Actes de CORIA-TALN 2026
Nantes, France, 29 juin au 3 juillet 2026

Un décodeur pour l'analyse sémantique AMR en français

Thomas Checchin[1] Julien Jacques[1] Adrien Guille[1]
(1) Université Lumière Lyon 2, Université Claude Bernard Lyon 1, ERIC, 69007, Lyon, France
thomas.checchin@univ-lyon2.fr, julien.jacques@univ-lyon2.fr,
adrien.guille@univ-lyon2.fr

RÉSUMÉ

AMR est un cadre de représentation de la sémantique des phrases sous la forme de graphes, initialement pensé pour la langue anglaise. De nombreuses techniques de traitement automatique de la langue s'appuient sur ces représentations, notamment pour compléter, corriger ou évaluer la production de grands modèles de langue. Or, il n'existe à ce jour pas d'analyseur sémantique AMR performant et librement disponible pour la langue française. C'est pourquoi, dans cet article, nous décrivons et partageons un analyseur basé sur un petit décodeur, entraîné selon une stratégie en deux temps, d'abord sur un vaste corpus AMR anglais, puis sur un corpus AMR français diversifié. L'évaluation montre les bonnes performances de cet analyseur et sa robustesse face à différents registres linguistiques. La quantification sur 8 bits, spécifiquement calibrée pour l'AMR, permet d'accroître l'efficacité de l'analyseur sans dégradation de la performance. Une version quantifiée sur 4 bits permet encore d'accroître son efficacité, au prix d'une légère baisse de la performance en analyse sémantique.

ABSTRACT

A Decoder for French AMR Parsing

AMR is a framework for representing sentence semantics in the form of graphs, initially designed for the English language. Numerous techniques rely on these representations, particularly to augment, correct, or evaluate the outputs of large language models. However, to date, there exists no high-performing and freely available AMR semantic parser for the French language. Thus, in this paper, we introduce and release a parser based on a lightweight decoder, trained with a 2-step strategy, relying on a large AMR corpus in English and a diversified French AMR corpus. Our evaluation demonstrates the parser's strong performance and its robustness across various linguistic registers. An 8-bit quantization, specifically calibrated for AMR, enhances the parser's computational efficiency without any degradation in performance. Furthermore, a 4-bit quantized version yields even greater efficiency, albeit at the cost of a minor performance drop in AMR parsing.

MOTS-CLÉS : Analyse sémantique, AMR, décodeur.

KEYWORDS: Semantic parsing, AMR, decoder.

1 Introduction

Le formalisme AMR (Banarescu *et al.*, 2013), acronyme anglais pour *Abstract Meaning Representation*, décrit une manière normalisée de représenter la sémantique d'une phrase, de sorte que différentes phrases ayant la même signification ont la même représentation. Une représentation AMR consiste en un graphe où les nœuds correspondent à des concepts et où les arcs expriment des relations

Actes des 33ème Conférence sur le Traitement Automatique des Langues Naturelles.
Volume 1 : articles scientifiques originaux, pages 610–620.
Cette œuvre est mise à disposition sous licence Attribution 4.0 International.

sémantiques entre eux. Ce graphe capture plusieurs phénomènes linguistiques (tels que les entités nommées, les coréférences ou encore les rôles sémantiques) ce qui en fait une représentation riche et utile pour la résolution de nombreuses tâches. Par exemple, les graphes AMR sont utilisés pour la détection des hallucinations des grands modèles de langue (Chen *et al.*, 2025), pour mesurer leur factualité (Qiu *et al.*, 2024) ou encore pour améliorer automatiquement les traductions vers l'anglais (Wein & Schneider, 2024). Leur pertinence a aussi été mise en avant dans le traitement de textes dans un contexte sensible, *e.g.* dans un contexte juridique, les prédictions basées sur les graphes AMR étant plus équitables que les prédictions de grands modèles de langue (Vijay & Hershcovich, 2024).

Depuis la mise au point du formalisme AMR et le partage du corpus AMR anglophone *LDC* (Knight *et al.*, 2020), de nombreux analyseurs sémantiques ont été développés. Alors que les premiers analyseurs se basent sur des automates finis (Flanigan *et al.*, 2014; Wang *et al.*, 2015, 2016), les analyseurs suivants (Procopio *et al.*, 2021; Bevilacqua *et al.*, 2021; Orlando *et al.*, 2022) reposent sur des modèles de langues (Charpentier *et al.*, 2024, 2025), de type encodeurs-décodeurs (dérivés de *BART* (Lewis *et al.*, 2020) ou *mT5* (Xue *et al.*, 2021)). Étant entraînés sur le corpus *LDC*, ceux-ci sont efficaces pour traiter la langue anglaise mais affichent de faibles performances sur les autres langues (Kang *et al.*, 2023). C'est pourquoi, plus récemment, plusieurs études ont été menées pour tenter de mettre au point un analyseur efficace en français (Kang *et al.*, 2023, 2024, 2025a). Ces analyseurs se basent tous sur le modèle encodeur-décodeur *mBART* (Liu *et al.*, 2020) et requièrent une traduction automatique, en français ou en plusieurs langues, du corpus *LDC* (en l'occurrence réalisée avec un outil fermé, à savoir *DeepL*). Par ailleurs, les poids de ces modèles ne sont pas disponibles et ne peuvent donc pas être réutilisés. Sans analyseur AMR pour le français prêt à l'emploi, la transposition des méthodes développées sur la base des graphes AMR à la langue française n'est donc pas évidente.

Dans cet article, nous présentons et partageons un nouvel analyseur sémantique AMR pour le français, en nous appuyant sur un décodeur pré-entraîné récent. Notre approche repose sur un modèle ouvert, à savoir le plus petit modèle de la famille *TranslateGemma* (Finkelstein *et al.*, 2026), qui est une variante spécifiquement ajustée de *Gemma 3* (Gemma Team, 2025) pour la traduction. Aussi, plutôt que de traduire le corpus *LDC*, nous optons pour une démarche d'apprentissage par transfert. Nous ajustons d'abord *TranslateGemma* sur le corpus *LDC* en anglais, puis poursuivons l'apprentissage sur une combinaison de trois corpus français annotés en AMR, lesquels sont publiquement disponibles, à savoir *ding-01*, un corpus de dialogue spontané (Kang *et al.*, 2025b), *Le Petit Prince* (Banarescu *et al.*, 2015) et la sous-partie française du corpus *MASSIVE*, un corpus de requêtes à un assistant (Regan *et al.*, 2024).

En résumé :

— Nous introduisons un nouvel analyseur AMR pour le français basé sur une architecture décodeur, *TranslateGemma*, marquant une transition depuis les modèles encodeur-décodeur plus anciens vers l'utilisation d'un modèle de langue bénéficiant des avancées récentes en matière de pré-entraînement et d'implémentation.
— Nous montrons la pertinence de ce changement d'architecture et l'efficacité de notre stratégie d'entraînement, laquelle est en deux phases, d'abord basée sur un vaste corpus anglais puis sur un corpus français plus restreint mais diversifié. En effet, le modèle obtenu affiche une performance significativement supérieure aux scores rapportés dans la littérature, tout en étant robuste face à différents registres linguistiques.
— Enfin, nous partageons librement cet analyseur et des variantes, dont une variante quantifiée sur 8 bits sans dégradation de performance en analyse AMR : `https://huggingface.co/AdrienGuille/GemmAMR-fr-v1`

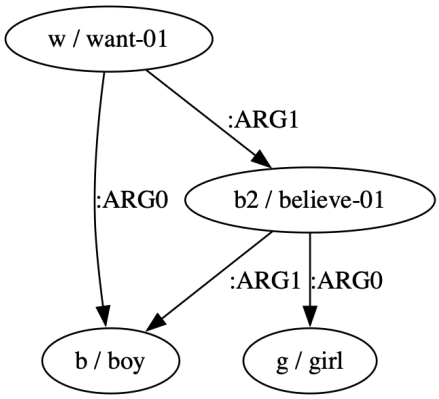

FIGURE 1 – Représentation visuelle du graphe AMR pour la phrase «Le garçon veut que la fille le croît.»

2 Travaux connexes

Un graphe AMR représente formellement le sens d'une phrase sous la forme d'un graphe orienté, enraciné et acyclique (Banarescu *et al.*, 2013). Les sommets représentent les concepts sémantiques intervenant dans la phrase, qui peuvent être des mots anglais, des «frames» PropBank (Kingsbury & Palmer, 2002) ou bien des concepts abstraits (*e.g.* «date-entity», «country»). Les arcs quant à eux représentent les interactions entre les concepts. Ils sont étiquetés, soit par des arguments numérotés (*e.g.* « :ARG0» et « :ARG1», qui, le plus souvent, servent à indiquer qui réalise une action et qui la subit) soit par des rôles généraux (*e.g.* « :location», « :time»). La figure 1 montre le graphe AMR représentant la phrase «Le garçon veut que la fille le croît.». Il est à noter que ce graphe AMR est le même que pour la phrase anglaise équivalente «The boy wants the girl to believe him.». Les graphes AMR s'expriment aussi textuellement, selon la notation Penman (Kasper, 1989). Par exemple, la représentation AMR de la phrase mentionnée précédemment selon cette notation est la suivante :

```
(w / want-01
    :ARG0 (b / boy)
    :ARG1 (b2 / believe-01
        :ARG0 (g / girl
        :ARG1 b))
```

Grâce au partage du corpus *LDC*, un grand corpus anglais annoté en AMR, aujourd'hui à sa troisième version (Knight *et al.*, 2020), plusieurs analyseurs AMR automatiques ont été mis au point pour la langue anglaise. Alors que les premiers travaux s'appuient sur des automates finis (Wang *et al.*, 2015, 2016), les travaux plus récents s'appuient sur des modèles de langue encodeur-décodeur (*BART* (Lewis *et al.*, 2020) ou *mT5* (Xue *et al.*, 2021)). Ceux-ci sont entraînés à générer à partir d'une phrase son codage AMR en notation Penman (Bevilacqua *et al.*, 2021; Procopio *et al.*, 2021; Orlando *et al.*, 2022). Kang *et al.* (2023) revendiquent les premiers travaux visant à mettre au point un analyseur AMR pour le français. Ils entraînent conjointement un encodeur-décodeur *mBART* (Liu *et al.*, 2020) à traduire des phrases du français vers l'anglais et à générer le codage AMR de phrases en anglais (sur la base du corpus *LDC*) afin d'obtenir un système capable de réaliser l'analyse AMR en français, en mode «0-shot». De premiers résultats sont ainsi mesurés en analyse AMR en français, sur le corpus *LDC* traduit en français (à l'aide de *DeepL*). Il a également été proposé d'entraîner conjointement le modèle *mBART* sur le corpus *LDC* traduit automatiquement (à l'aide de *DeepL*) dans 13 langues, dont

le français, pour obtenir un analyseur multilingue (Kang *et al.*, 2024) [1]. Récemment, *DinG*, un corpus de transcriptions de dialogues spontanés en français (Boritchev & Amblard, 2022) a été annoté en AMR (Kang *et al.*, 2025b), nommé *ding-01*, ouvrant la voie à l'entraînement d'un analyseur sur des données nativement en français. Kang *et al.* (2025a) proposent d'entraîner un modèle *mBART* en deux temps, d'abord sur le corpus *LDC* traduit en français (à l'aide de *DeepL*), avant de le spécialiser sur le corpus *ding-01*. Les résultats obtenus sont encourageants et montrent l'intérêt de l'entraînement en deux temps, puisqu'entraîner *mBART* uniquement sur le corpus *ding-01* donne des résultats nettement inférieurs. Il est à noter que d'autres corpus français annotés en AMR existent, à savoir *Le Petit Prince*, l'annotation ayant été faite à partir de sa traduction originale en anglais (Banarescu *et al.*, 2015; Kang *et al.*, 2023) et une sous-partie du jeu de données *MASSIVE*, composé de transcriptions de requêtes à un assistant vocal (Regan *et al.*, 2024).

3 Modèle proposé

Les études précédentes en matière d'analyse AMR en français se basent toutes sur *mBART*, un modèle relativement ancien, dont l'entraînement repose sur des variantes du corpus *LDC* obtenues par traduction automatique et, au mieux, une partie des données francophones disponibles (à savoir, uniquement le jeu de données *ding-01*). De plus, les modèles obtenus ne sont pas disponibles. Ceci nous motive à mettre au point un analyseur sémantique AMR pour la langue française, basé sur un modèle de langue contemporain, en exploitant tous les corpus francophones disponibles, à notre connaissance, et sans avoir recours à la traduction automatique. Notre modèle est disponible sur *HuggingFace* : `https://huggingface.co/AdrienGuille/GemmAMR-fr-v1`.

3.1 Modèle pré-entraîné

Nous nous basons sur le plus petit modèle de la famille *TranslateGemma* [2] [3]. Il s'agit d'un décodeur obtenu par ajustement, à la fois supervisé et par renforcement (Finkelstein *et al.*, 2026), du modèle *Gemma 3* à un peu moins de 4 milliards de paramètres (Gemma Team, 2025). En plus d'accroître les performances en traduction multilingue, cet ajustement a aussi pour conséquence d'effacer le comportement conversationnel de *Gemma 3*. Ceci en fait un modèle particulièrement adapté à la résolution d'une tâche de type séquence à séquence, telle que la génération du codage AMR d'une phrase en français. Il est à noter que, s'agissant d'un modèle multimodal, le nombre de paramètres inclut ceux de l'encodeur visuel ; celui-ci nous étant inutile pour l'analyse sémantique, nous le supprimons, ce qui réduit le nombre de paramètres à 3,2 milliards.

3.2 Données utilisées pour l'ajustement

Corpus anglais. Nous utilisons le jeu de données *LDC* 3.0 (Knight *et al.*, 2020), qui comporte 59 255 phrases annotées en AMR, extraites notamment d'articles publiés dans divers médias, d'articles Wikipédia mais aussi de la traduction anglaise des Fables d'Ésope. Il est pré-découpé en échantillons

1. Un lien Google Drive pour télécharger le modèle est donné à l'adresse suivante, mais celui est mort au moment de la rédaction de cet article : `https://github.com/Emvista/Meta-XAMR-2024`
2. Poids du modèle : `https://huggingface.co/google/translategemma-4b-it`
3. Conditions d'utilisation du modèle : `https://ai.google.dev/gemma/terms`

d'entraînement, de développement et de test (93.9%, 2.9% et 3.2%), découpage que nous respectons pour nos expériences.

Corpus français. Nous combinons trois jeux de données pour constituer le corpus français :

— *ding-01* : qui comporte 2 915 transcriptions de répliques annotées, illustrant les dynamiques et disfluences propres au dialogue francophone spontané (Kang *et al.*, 2025b) ;

— *Le Petit Prince* : qui comporte 1 548 phrases annotées, illustrant le registre narratif et littéraire (Banarescu *et al.*, 2015; Kang *et al.*, 2023) ;

— *MASSIVE* (sous-partie française) : qui comporte 1 471 requêtes adressées à un assistant (Regan *et al.*, 2024).

Nous constituons un échantillon d'entraînement, un échantillon de développement et un échantillon de test pour nos expériences. Le jeu de données *ding-01* comporte une partie dédiée au test, celle-ci ayant fait l'objet d'une résolution de conflits entre les annotateurs AMR. Nous la préservons donc pour l'échantillon de test, que nous complétons avec la partie dédiée au test du jeu de données *MASSIVE* ainsi que les 548 dernières phrases du *Petit Prince*, celui-ci n'étant pas pré-découpé. Nous constituons les échantillons d'entraînement et de développement selon un ratio 90/10 en piochant parmi le reste des données *ding-01*, *MASSIVE* et les 1000 premières phrases du *Petit Prince*.

3.3 Protocole d'ajustement du modèle

Formatage des entrées et des sorties. Nous modifions le formatage recommandé par Finkelstein *et al.* (2026) pour passer le texte à traduire au modèle, afin de l'adapter à l'analyse sémantique AMR. Nous pré-traitons également la sortie, c'est-à-dire la notation Penman du graphe AMR, de la même manière que Kang *et al.* (2025a). Nous le linéarisons, nous renumérotons les variables pour utiliser une numérotation séquentielle et nous ajoutons des espaces autour des parenthèses. L'encadré ci-après illustre le formatage d'une paire entrée-sortie pour un exemple tiré du jeu de données *ding-01*.

Entrée reçue et sortie attendue

<bos><start_of_turn>user
You are a professional French (fr) to Abstract Meaning Representation (AMR) translator. Your goal is to accurately convey the meaning of the original French text while adhering to AMR annotation guidelines.
Produce only the AMR in Penman format, without any additional explanations or commentary. Please translate the following French text into AMR :

hum donc comment est-ce qu'on peut obtenir des points de victoire ?<end_of_turn>
<start_of_turn>model

(p / possible-01 :ARG1 (o / obtain-01 :ARG1 (p1 / point :mod (v / victory)) :manner (a / amr-unknown) :mod (h / hmm :mode "expressive") :discourse-marker "donc")) <end_of_turn>

Ajustement supervisé. Nous réalisons l'ajustement en deux phases, selon la méthode *LoRA* (Hu *et al.*, 2022), en ciblant toutes les transformations linéaires et avec des lots de taille 8. La particularité de la méthode *LoRA*, par rapport à un ajustement classique, est qu'elle ne modifie pas directement les paramètres du modèle, mais apprend les corrections à apporter aux matrices ciblées, corrections qu'elle approche par des produits de matrices de rang faible, ici de rang 32. D'abord, les corrections

sont calculées uniquement d'après les données du corpus *LDC*, de sorte à permettre au modèle d'apprendre les concepts et la grammaire AMR. Lorsque cette première phase d'ajustement est terminée (ce que nous déterminons par validation sur l'échantillon de développement du corpus *LDC*, réalisée tous les 500 pas), les corrections sont incorporées (par simple sommation) dans le modèle. La deuxième phase d'ajustement consiste alors à initialiser de nouvelles corrections, toujours en rang 32, et à les apprendre cette fois-ci d'après le corpus français. L'arrêt de cette phase est contrôlé par validation sur l'échantillon de développement du corpus français, réalisée tous les 50 pas.

3.4 Quantification du modèle

Dans l'optique de réduire l'empreinte mémoire du modèle, nous appliquons la méthode de quantification post-entraînement *GPTQ* (Frantar *et al.*, 2023). Cet algorithme de compression a la particularité de quantifier, usuellement sur 8 ou 4 bits, consécutivement et indépendamment, les couches du modèle à l'aide d'un échantillon de calibration. Nous utilisons, ici, un échantillon de 512 couples entrées-sorties piochés aléatoirement dans le jeu de validation du corpus français. *GPTQ* résout, pour chaque couche, un problème de reconstruction des paramètres en approximant l'information du second ordre. Pour ce faire, pour chaque couche, l'optimisation de cet objectif est guidée par l'approximation de la matrice hessienne inverse selon la décomposition de Cholesky.

4 Évaluation

Nous évaluons la qualité des AMR générés sur l'échantillon de test du corpus français, selon le score *Smatch*, plus précisément la F-mesure, en utilisant la bibliothèque *smatchpp* (Opitz, 2023).

4.1 Expérience préliminaire

Tout d'abord, nous reproduisons l'expérience menée par Kang *et al.* (2025a), en entraînant un premier modèle *mBART* sur le corpus *ding-01*, puis en entraînant un second modèle *mBART*, d'abord sur le corpus *LDC*, que nous ne traduisons pas contrairement à l'expérience originale, puis que nous ajustons sur le corpus *ding-01*. Kang *et al.* mesurent, respectivement, sur le jeu de test du corpus *ding-01*, des scores de 68,1 et de 73,5. Nous mesurons des scores proches, à savoir 69,5 et 73,8. Ces premiers résultats suggèrent qu'un modèle multilingue tel que *mBART* ne nécessite pas forcément la traduction du corpus *LDC* si le modèle est ensuite ajusté sur un corpus français. Ceci justifie empiriquement notre choix de ne pas traduire le corpus *LDC*, choix initialement motivé par la volonté de ne pas dépendre d'un outil additionnel. Entraîné dans les mêmes conditions, *TranslateGemma* obtient des scores de 73,5 et 77,8. Il est à noter que le corpus *ding-01* a été publié en ligne pour la première fois en mai 2025 [4], alors que l'entraînement de *Gemma 3* s'est achevé en août 2024 [5] et que *TranslateGemma* a été ajusté sur des corpus parallèles antérieurs à 2024. Par conséquent, la supériorité des résultats obtenus par rapport à *mBART* ne peut être expliquée par une contamination des données et est certainement due à la différence d'échelle des modèles.

4. Enregistrement sur Zenodo : https://zenodo.org/records/15537426
5. Fiche technique du modèle : https://ai.google.dev/gemma/docs/core/model_card_3

	LDC → ding-01	*LDC → ding-01 + LPP*	*LDC → ding-01 + LPP + MASSIVE*
ding-01	77,8	77,2	77,4
LPP	45,8	63,4	63,3
MASSIVE	46,1	56,6	80,0
score moyen	56,6	65,7	**73,6**

TABLE 1 – Scores *Smatch* obtenus avec notre modèle selon la composition du corpus français

	LDC → ding-01	*LDC → ding-01 + LPP*	*LDC → ding-01 + LPP + MASSIVE*
ding-01	73,8	71,6	71,2
LPP	47,2	59,2	58,4
MASSIVE	42,4	44,7	74,6
score moyen	54,5	58,5	68,1

TABLE 2 – Scores *Smatch* obtenus avec le modèle *mBART* selon la composition du corpus français

4.2 Évaluation du modèle proposé

La table 1 rapporte les scores obtenus sur les échantillons de test par notre modèle, selon qu'il soit entraîné lors de la deuxième phase sur le corpus français complet ou bien sur une sous-partie. Il apparaît que l'ajout d'un jeu de données augmente la performance en test sur ce jeu de données, ainsi que la performance en test sur le ou les jeux de données non vus à l'entraînement. Par exemple, ajouter le jeu de données *Le Petit Prince* (abrégé *LPP* dans la table) augmente de 10 points le score sur le jeu de données *MASSIVE*, non vu à l'entraînement. Dans le même temps, l'ajout d'un jeu de données ne dégrade pas la performance en test pour les jeux de données déjà vus. Autrement dit, il semble que le modèle soit capable de généraliser de mieux en mieux avec l'ajout de plusieurs jeux de données. Le modèle le plus performant est obtenu avec le corpus français complet, affichant un score moyen de 73,6, montrant la pertinence de notre stratégie d'entraînement en deux temps, d'abord en anglais sur *LDC* puis en français sur le corpus combinant les 3 jeux de données français. C'est celui que nous partageons.

4.3 Comparaison avec un modèle *mBART*

La table 2 rapporte les scores obtenus avec le modèle *mBART*. Les scores sont dans l'ensemble nettement inférieurs à ceux obtenus avec le modèle que nous proposons, et il apparaît également que sa capacité de généralisation est moindre. On constate en effet que le score sur le corpus *ding-01* diminue avec l'incorporation d'autres jeux de données dans l'entraînement. Aussi, là où l'ajout du corpus *Le Petit Prince* augmente le score de 10 points sur le corpus *MASSIVE* pour notre modèle, ici le gain n'est que de 2,3 points. Le meilleur modèle *mBART* affiche un score moyen inférieur de plus de 5 points à notre modèle.

4.4 Impact de la quantification

Même si à l'ère des grands modèles de langue, un modèle à 3,2 milliards de paramètres se range dans la catégorie des petits modèles, son déploiement peut être compliqué dans un environnement contraint matériellement. Afin de faciliter sa réutilisation, nous le quantifions selon la méthode *GPTQ* afin de représenter les poids sur 8 bits au lieu de 16, la calibration de la quantification étant réalisée sur l'échantillon de développement du corpus français. Cette version quantifiée affiche une performance comparable au modèle initial, réalisant un score de 77,2 sur *ding-01*, de 63,6 sur *Le Petit Prince* et 79,9 sur *MASSIVE*, pour un score moyen identique au modèle initial. Ainsi, il est possible de réduire la mémoire nécessaire à son chargement et d'accélèrer les calculs sur du matériel gérant efficacement les opérations sur 8 bits, sans compromis en termes d'analyse sémantique. Nous quantifions également le modèle sur 4 bits afin de permettre une réutilisation dans un environnement très contraint. Cette version obtient des scores dégradés, à savoir 75,4 sur *ding-01*, 62,0 sur *Le Petit Prince* et 77,5 sur *MASSIVE*, ce qui reste cependant nettement supérieur aux scores obtenus avec le meilleur modèle *mBART*, alors que cette version quantifiée a une empreinte mémoire comparable à celle du modèle *mBART*.

5 Conclusion et perspectives

Nous avons présenté un analyseur sémantique AMR pour le français, basé sur un décodeur récent à 3,2 milliards de paramètres et entraîné sur des textes annotés en AMR, en anglais et en français. Ses performances sont nettement supérieures à celles obtenues avec un encodeur-décodeur, l'architecture habituellement employée pour cette tâche. La quantification calibrée, sur 8 bits, permet d'améliorer l'efficacité du modèle, sans dégrader sa performance en analyse sémantique. La quantification calibrée sur 4 bits permet d'atteindre une empreinte mémoire comparable à celle d'un encodeur-décodeur classique au prix d'une perte de performance, bien que celle-ci se maintienne encore nettement au-dessus de la performance de l'encodeur-décodeur. Nous pensons que cet analyseur serait utile à la communauté TALN francophone, en permettant d'explorer l'application de techniques récentes basées sur AMR, jusque-là cantonnées à l'anglais, *e.g.* pour la détection des hallucinations dans les énoncés produits par les grands modèles de langues (Chen *et al.*, 2025), pour la mesure de la factualité de ceux-ci (Qiu *et al.*, 2024), pour l'amélioration de la traduction vers le français (Wein & Schneider, 2024) ou encore pour développer des prédicteurs plus équitables (Vijay & Hershcovich, 2024). Dans l'hypothèse où de nouveaux jeux de données AMR en français ne seraient pas partagés à l'avenir, une perspective intéressante pour une future version de l'analyseur serait de partir d'une version plus grande de *TranslateGemma*, qui existe en version 12 et 27 milliards de paramètres, laquelle pourrait atteindre une performance supérieure car permettant un apprentissage par transfert plus puissant, puis de distiller le modèle résultant dans le plus petit modèle *TranslateGemma*, en espérant retenir une part importante de cette performance accrue.

Remerciements

Ce travail a bénéficié d'une aide de l'État gérée par l'Agence Nationale de la Recherche au titre du projet IMAGE-TEXTE-AVC portant la référence ANR-25-CE45-0054.

Références

BANARESCU L., BONIAL C., CAI S., GEORGESCU M., GRIFFITT K., HERMJAKOB U., KNIGHT K., KOEHN P., PALMER M. & SCHNEIDER N. (2013). Abstract Meaning Representation for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*.

BANARESCU L., BONIAL C., CAI S., GEORGESCU M., GRIFFITT K., HERMJAKOB U., KNIGHT K., KOEHN P., PALMER M. & SCHNEIDER N. (2015). Abstract Meaning Representation (AMR) 1.2.2 specification.

BEVILACQUA M., BLLOSHMI R. & NAVIGLI R. (2021). One SPRING to rule them both : Symmetric AMR semantic parsing and generation without a complex pipeline. In *Proceedings of AAAI*.

BORITCHEV M. & AMBLARD M. (2022). A multi-party dialogue ressource in French. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, Marseille, France : European Language Resources Association.

CHARPENTIER F., CUGLIARI J. & GUILLE A. (2024). Exploring semantics in pretrained language model attention. In *13th Joint Conference on Lexical and Computational Semantics (StarSEM)*.

CHARPENTIER F., CUGLIARI J. & GUILLE A. (2025). Probing attention in pre-trained llms to detect semantics. In *Proceedings of the 9th International Conference on Natural Language Processing and Information Retrieval (NLPIR)*.

CHEN K., CHEN Q., ZHOU J., TAO X., DING B., XIE J., XIE M., LI P. & FENG Z. (2025). Enhancing uncertainty modeling with semantic graph for hallucination detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

FINKELSTEIN M., CASWELL I., DOMHAN T., PETER J.-T., JURASKA J., RILEY P., DEUTSCH D., KOVACS G., DILANNI C., CHERRY C., BRIAKOU E., NIELSEN E., LUO J., BLACK K., MULLINS R., AGRAWAL S., XU W., KATS E., JASKIEWICZ S., FREITAG M. & VILAR D. (2026). Translategemma technical report.

FLANIGAN J., THOMSON S., CARBONELL J., DYER C. & SMITH N. A. (2014). A discriminative graph-based parser for the Abstract Meaning Representation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*.

FRANTAR E., ASHKBOOS S., HOEFLER T. & ALISTARH D. (2023). Gptq : Accurate post-training quantization for generative pre-trained transformers.

GEMMA TEAM (2025). Gemma 3 technical report.

HU E. J., SHEN Y., WALLIS P., ALLEN-ZHU Z., LI Y., WANG S., WANG L. & CHEN W. (2022). Lora : Low-rank adaptation of large language models. In *ICLR 2022*.

KANG J., BORITCHEV M. & COAVOUX M. (2025a). ding-01 :ARG0 : An AMR corpus for spontaneous French dialogue. In *Proceedings of the 16th International Conference on Computational Semantics*.

KANG J., BORITCHEV M. & COAVOUX M. (2025b). ding-01 :ARG0 un corpus AMR pour le français parlé spontané. In *Actes des 32ème Conférence sur le Traitement Automatique des Langues Naturelles (TALN), volume 1 : articles scientifiques originaux*.

KANG J., COAVOUX M., LOPEZ C. & SCHWAB D. (2024). Méta-apprentissage pour l'analyse AMR translingue. In *Actes de la 31ème Conférence sur le Traitement Automatique des Langues Naturelles, volume 1 : articles longs et prises de position*.

KANG J., COAVOUX M., SCHWAB D. & LOPEZ C. (2023). Analyse sémantique AMR pour le français par transfert translingue. In *Actes de CORIA-TALN 2023. Actes de la 30e Conférence sur le Traitement Automatique des Langues Naturelles (TALN), volume 2 : travaux de recherche originaux – articles courts*, Paris, France : ATALA.

KASPER R. (1989). A flexible interface for linking applications to penman's sentence generator. In *Speech and Natural Language : Proceedings of a Workshop Held at Philadelphia, Pennsylvania*.

KINGSBURY P. & PALMER M. (2002). From TreeBank to PropBank. In *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC)*.

KNIGHT K., BADARAU B., BARANESCU L., BONIAL C., BARDOCZ M., GRIFFITT K., HERMJAKOB U., MARCU D., PALMER M., O'GORMAN T. & SCHNEIDER N. (2020). Abstract Meaning Representation (AMR) annotation release 3.0.

LEWIS M., LIU Y., GOYAL N., GHAZVININEJAD M., MOHAMED A., LEVY O., STOYANOV V. & ZETTLEMOYER L. (2020). BART : Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

LIU Y., GU J., GOYAL N., LI X., EDUNOV S., GHAZVININEJAD M., LEWIS M. & ZETTLEMOYER L. (2020). Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, **8**.

OPITZ J. (2023). SMATCH++ : Standardized and extended evaluation of semantic graphs. In *Findings of the Association for Computational Linguistics : EACL 2023*, p. 1595–1607, Dubrovnik, Croatia : Association for Computational Linguistics.

ORLANDO R., CONIA S., FARALLI S. & NAVIGLI R. (2022). Universal semantic annotator : the first unified API for WSD, SRL and semantic parsing. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*.

PROCOPIO L., TRIPODI R. & NAVIGLI R. (2021). SGL : Speaking the graph languages of semantic parsing via multilingual translation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*.

QIU H., HUANG K.-H., QU J. & PENG N. (2024). AMRFact : Enhancing summarization factuality evaluation with AMR-driven negative samples generation. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies (Volume 1 : Long Papers)*.

REGAN M., WEIN S., BAKER G. & MONTI E. (2024). MASSIVE multilingual Abstract Meaning Representation : A dataset and baselines for hallucination detection. In *Proceedings of the 13th Joint Conference on Lexical and Computational Semantics (*SEM 2024)*, Mexico City, Mexico.

VIJAY S. & HERSHCOVICH D. (2024). Can Abstract Meaning Representation facilitate fair legal judgement predictions ? In *Proceedings of the Fifth Workshop on Insights from Negative Results in NLP*.

WANG C., PRADHAN S., PAN X., JI H. & XUE N. (2016). Camr at semeval-2016 task 8 : An extended transition-based amr parser. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*.

WANG C., XUE N. & PRADHAN S. (2015). A transition-based algorithm for amr parsing. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*.

WEIN S. & SCHNEIDER N. (2024). Lost in translationese ? reducing translation effect using Abstract Meaning Representation. In *Proceedings of the 18th Conference of the European Chapter*

of the Association for Computational Linguistics (Volume 1 : Long Papers), St. Julian's, Malta : Association for Computational Linguistics.

XUE L., CONSTANT N., ROBERTS A., KALE M., AL-RFOU R., SIDDHANT A., BARUA A. & RAFFEL C. (2021). mT5 : A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*.