Augmentation des données par LLM pour améliorer la détection automatique des erreurs de coordination

Chunxiao Yan[1] Iris Eshkol-Taravella[1] Sarah De Vogué[1] Marianne Desmets[2]
(1) Laboratoire MoDyCo, Nanterre, France
(2) Laboratoire de Linguistique Formelle, Paris, France
chunxiao.y@parisnanterre.fr, ieshkolt@parisnanterre.fr

RÉSUMÉ

Afin d'améliorer les performances d'un outil de détection automatique des erreurs de coordination, cette étude explore l'utilisation de grands modèles de langage (LLM) pour remédier au déséquilibre des classes et à la limitation des données. En générant des phrases erronées simulées par un LLM pour former un corpus synthétique, nous améliorons la détection d'une classe sous-représentée ainsi que les performances globales du modèle. Nous étudions également l'application des LLM à l'annotation des données, avec pour objectif d'intégrer ces annotations à l'entraînement afin d'optimiser l'apprentissage du modèle.

ABSTRACT

Data augmentation using LLMs to improve automatic detection of coordination errors

In order to improve the performance of an automatic coordination-error detection tool, this study investigates the use of large language models (LLMs) to address class imbalance and data scarcity. By using an LLM to generate erroneous sentences and build a synthetic corpus, we enhance the detection of an under-represented class as well as the model's overall performance. We also explore applying LLMs to data annotation, with the goal of incorporating these annotations into training to further improve the model's learning.

MOTS-CLÉS : erreur de coordination, apprentissage profond, corpus synthétique, LLM.

KEYWORDS: coordination error, deep learning, synthetic corpus, LLM.

1 Introduction

Dans le cadre du projet écri+ (PIA n°ANR-17-NCUN-0015), cette étude vise à améliorer un outil de détection automatique des erreurs de coordination dans les rédactions d'étudiants universitaires. L'outil identifie ces erreurs et fournit des messages pour guider les étudiants dans leur correction, améliorant ainsi leurs compétences rédactionnelles.

Les travaux existants (Noreskal, 2022; Noreskal *et al.*, 2023) fournissent une base théorique et technique solide pour le développement de cet outil. Dans ces travaux, des approches classiques d'apprentissage automatique ainsi que d'apprentissage profond basé sur Camembert (Martin *et al.*, 2020) et sur Flaubert (Le *et al.*, 2019), ont été explorées afin d'obtenir des modèles capables de détecter automatiquement ces erreurs. Il s'agissait de deux types de tâches d'entraînement : une tâche de classification binaire visant à déterminer si une phrase contient une erreur de coordination, et une

tâche de classification multi-label destinée à identifier précisément le type d'erreur. Les résultats de Noreskal *et al.* (2023) ont montré que le modèle de classification binaire basé sur l'apprentissage profond a obtenu des performances satisfaisantes. Cependant, des améliorations significatives restent nécessaires pour la classification multi-label.

Les problèmes liés à la carence de données et au déséquilibre des catégories constituent des défis majeurs pour l'optimisation des modèles. Ces dernières années, de grands modèles de langue (angl., *large language model* - LLM, comme GPT-4, LLaMA, etc.) ont connu un grand succès. Nous estimons qu'ils pourraient offrir de nouvelles perspectives pour surmonter ces difficultés. D'une part, des travaux ont déjà été menés sur la construction de corpus synthétiques à l'aide de la génération de textes par des LLM (Whitehouse *et al.*, 2023), notamment utilisée dans la classification de textes (Li *et al.*, 2023). D'autre part, les LLM sont également utilisés pour l'annotation automatique des données (Meoni *et al.*, 2023). Cette approche permet d'enrichir les corpus existants tout en réduisant les coûts de main-d'œuvre associés à l'annotation manuelle.

Le travail présenté dans cet article couvre deux directions. La première consiste à augmenter les données en utilisant un LLM pour générer des phrases avec des types d'erreurs spécifiés, afin d'entraîner des modèles fondés sur CamemBERT. L'autre direction concerne une première exploration de l'utilisation des LLM pour la tâche d'annotation de corpus.

Nos tâches ont ceci de particulier qu'elles visent des données ayant des caractéristiques syntaxiques spécifiques. En effet, les travaux sur l'augmentation des données pour des tâches spécifiques à la syntaxe sont moins nombreux et nécessitent une mise à jour. En plus d'améliorer les performances de nos propres modèles, nous espérons également explorer les capacités des LLM pour générer et identifier des erreurs syntaxiques spécifiques.

2 Erreur de coordination

D'après Abeillé & Godard (2021), la coordination établit une relation entre au moins deux mots, deux syntagmes ou deux séquences de mots, qui reçoivent la même fonction. Dans ce travail, nous nous intéressons plus précisément aux coordinations introduites par une conjonction de coordination (par exemple, « et », « ou », « car », etc.), aux structures coordonnées reliées par un adverbe de liaison, tels que « ensuite » ou « donc », ainsi qu'à la juxtaposition.

Dans le cadre de leur formation, les étudiants sont amenés à produire certains types d'écrits, dont les phrases visent à expliciter un sujet. Or, ce contexte rédactionnel est propice à l'émergence de différents problèmes syntaxiques relatifs à la coordination. Ainsi, Noreskal (2022) relève que les erreurs de coordination sont fréquentes dans des phrases longues et complexes. Les erreurs de coordination peuvent affecter la grammaticalité de la phrase, mais génèrent aussi des ambiguïtés, ou des incertitudes, entraînant des difficultés de lisibilité et de compréhension. L'auteur a analysé les erreurs et proposé une typologie, divisant les erreurs de coordination en neuf types.

L'exemple 1 appartient au type **mauvaise cohérence des groupes syntaxiques**. Ce type d'erreur survient lorsqu'on observe que les éléments coordonnés ne respectent pas le parallélisme. On constate que le verbe « parler » n'est pas compatible avec « que » dans la phrase.

L'exemple 2 appartient aux types **structure lourde** et **absence de ponctuation**. La phrase est lourdement chargée d'informations, ce qui nuit à sa lisibilité, et manque de ponctuation pour la

segmentation.

1. *Il ne me parle pas d'un divorce en particulier, mais que l'action de divorcer était interdite.*
2. *Dans cette caricature il y a deux hommes assis autour d'une table, une personne du nord-est représentée en blanc, gros riche qui est assis sur un coffre-fort en bas de lui il y a des lingots d'or des billets des bijoux et de la bière sur son côté de la table il y a beaucoup de nourriture des beaux plats.*

En outre, les autres types sont les suivants : erreur de préposition dans les conjoints sauf le premier (remplacement d'une préposition par une autre, préposition non attendue et absence de préposition), grande distance entre conjoints, mauvais accord sujet-verbe et autre (cas particuliers).

3 Données

3.1 Révision des données

Il est nécessaire de réviser à nouveau nos données avant de procéder à l'augmentation et aux expériences d'entraînement, pour au moins deux raisons. Premièrement, certaines catégories sont limitées à un très petit nombre de phrases. Cela conduit à un entraînement moins bon pour la classification multi-label. Deuxièmement, il existe certaines phrases erronées dont la cause de l'erreur n'est pas due à la coordination. Ces phrases provoquent du bruit dans l'apprentissage, ce qui affecte l'apprentissage du modèle.

Grâce à l'analyse de deux experts linguistiques, des phrases erronées ont été revérifiées et les types d'erreur ont été répartis en trois grandes catégories pour la classification multi-label :

- **Mauvaise cohérence des groupes syntaxiques (MCGS)** est élargie en ajoutant les deux types d'erreurs de préposition (remplacement d'une préposition par une autre et préposition non attendue). Ce type d'erreur concerne la forme syntaxique de la phrase. Elle résulte directement du fait que la structure coordonnée n'est pas grammaticale.
- **Problème de rattachement** comprend les catégories structure lourde, absence de préposition et grande distance entre conjoints. Ce type d'erreurs affecte la lisibilité de la phrase, mais la structure reste correcte au niveau grammatical. Elles présentent en particulier deux caractéristiques :
  - Complexité : l'imbrication excessive de propositions (subordonnées, relatives, etc.) dans la structure coordonnée. Cela nécessite un effort cognitif important pour identifier le rattachement du conjoint suivant.
  - Ambiguïté : les constituants coordonnés pourraient être rattachés à différentes têtes syntaxiques induisant différentes significations.
- **Problème de ponctuation** concerne la ponctuation manquante ou abusive et d'autres problèmes de ponctuation pour la coordination et pour la juxtaposition.

L'ensemble de données révisé se compose de 669 phrases erronées, avec 128 erreurs de MCGS, 292 erreurs liées à un problème de rattachement et 300 erreurs liées à un problème de ponctuation. Il comprend également des erreurs moins fréquentes : les mauvais accords sujet-verbe (21 occurrences) ou les erreurs dans le choix du coordonnant (44 occurrences). Ces dernières ne seront utilisées que dans la tâche binaire (voir la section 4.1.1) et ne seront pas utilisées dans la classification multi-label (voir la section 4.1.2) en raison de leur rareté.

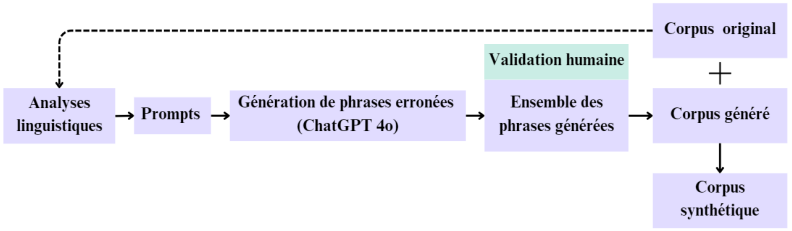

FIGURE 1 – Processus du travail de l'augmentation des données

Rôle du système	*Vous êtes un assistant chargé de générer des phrases en français contenant des erreurs spécifiques de coordination.*
Message principal	*Générez 10 phrases érronées en français qui répondent aux critères suivants :* *## Description d'erreur liée à la coordination :* **[description d'erreur avec un exemple]** *## Les phrases doivent varier en longueur de 10 à 40 mots.* *## Les phrases doivent aborder les thèmes de [Thème 1], [Thème 2], [Thème 3]...* *Fournissez la phrase incorrecte dans le format suivant : Phrase incorrecte ici*

FIGURE 2 – Prompt pour génération des phrases erronées

3.2 Augmentation des données

Pour équilibrer les nombres des trois grandes catégories, il nous faut plus de phrases ayant MCGS. Les MCGS sont de graves erreurs de structure et notre outil doit être très performant pour les détecter. Ainsi, nous essayons d'augmenter les données de la classe MCGS à l'aide des LLM. De nombreux travaux ont tenté d'utiliser les LLM dans l'augmentation des données afin de construire des corpus synthétiques (Zhou *et al.*, 2024). Concernant notre travail, l'idée est d'utiliser le modèle GPT 4o de OpenAI pour produire des phrases en imitant des erreurs réelles.

La figure 1 montre le processus du travail d'augmentation. Tout d'abord, le linguiste analyse des phrases MCGS dans le corpus original et élabore une description précise de l'erreur pour chaque exemple sélectionné, afin d'établir des prompts. Nous soutenons que les phrases générées doivent à la fois maintenir la similarité du sujet avec le corpus original et rester aussi diversifiées que possible sur le plan lexical. Selon le travail de Li *et al.* (2024), les auteurs soulignent l'importance de générer des sujets ou des descriptions étroitement liés au cas d'utilisation cible. Les auteurs soutiennent que les données synthétiques sont plus étroitement alignées sur la distribution réelle, améliorant ainsi les performances de la tâche de classification. Ainsi, nous avons choisi 300 sujets dans le domaine des sciences humaines et sociales et avons aléatoirement pris 3 à 5 d'entre eux à chaque fois pour générer les phrases erronées. Cela permet d'avoir un haut degré de contrôle sur la génération.

Une fois les prompts (voir la figure 2) préparés, nous utilisons le modèle GPT 4o pour effectuer de la génération. Au total, 25 prompts contenant différents sous-types d'erreurs MCGS sont utilisés. 250 phrases générées par le modèle sont obtenues. Ensuite, des validations manuelles ont été effectuées sur les phrases générées, principalement pour vérifier si les phrases erronées correspondaient à la description.

Enfin, 164 phrases MCGS sont intégrées au corpus original pour former un corpus synthétique (voir annexe A1 pour des exemples). Le nombre d'erreurs dans chaque catégorie principale du corpus synthétique est le suivant : 292 erreurs de MCGS, 292 erreurs liées à un problème de rattachement

et 300 erreurs liées à un problème de ponctuation. Le corpus synthétique présente une quantité équilibrée d'erreurs dans les trois catégories. Dans la section suivante, nous l'utiliserons dans la tâche de classification multi-label.

4 Prédiction automatique

Dans cette section, nous menons une série d'expériences de prédiction automatique articulées autour de deux approches. La première (Section 4.1), fondée sur CamemBERT, cherche à mesurer les gains apportés par l'augmentation ciblée des données dans la classification multi-label visant à identifier le type précis d'erreur. La seconde (Section 4.2) explore les capacités des LLM pour annoter les étiquettes directement, en variant la taille des modèles et le nombre d'exemples fournis. Pour chaque approche, nous présentons d'abord la méthodologie, puis les résultats obtenus. Enfin, nous discutons ces résultats dans la Section 4.3.

4.1 Classification fondée sur CamemBERT

4.1.1 Classification binaire

Afin d'évaluer l'impact de la révision du corpus sur les performances du modèle, nous avons entraîné un modèle binaire en utilisant le corpus révisé, sans augmentation des données. L'ensemble de données comporte un nombre équilibré de phrases correctes et incorrectes (669 de chaque). Les données ont été divisées en 80 % pour l'entraînement (train), 10 % pour le développement (dev) et 10 % pour le test, avec un échantillonnage stratifié selon les types d'erreurs.

Dans les travaux de Noreskal *et al.* (2023), le modèle binaire atteignait une exactitude de 0,77. Comme le montre le tableau 1, notre modèle, entraîné sur le corpus révisé, a obtenu une exactitude de 0,80 sur l'ensemble de développement et 0,79 sur l'ensemble de test. De plus, les performances sur dev et sur test restent stables, ce qui témoigne de la capacité de généralisation du modèle.

Catégorie	Train	Dev	Test	F1 score dev	F1 score test	Exactitude dev	Exactitude test
Correcte	535	67	67	0,80	0,79		
Erronée	535	67	67	0,79	0,79		
						0,80	0,79

TABLE 1 – Résultats de la classification binaire basée sur CamemBERT

4.1.2 Classification multi-label à l'aide de corpus synthétiques

Notre travail se concentre principalement sur l'amélioration des performances de la classification multi-label. Pour ce faire, nous avons augmenté les données (3.2) afin de parvenir à un équilibre entre les trois grands types d'erreurs. Trois expériences sont réalisées : une expérience utilisant le corpus original révisé (CO) et deux expériences utilisant des corpus synthétiques (CS1 et CS2). Nous avons appliqué la même méthode d'échantillonnage stratifié pour diviser chaque ensemble de données selon

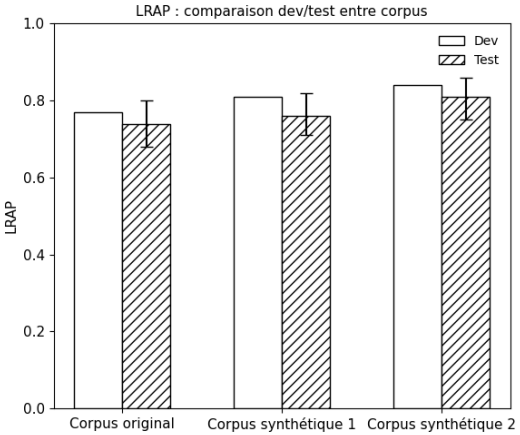

FIGURE 3 – Résultats LRAP avec intervalles de confiance à 95 % (1 000 échantillons bootstrap)

une répartition de 80 % pour l'entraînement (train), 10 % pour le développement (dev) et 10 % pour le test (voir les détails dans le tableau 2) .

Catégorie	Train CO	Train CS1	Train CS2	Dev CO	Dev CS1, CS2	Test CO	Test CS1, CS2
MCGS	102	**232**	**232**	12	30	14	30
Rattachement	241	241	241	30	30	29	29
Ponctuation	233	233	233	30	30	29	29
Correcte	241	241	**576**	30	30	30	30

TABLE 2 – Distribution des phrases par type

Pour l'expérience CS1, la catégorie Correcte est équilibrée en termes de quantité avec les trois autres catégories (MCGS, Ponctuation, Rattachement). On espère que cet équilibre permettra d'obtenir des performances similaires pour chaque catégorie lors de l'entraînement. Étant donné que nous disposons d'un grand nombre de phrases correctes dans le projet, le CS2 a uniquement ajusté le nombre d'exemples de la catégorie Correcte dans l'ensemble d'entraînement, en le fixant à la somme des exemples réels des trois autres catégories (soit 576 phrases). Cette modification vise à fournir davantage d'informations sur la catégorie Correcte afin d'observer si cela peut améliorer les performances du modèle.

La figure 3 illustre les résultats de la précision moyenne du classement des étiquettes (LRAP, pour *Label Ranking Average Precision*) sur l'ensemble de développement (Dev) et le test final pour les trois expériences. Nous observons que CS2 obtient le score le plus élevé, avec $0{,}84$ sur Dev et $0{,}81$ sur Test. CS1 suit avec des scores de $0{,}81$ sur Dev et $0{,}76$ sur Test. Comparé au corpus original ayant $0{,}77$ sur Dev et $0{,}74$ sur Test, l'augmentation des données a conduit à une amélioration des performances.

En ce qui concerne les améliorations pour chaque catégorie, la figure 4 présente les scores F1 pour les trois expériences. Tout d'abord, pour la catégorie MCGS, des améliorations importantes sont

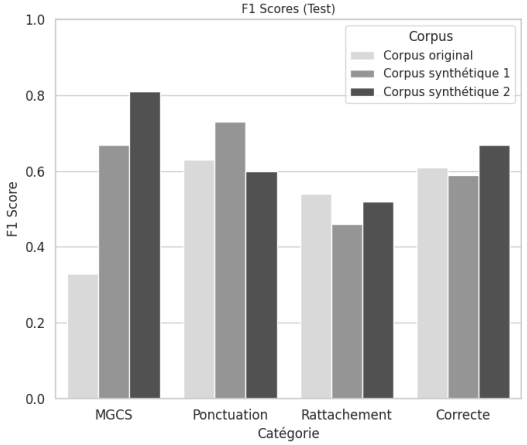

FIGURE 4 – Les scores F1 pour les quatre catégories

observées. Plus précisément, le score F1 de MCGS passe de $0{,}33$ pour le corpus original à $0{,}67$ pour le corpus synthétique 1 et à $0{,}81$ pour le corpus synthétique 2. Ensuite, pour la catégorie Correcte, le corpus synthétique 2 contient davantage de phrases de cette catégorie. Nous observons également que la performance de la catégorie Correcte s'améliore dans le corpus synthétique 2. Le score F1 est de $0{,}67$, contre $0{,}59$ dans le corpus synthétique 1 et $0{,}61$ dans le corpus original. Pour les catégories Ponctuation et Rattachement, les résultats n'indiquent pas encore d'amélioration marquée. Le corpus synthétique 1 obtient le meilleur score pour Ponctuation, tandis que le corpus original atteint le meilleur score pour Rattachement.

4.2 Classification avec des LLM

Afin de constituer un corpus plus large, cette section explore l'utilisation des LLM pour les tâches d'annotation. Une approche consiste à fournir au LLM un petit nombre d'exemples sous forme de paires **(exemple, étiquette)** dans le prompt. Cela lui permet de généraliser à de nouvelles tâches sans mise à jour de ses paramètres (Brown *et al.*, 2020; Min *et al.*, 2022; Akyürek *et al.*, 2022; Bai *et al.*, 2023). Cette méthode est également connue sous le nom d'apprentissage en contexte (*in-context learning*).

Les travaux antérieurs ont montré que l'apprentissage en contexte est sensible aux exemples fournis. Zhao *et al.* (2021) ont observé que l'ordre des exemples influence les résultats finaux. Yoo *et al.* (2022), en testant différents nombres d'exemples (de 1 à 16), ont constaté une corrélation positive entre le nombre d'exemples et les performances du modèle. De plus, Tang *et al.* (2024) ont souligné l'importance du choix des exemples et ont proposé des méthodes pour les sélectionner de manière optimale. Selon Yoo *et al.* (2022), afin de réduire la sensibilité du modèle aux exemples et d'améliorer la robustesse des performances, il est recommandé de fournir une description détaillée et explicite de la tâche. Basé sur ces remarques, la figure 5 montre un template de prompt structuré qui décrit précisément la tâche avec démonstration d'exemples.

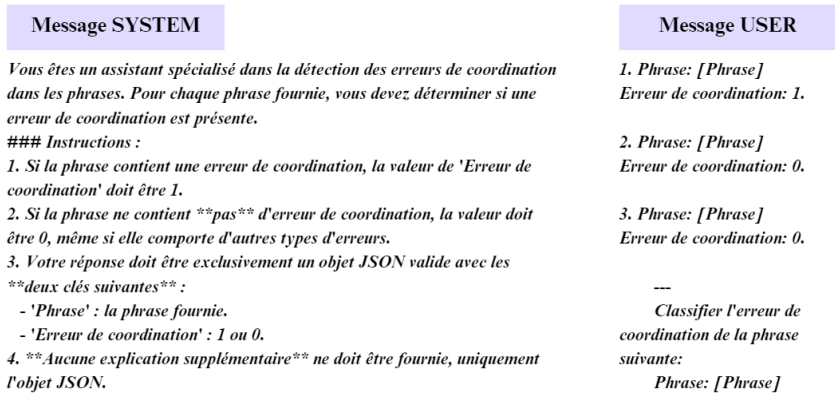

FIGURE 5 – Prompt pour la tâche de prédiction

4.2.1 Expériences

Nos expériences utilisent les versions Instruct (FP16) des modèles LLaMA (Touvron *et al.*, 2023) et Qwen (Yang *et al.*, 2024), couvrant une gamme de tailles de 3 milliards (3B) à 70 milliards (70B) de paramètres. Cela permet d'observer l'impact de la taille du modèle sur les performances de la tâche.

L'ensemble de test est constitué de 100 phrases sélectionnées aléatoirement, comprenant 50 phrases correctes et 50 phrases erronées. Les exemples utilisés pour la démonstration dans les prompts sont également choisis aléatoirement parmi le reste du corpus.

Pour chaque modèle, nous construisons cinq configurations avec 0, 3, 10, 20 et 50 exemples. Le cas zéro exemple correspond à une configuration où aucun exemple n'est fourni ; seule la description de la tâche est incluse dans le prompt.

Concernant les hyperparamètres, nous avons fixé la température entre 0 et $0{,}1$ et le seed à 42 afin d'anticiper la variation des réponses du modèle.

Le tableau 3 montre les résultats des prédictions par les modèles choisis. Le modèle LLaMA 3B atteint son meilleur score avec 20 exemples, obtenant une exactitude de $0{,}67$. De son côté, Qwen 3B atteint sa meilleure exactitude de $0{,}64$ avec 10 et 20 exemples. Toutefois, lorsque le nombre d'exemples se trouve à 50, les performances chutent fortement, $0{,}53$ pour LLaMA 3B et $0{,}58$ pour Qwen 3B.

Le modèle LLaMA 8B obtient son meilleur score avec 10 exemples, avec une exactitude de $0{,}69$. Cependant, ses performances diminuent progressivement avec davantage d'exemples et subissent une chute brutale avec 50 exemples. Pour Qwen 7B, le meilleur score est obtenu aussi avec 10 exemples, atteignant une exactitude de $0{,}71$, avant de décliner par la suite.

Le modèle LLaMA 70B a obtenu sa meilleure exactitude de $0{,}69$ avec 50 exemples, suivi de près par 10 exemples $(0{,}68)$. Toutefois, ce score ne montre pas d'amélioration significative par rapport à LLaMA 8B, bien que le modèle parvienne encore à maintenir des performances élevées avec 50 exemples. Quant à Qwen 32B, il atteint son meilleur score avec 10 exemples, avec une exactitude de $0{,}73$ et un score F1 de $0{,}77$, ce dernier étant le meilleur résultat obtenu parmi toutes les expériences. Lorsque 50 exemples sont utilisés, les performances chutent significativement.

Modèle	Nombre d'exemples	F1 score	Exactitude
LLaMA 3.2 3B	0	0,51	0,64
	3	0,66	0,65
	10	0,58	0,64
	20	**0,7**	**0,67**
	50	0,23	0,53
Qwen 2.5 3B	0	0,45	0,59
	3	0,44	0,56
	10	**0,58**	**0,64**
	20	**0,58**	**0,64**
	50	0,46	0,58
LLaMA 3.1 8B	0	0,5	0,64
	3	0,7	0,67
	10	**0,71**	**0,69**
	20	0,67	0,66
	50	0,46	0,62
Qwen 2.5 7B	0	0,60	0,65
	3	0,72	0,67
	10	**0,71**	**0,71**
	20	0,62	0,67
	50	0,62	0,67
LLaMA 3.3 70B	0	0,51	0,64
	3	0,49	0,63
	10	0,61	0,68
	20	0,56	0,64
	50	**0,65**	**0,69**
Qwen 2.5 32B	0	0,70	0,71
	3	0,75	0,7
	10	**0,77**	**0,73**
	20	0,75	0,72
	50	0,73	0,64

TABLE 3 – Résultats des prédictions par les LLM

4.3 Discussion

4.3.1 Corpus synthétiques dans l'apprentissage profond basé sur CamemBERT

L'utilisation des corpus synthétiques (CS1 et CS2) permet une amélioration par rapport au corpus original (CO). Notamment, le score LRAP atteint $0{,}84$ sur Dev et $0{,}81$ sur Test pour CS2, dépassant ainsi celui de CO ($0{,}77$ sur Dev et $0{,}74$ sur Test). Ces résultats indiquent que la stratégie d'augmentation permet de mieux équilibrer les catégories et, par conséquent, d'améliorer la capacité du modèle à classer les étiquettes pertinentes.

Pour la catégorie MCGS, son score F1, passe de $0{,}33$ pour le corpus original à $0{,}81$ pour CS2. Cela

montre que l'augmentation de cette catégorie, initialement sous-représentée, est déterminante pour l'amélioration des performances de prédiction.

Concernant la catégorie Correcte, CS2, ayant plus de phrases correctes par rapport à CS1, permet également une progression des performances. Cela signifie que plus de phrases correctes permettrait au modèle de mieux apprendre ce qui constitue une phrase correcte, réduisant ainsi la probabilité qu'une phrase réellement correcte soit classée à tort comme erronée (c'est-à-dire, diminuer les faux négatifs).

Enfin, les résultats pour les catégories Ponctuation et Rattachement restent instables. Ceci indique que l'augmentation des données n'est pas uniformément bénéfique pour toutes les classes.

4.3.2 Annotation avec des LLM

Dans la tâche d'annotation avec les LLM, nous nous rendons compte que l'apprentissage en contexte repose non seulement sur la compréhension de la tâche décrite, avec ou sans démonstration d'exemples, mais aussi sur la capacité du modèle à mobiliser les connaissances acquises lors du pré-entraînement pour prédire de nouvelles données. Ainsi, trois facteurs clés influencent les performances : la taille du modèle, la formulation de la tâche et le nombre d'exemples fournis.

Nous observons que, dans la plupart des cas, les performances s'améliorent progressivement à mesure que le nombre d'exemples augmente jusqu'à 10. Cela suggère que, bien que les modèles possèdent une certaine connaissance de la coordination, ils ont besoin d'exemples pour mieux s'adapter à la tâche. Cependant, au-delà de 10 exemples, les performances deviennent instables. En particulier, avec 50 exemples, seul LLaMA 70B parvient à obtenir son meilleur score, ce qui indique que les petits modèles sont plus sensibles aux limitations liées à la longueur du contexte.

Par ailleurs, nous constatons que la taille des modèles Qwen est corrélée de façon négative avec leur sensibilité au nombre d'exemples. Le modèle le plus grand, Qwen 32B, a obtenu les scores supérieurs à $0{,}7$, sauf avec 50 exemples. En revanche, cette tendance n'est pas aussi marquée pour la série LLaMA, dont les performances varient de manière moins systématique en fonction du nombre d'exemples.

5 Conclusion et perspectives

Cette étude explore deux approches visant à améliorer les performances d'un outil de détection des erreurs de coordination, en répondant aux défis du déséquilibre des classes et du volume limité de corpus annotés.

La première approche repose sur l'augmentation des données afin d'améliorer les performances du modèle fondé sur CamemBERT. Concrètement, nous avons utilisé un LLM pour générer artificiellement des phrases erronées en imposant la reproduction de types spécifiques d'erreurs. Ces phrases ont ensuite été intégrées au corpus original afin d'établir un corpus synthétique. Grâce à cette méthode, nous avons pu équilibrer la distribution des erreurs. Les résultats expérimentaux montrent une progression significative : le score LRAP en classification multi-label est passé de $0{,}74$ à $0{,}81$; le score F1 de la catégorie MCGS est passée de $0{,}33$ à $0{,}81$. Toutefois, comme l'a souligné dans Li *et al.* (2023), il n'existe actuellement pas de métrique standardisée permettant d'évaluer la qualité des

données générées. Un axe de recherche futur consistera à considérer des métriques pour des données synthétiques, afin d'analyser l'impact de différentes distributions de données sur les performances des modèles.

Une seconde approche est l'annotation automatique avec les LLM. Pour ce faire, nous avons testé des modèles de tailles variées, en expérimentant avec 0, 3, 10, 20 et 50 exemples fournis en démonstration. Nos résultats indiquent qu'autour de 10 exemples pourraient être un choix idéal, tandis qu'au-delà de 20 exemples, l'amélioration des performances n'est pas évidente. À l'avenir, il sera possible d'intégrer un raisonnement explicite dans les prompts. Les linguistes pourraient ainsi décomposer le processus d'analyse des erreurs en étapes, permettant aux modèles de mieux structurer leur raisonnement sur cette tâche. Une fois que nos templates de prompt seront mieux développés, nous envisagerons d'expérimenter avec des modèles plus puissants (tels que GPT-4o ou DeepSeek V3) et sur des ensembles de test plus étendus.

Références

ABEILLÉ A. & GODARD D. (2021). *La grande grammaire du français*. Éditions Actes Sud.

AKYÜREK E., SCHUURMANS D., ANDREAS J., MA T. & ZHOU D. (2022). What learning algorithm is in-context learning ? investigations with linear models. *arXiv preprint arXiv :2211.15661*.

BAI Y., CHEN F., WANG H., XIONG C. & MEI S. (2023). Transformers as statisticians : Provable in-context learning with in-context algorithm selection. *Advances in neural information processing systems*, **36**, 57125–57211.

BROWN T., MANN B., RYDER N., SUBBIAH M., KAPLAN J. D., DHARIWAL P., NEELAKANTAN A., SHYAM P., SASTRY G., ASKELL A. *et al.* (2020). Language models are few-shot learners. *Advances in neural information processing systems*, **33**, 1877–1901.

LE H., VIAL L., FREJ J., SEGONNE V., COAVOUX M., LECOUTEUX B., ALLAUZEN A., CRABBÉ B., BESACIER L. & SCHWAB D. (2019). Flaubert : Unsupervised language model pre-training for french. *arXiv preprint arXiv :1912.05372*.

LI Y., BONATTI R., ABDALI S., WAGLE J. & KOISHIDA K. (2024). Data generation using large language models for text classification : An empirical case study. *arXiv preprint arXiv :2407.12813*.

LI Z., ZHU H., LU Z. & YIN M. (2023). Synthetic data generation with large language models for text classification : Potential and limitations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, p. 10443–10461.

MARTIN L., MULLER B., SUÁREZ P. J. O., DUPONT Y., ROMARY L., DE LA CLERGERIE É. V., SEDDAH D. & SAGOT B. (2020). Camembert : a tasty french language model. In *ACL 2020-58th Annual Meeting of the Association for Computational Linguistics*.

MEONI S., RYFFEL T. & DE LA CLERGERIE É. V. (2023). Annotation d'entités cliniques en utilisant les larges modèles de langue. In *CORIA-TALN (1)*.

MIN S., LYU X., HOLTZMAN A., ARTETXE M., LEWIS M., HAJISHIRZI H. & ZETTLEMOYER L. (2022). Rethinking the role of demonstrations : What makes in-context learning work ? *arXiv preprint arXiv :2202.12837*.

NORESKAL L. (2022). *Erreurs dans les phrases coordonnées au sein des rédactions universitaires : typologie et détection*. Thèse de doctorat, Université Paris Nanterre.

NORESKAL L., ESHKOL I. & DESMETS M. (2023). Détecter une erreur dans les phrases coordonnées au sein des rédactions universitaires. In *Actes de CORIA-TALN 2023. Actes de la 30e*

Conférence sur le Traitement Automatique des Langues Naturelles (TALN), volume 1 : travaux de recherche originaux–articles longs, p. 248–261.

TANG C., QU F. & WU Y. (2024). Ungrammatical-syntax-based in-context example selection for grammatical error correction. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies (Volume 1 : Long Papers)*, p. 1758–1770.

TOUVRON H., LAVRIL T., IZACARD G., MARTINET X., LACHAUX M.-A., LACROIX T., ROZIÈRE B., GOYAL N., HAMBRO E., AZHAR F. *et al.* (2023). Llama : Open and efficient foundation language models. *arXiv preprint arXiv :2302.13971*.

WHITEHOUSE C., CHOUDHURY M. & AJI A. F. (2023). Llm-powered data augmentation for enhanced cross-lingual performance. *arXiv preprint arXiv :2305.14288*.

YANG A., YANG B., ZHANG B., HUI B., ZHENG B., YU B., LI C., LIU D., HUANG F., WEI H. *et al.* (2024). Qwen2. 5 technical report. *arXiv preprint arXiv :2412.15115*.

YOO K. M., KIM J., KIM H. J., CHO H., JO H., LEE S.-W., LEE S.-G. & KIM T. (2022). Ground-truth labels matter : A deeper look into input-label demonstrations. *arXiv preprint arXiv :2205.12685*.

ZHAO Z., WALLACE E., FENG S., KLEIN D. & SINGH S. (2021). Calibrate before use : Improving few-shot performance of language models. In *International conference on machine learning*, p. 12697–12706 : PMLR.

ZHOU Y., GUO C., WANG X., CHANG Y. & WU Y. (2024). A survey on data augmentation in large model era. *arXiv preprint arXiv :2401.15422*.

Annexes

A1 Détails supplémentaires sur l'augmentation des données

Nous présentons dans le tableau 4 les détails de deux sous-types d'erreur MCGS traités. Pour les cas présentant une régularité syntaxique plus simple, il est possible de décrire l'erreur sans recourir à un exemple original. Pour les cas plus complexes à décrire, nous incluons un exemple dans la description.

Description d'erreur	Exemples des phrases générées
Préposition ajoutée de manière incorrecte pour le GN après la conjonction : [GN] + [Conjonction de Coordination] + [PREP + GN].	— *Le street art questionne les normes urbaines et sur l'existence des murs comme espaces d'expression.* — *La peinture impressionniste a transformé l'art traditionnel et dans des perceptions classiques aussi.* — *Ils valorisent la maîtrise des langues et de l'élitisme dans le discours académique.*
La première coordonnée est une relative en "dont", la seconde est coordonnée à un verbe infinitif de la première relative et présente un verbe à l'infinitif qui n'attend pas de groupe prépositionnel en "de". Fournis des exemples similaires au suivant : "Cette caricature défend le fait que les personnes se plaignent de choses dont elles pourraient se passer ou trouver des solutions."	— *Les romans dystopiques inspirent des mouvements écologiques dont ils devraient s'inquiéter ou changer leur stratégie.* — *La photographie de mode montre des corps féminins dont il faudrait s'inspirer ou respecter la diversité.* — *L'humour absurde dans la littérature satirique provoque des réactions dont le public peut rire ou arrêter de lire.*

TABLE 4 – Exemples générés