différence est ici le LLM utilisé, en l'occurrence Mixtral-8x22B-Instruct-v0.1[5]. Nous avons repris le protocole utilisé pour le run 1, avec les mêmes étapes (la segmentation, le *prompting*, l'annotation et le post-traitement). Nous avons déployé le LLM sur une machine dédiée avec un GPU A100 le temps du run. Il a fallu 27 min pour traiter les 216 *chunks* de texte et un coût de l'ordre de 2 $. Nous estimons que l'inférence produit 100g de de CO2e.

Fine-tuning

Nous avions aussi essayé d'effectuer un fine-tuning de Mixtral-8x7B-Instruct-v0.1 avec le jeu de données à notre disposition mais nous avons constaté que les performances obtenues ainsi sur le jeu de « dev » sont restées inférieures à celles d'un Mixtral-8x22B-Instruct-v0.1 « vanilla ». Nous avons abandonné cette version et sommes restés sur un moteur Mixtral-8x22B-Instruct-v0.1.

« Hallucination »

À l'issue des expérimentations et du run final, des « hallucinations » sont apparues, notamment sur les étiquettes utilisées : en plus des étiquettes permises par le guide d'annotation, une étiquette nommée « SOURCE » a été ajoutée. Ce phénomène est toutefois bien plus marginal qu'avec les autres moteurs envisagés dans un premier temps, notamment Mixtral-8x7B et DBRX, mais nécessite

[5] https://huggingface.co/mistralai/Mixtral-8x22B-Instruct-v0.1

un mécanisme de contrôle. Rien n'assure que GPT-4o soit exempt de telles hallucinations mais nous n'en avons pas constaté durant nos différents runs.

4 Description de l'approche classique Bi-LSTM-CRF (run 2)

Le run 2 consiste en l'entraînement *few-shot* d'un double modèle Bi-LSTM-CRF avec des *embeddings* contextuels sur les exemples contenus dans les documents fournis et ceux présents dans le guide d'annotation. Le tout forme un corpus d'une centaine d'exemples. Par la nature du format d'entraînement (IOB) de ce genre d'architecture, chaque token ne peut appartenir qu'à une étiquette maximum. Cela ne permet pas nativement la possibilité d'entités imbriquées, comme dans l'exemple ci-dessous :

[L'ambassadeur [américain]_ORGANIZATION [à Séoul]_LOCATION]_FUNCTION a [pris la parole]_EVENT sur la situation.

C'est pourquoi nous avons expérimenté un système impliquant deux modèles utilisés en cascade, à même de pouvoir produire des annotations imbriquées.

4.1 Pré-traitement

Un script python dédié nous permet de séparer le corpus en deux sous-corpus selon la procédure décrite ci-dessous.

1. Dans le corpus principal appelé « Longest Match », nous ne conservons que les annotations de premier niveau en supprimant toutes les annotations imbriquées.

2. Quand ces annotations « Longest Match » contiennent des annotations imbriquées, nous générons un segment dans un corpus secondaire appelé « Cascade » qui reprend le texte couvert par l'annotation de premier niveau en laissant cette fois-ci uniquement les sous-annotations.

Si nous reprenons l'exemple cité précédemment :

[L'ambassadeur [américain]_ORGANIZATION [à Séoul]_LOCATION]_FUNCTION a [pris la parole]_EVENT sur la situation,

nous garderons uniquement dans le corpus « Longest Match » le segment : [L'ambassadeur américain à Séoul]_FUNCTION a [pris la parole]_EVENT sur la situation

et nous génèrerons dans le corpus « Cascade » le segment :

L'ambassadeur [américain]_ORGANIZATION [à Séoul]_LOCATION.

4.2 Modèles et pipeline

À partir des deux sous-corpus « Longest Match » et « Cascade », nous avons entraîné deux modèles Bi-LSTM-CRF que l'on compose ensuite dans un pipeline qui va successivement :

1. appliquer le modèle « Longest Match » sur l'ensemble du document d'entrée,
2. appliquer le modèle « Cascade » uniquement sur le texte des annotations trouvées précédemment.

De cette façon nous avons pu simuler deux niveaux d'imbrication, ce qui nous semble suffisant pour traiter l'ensemble des exemples fournis dans le cadre du challenge.

Il a fallu 4 min pour traiter les 24 documents sur un GPU de petite capacité (GeForce GTX 1080 Ti) avec un coût d'opération négligeable. Nous estimons que l'entraînement produit 23 g de CO2e et l'inférence 3 g de CO2e.

Il est à noter que cette approche plus classique et plus frugale permet également un étalonnage des performances obtenues avec des LLM. Par ailleurs, au fur et à mesure que le corpus disponible s'enrichit de nouveaux exemples, cette approche peut en bénéficier pleinement pour améliorer la qualité de ses prédictions. Au contraire, un système à base de *prompting* LLM aura selon toute vraisemblance plus de mal à exploiter ces nouvelles données à cause notamment de la taille maximale du prompt, du rendement décroissant en complexifiant les instructions et d'une latence augmentée. C'est pourquoi nous avons choisi cette approche complémentaire pour le run 2.

5 Résultats

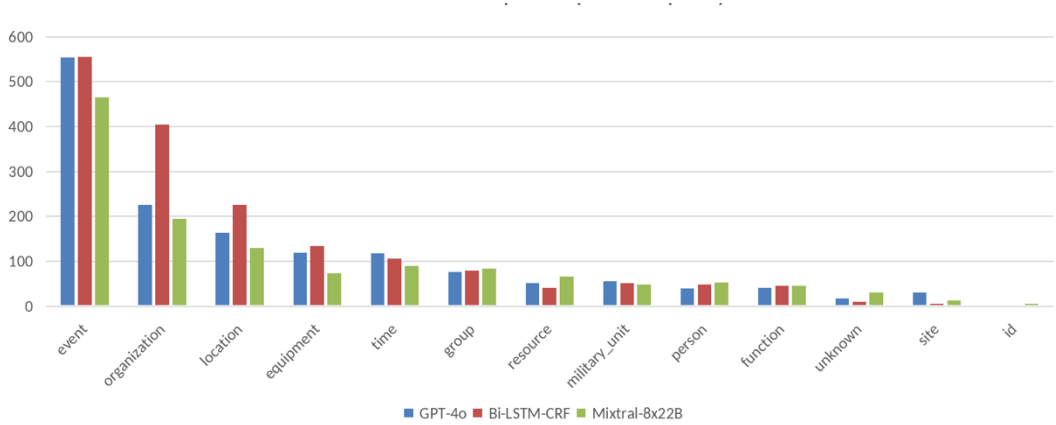

FIGURE 1 : Nombre d'annotations par Étiquette et par système

Nous présentons en figure 1 le nombre d'annotations produites par étiquette et par système sur les 24 documents de développement. Le corpus proposé pour le challenge, tout du moins dans cette partition de développement, semble comporter une distribution hétérogène des étiquettes.

Cette différence de représentation des étiquettes en corpus constitue un biais pour l'apprentissage d'un système comme le Bi-LSTM-CRF puisque la qualité des résultats est notamment corrélée au nombre d'exemples fournis. Concernant les LLM, une quantité d'exemples moindre entraîne potentiellement plus de difficulté dans la compréhension du prompt et aboutisse ainsi à des résultats de moins bonne qualité.

Le classement des étiquettes en nombre d'annotations est plutôt similaire entre les systèmes, indiquant une certaine homogénéité entre les approches dans la prise en compte des différents types d'entités. Les quelques différences concernent majoritairement des étiquettes relativement peu représentées, avec tout de même le cas de la quantité beaucoup plus élevée d'annotations « Organisation » par le Bi-LSTM-CRF, peut-être liée à une quantité d'exemples fournis plus élevée.

GPT-4o	Mixtral-8x22B	Bi-LSTM-CRF
Tous les 4 tokens	Tous les 7 tokens	Tous les 5 tokens

TABLE 1 : Fréquence d'annotation par système en nombre de tokens

Nous présentons dans le tableau 1 la fréquence d'annotation par nombre de tokens dans les 24 documents de développement selon le système utilisé. Quel que soit le système utilisé, la densité d'annotation est élevée, avec une annotation tous les 4 à 7 tokens. Cela semble montrer que le corpus à traiter comporte beaucoup d'entités à identifier, induisant un traitement exigeant par les systèmes avec en plus un grand nombre d'étiquettes. Nous n'observons pas de différence importante de fréquence d'annotation entre les systèmes et en particulier nous n'observons pas de tendance commune aux LLM, au contraire, la différence est plus marquée entre GPT-4o et Mixtral-8x22B, montrant que le choix du LLM est impactant.

Run	F1		Précision		Rappel	
	Macro	Micro	Macro	Micro	Macro	Micro
GPT-4o	**57**	54,98	**64,35**	**62,86**	**51,68**	48,86
Mixtral-8x22B	38,06	37,97	43,16	47,38	35,89	31,68
Bi-LSTM-CRF	45,14	**55,75**	51,26	60,84	40,98	**51,45**

TABLE 2 : Performances par système sur toutes les étiquettes

Le tableau 2 contient les résultats de performance par système sur toutes les étiquettes. En majorité, les meilleures performances (en gras) sont obtenues avec GPT-4o. Cependant, le fait que le système Bi-LSTM-CRF arrive à surpasser la performance de GPT-4o en F1 et en rappel avec des formules « micro » est particulièrement intéressant. Les formules « micro » ont la particularité de tenir compte de la quantité d'annotations par étiquette pour pondérer les scores moyens. Or nous avons précédemment observé que le corpus avait une distribution des étiquettes hétérogène. La sensibilité au nombre d'exemples en corpus du système classique Bi-LSTM-CRF explique sa performance plus élevée avec des métriques « micro ».

Étiquette	GPT-4o	Mixtral-8x22B	Bi-LSTM-CRF
F1_PERSON	**84,4**	76,9	**84,4**
F1_TIME	**69,2**	46,3	60,3
F1_ID	**100**	75	0
F1_GROUP	**60**	49,3	53,9
F1_EVENT	53,9	40	**62,2**
F1_ORGANIZATION	55,2	39,3	**59,7**
F1_LOCATION	**57,4**	44,7	52,1
F1_MILITARY_UNIT	**59,5**	15,5	55
F1_EQUIPMENT	**51,1**	27,5	48,5
F1_FUNCTION	**49,3**	27,8	38,6
F1_UNKNOWN	**40**	17,4	36,4
F1_RESOURCE	**32,9**	21,2	27,7

TABLE 3 : Scores F1 par étiquette et par système

Dans le tableau 3, nous donnons les scores F1 par étiquette et par système, avec en gras les valeurs les plus élevées par étiquette. En préambule, c'est important de relativiser la significativité de ces scores avec le nombre d'exemples traités par étiquette (cf. Figure 2). Notamment, le score de l'étiquette « ID » ne repose que sur quelques exemples. Comme avec les scores globaux, la majorité des étiquettes est mieux détectée avec GPT-4o. Certaines étiquettes sont aussi bien ou mieux détectées avec le Bi-LSTM-CRF. En lien avec les meilleures performances en « micro », cela semble être dû à une quantité d'exemples élevée, notamment « EVENT ». Au contraire, l'étiquette « ID » représentée que par quelques exemples n'est pas bien traitée par ce système. Sur un autre aspect, la sémantique des étiquettes, comme « ID », hétéroclite, peut aussi constituer une difficulté.

Run	GPT-4o	Mixtral-8x22B	Bi-LSTM-CRF
Nbre d'annotations	**1497**	833	1152
Temps de traitement (s)	480	1620	**240**
Empreinte carbone de l'inférence (g CO2e)	6000	100	**3**
Prix du traitement ($)	10	2	**0,2**

TABLE 4 : Données sur les coûts de traitement par système

Nous présentons dans le tableau 4 des données sur les coûts de traitement par système selon le nombre d'annotations produites. Notons en préambule que l'empreinte carbone associée aux LLM ne porte que sur l'inférence associée au traitement et n'inclut pas l'entraînement, beaucoup plus impactant. Les approches avec LLM sont beaucoup plus coûteuses sur tous les plans à quantités d'annotations comparables. L'empreinte carbone est jusqu'à 2000 fois plus élevée avec GPT-4o face

au Bi-LSTM-CRF, le prix du traitement jusqu'à 50 fois plus élevé, enfin, le temps de traitement est presque sept fois plus élevé avec Mixtral-8x22B face au Bi-LSTM-CRF.

6 Conclusion

Nous avons mis en place trois systèmes capables d'extraire treize types d'entités nommées à partir d'un guide d'annotation fourni et de cinq documents d'entraînement. Aucune autre donnée n'a été utilisée, que ce soit des documents additionnels, des lexiques ou des bases de connaissances. Nous avons choisi des approches généralistes et relativement simples dans leur implémentation que nous pensons applicables à toute annotation d'entités nommées en conditions *few-shot*.

L'approche LLM GPT-4o a donné majoritairement les résultats de meilleure qualité. Nous avons aussi montré qu'un système d'apprentissage automatique plus classique, en l'occurrence un BiLSTM-CRF avec *embeddings* contextuels, pouvait apprendre sur la base de la centaine d'exemples fournis et constituer une alternative crédible bien plus frugale que l'approche proposée par les LLM, avec des résultats meilleurs que GPT-4o sur des étiquettes plus représentées en corpus. Au contraire, l'approche LLM avec Mixtral-8x22B, en étant coûteuse, a donné des résultats décevants. Les résultats entre GPT-4o et le Bi-LSTM-CRF posent la question du rapport entre le coût et la qualité produite ou attendue pour faire un choix d'approche.

De manière générale, pour favoriser une bonne exploitation d'un guide d'annotation par un LLM, cela nous semble indispensable de créer un ensemble de bonnes pratiques pour sa rédaction. Notamment, nous pensons que c'est essentiel de privilégier le texte aux images et de fournir un effort particulier pour que la structure et la rédaction du guide soient explicites au maximum, sans renvoi à d'autres paragraphes quand cela est possible et avec une ambiguïté minimale.

Références

ARORA, S., NARAYAN, A., CHEN, M. F., ORR, L., GUHA, N., BHATIA, K., ... & RE, C. (2022). Ask me anything: A simple strategy for prompting language models. In *The Eleventh International Conference on Learning Representations*.

SONG N., WANG T., CAI P., MONDAL, S. K. & SAHOO J. P. (2023). A comprehensive survey of few-shot learning: Evolution, applications, challenges, and opportunities. *ACM Computing Surveys*, *55*(13s), 1-40.

ZAMFIRESCU-PEREIRA, J. D., WONG, R. Y., HARTMANN, B., & YANG, Q. (2023). Why Johnny can't prompt: how non-AI experts try (and fail) to design LLM prompts. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (pp. 1-21).