



HAL
open science

Méta-apprentissage pour l'analyse AMR translingue

Jeongwoo Kang, Maximin Coavoux, Cédric Lopez, Didier Schwab

► To cite this version:

Jeongwoo Kang, Maximin Coavoux, Cédric Lopez, Didier Schwab. Méta-apprentissage pour l'analyse AMR translingue. 35èmes Journées d'Études sur la Parole (JEP 2024) 31ème Conférence sur le Traitement Automatique des Langues Naturelles (TALN 2024) 26ème Rencontre des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues (RECITAL 2024), Jul 2024, Toulouse, France. pp.144-156. <hal-04623014>

HAL Id: hal-04623014

<https://inria.hal.science/hal-04623014v1>

Submitted on 1 Jul 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons CC BY 4.0 - Attribution - International License

Méta-apprentissage pour l’analyse AMR translingue

Jeongwoo Kang^{1,2} Maximin Coavoux¹ Cédric Lopez² Didier Schwab¹

(1) Univ. Grenoble Alpes, CNRS, Grenoble INP, LIG, 38000 Grenoble, France

(2) Emvista, Immeuble Le 610, 10 Rue Louis Breguet Bâtiment D, 34830 Jacou, France

¹{prénom}. {nom}@univ-grenoble-alpes.fr

²{prénom}. {nom}@emvista.com

RÉSUMÉ

L’analyse AMR translingue consiste à prédire des analyses sémantiques AMR dans une langue cible lorsque les données d’entraînement ne sont disponibles que dans une langue source. Cette tâche n’a été étudiée que pour un petit nombre de langues en raison du manque de données multilingues. En s’inspirant de [Langedijk et al. \(2022\)](#), qui appliquent le méta-apprentissage à l’analyse syntaxique en dépendances translingue, nous étudions le méta-apprentissage pour l’analyse AMR translingue. Nous évaluons nos modèles dans des scénarios *zero-shot* et *few-shot* en croate, en farsi, en coréen, en chinois et en français. En particulier, nous développons dans le cadre de cet article des données d’évaluation en coréen et en croate, à partir du corpus AMR anglais *Le Petit Prince*. Nous étudions empiriquement cette approche en la comparant à une méthode classique d’apprentissage conjoint.

ABSTRACT

Meta learning for cross-lingual AMR parsing

Cross-lingual AMR parsing is the task of predicting AMR graphs in a target language when training data is available only in a source language. Due to limited multilingual data for the task, cross-lingual AMR parsing has only been explored in a small set of languages. Taking inspiration from [Langedijk et al. \(2022\)](#) who used meta-learning for cross-lingual dependency parsing, we investigate the use of meta-learning for cross-lingual AMR parsing. We evaluate our models in zero-shot and few-shot scenarios and assess their effectiveness in Croatian, Farsi, Korean, Chinese, and French. Notably, we develop Korean and Croatian test sets for this work, based on the existing *The Little Prince* AMR corpus, and make it publicly available. We empirically study this approach by comparing it to a classical joint learning method.

MOTS-CLÉS : L’analyse AMR translingue, Méta-apprentissage, Apprentissage zéro/few-shot.

KEYWORDS: Crosslingual AMR parsing, Meta-learning, Zero/Few-shot learning.

1 Introduction

Abstract Meaning Representation ([Banarescu et al., 2013](#), AMR) est un formalisme qui représente le sens des textes sous la forme de graphes acycliques orientés (en anglais *directed acyclic graphs*). Les graphes AMR capturent la sémantique des textes tout en faisant abstraction de leurs réalisations syntaxiques. Le formalisme a été conçu à l’origine pour les textes en anglais uniquement. Cependant, [Damonte & Cohen \(2018\)](#) ont montré qu’AMR pouvait être utilisé pour d’autres langues telles que

l’espagnol, l’italien, le chinois et l’allemand. AMR est un formalisme non ancré, c’est-à-dire que les tokens de la phrase ne sont pas des nœuds du graphe. Il est donc possible de construire des données multilingues *silver* en utilisant la traduction automatique, une phrase et sa traduction ayant en théorie la même représentation AMR. Depuis lors, de nombreuses approches ont adopté AMR dans le contexte multilingue (Procopio *et al.*, 2021; Bliloshi *et al.*, 2020; Xu *et al.*, 2021; Cai *et al.*, 2021; Sheth *et al.*, 2021). Néanmoins, l’un des principaux verrous de cette tâche est le manque de données. Actuellement, les données d’entraînement ne sont disponibles qu’en anglais (Knight *et al.*, 2017, 2020) et les données d’évaluation pour 6 langues : l’anglais, l’allemand, l’espagnol, l’italien, le chinois (Damonte & Cohen, 2018), langues sur lesquels se concentrent la plupart des travaux en AMR translingue ; et plus récemment pour le français (Kang *et al.*, 2023).

Dans cette étude, notre objectif est d’appliquer l’analyse AMR à des langues plus diverses qui ont été peu ou pas étudiées dans les précédentes approches et de remédier au manque de données d’entraînement à l’aide de l’apprentissage *few-shot*. En s’inspirant de Langedijk *et al.* (2022), qui ont appliqué le méta-apprentissage pour l’analyse syntaxique translingue en *few-shot*, nous appliquons le méta-apprentissage pour l’analyse AMR translingue. Pour examiner la pertinence de cette méthode, nous la comparons à une méthode classique d’apprentissage conjoint (*joint learning*). Pour cela, nous nous concentrons sur plusieurs aspects que nous faisons varier indépendamment des autres : la robustesse du modèle par rapport à la qualité de la traduction d’entrée, la quantité de données d’entraînement, les hyperparamètres utilisés pour l’affinage des modèles finaux (nombres d’exemples d’entraînement – *shots* –, taux d’apprentissage). Nos contributions à l’analyse AMR translingue sont les suivantes :

- Nous présentons le **premier modèle de méta-apprentissage pour l’analyse AMR translingue** ;
- Nous entraînons et évaluons notre modèle dans des langues peu ou pas explorées pour l’analyse AMR : le coréen, le croate, le français et le farsi ;
- Nous publions de nouvelles données d’évaluation en coréen et en croate, basées sur *Le Petit Prince* ;
- Nous publions un analyseur AMR multilingue qui peut être évalué dans de nombreuses langues en *zéro-shot*. Nous publions également le code permettant d’entraîner et d’évaluer le modèle.

2 Contexte scientifique

Analyse AMR translingue La tâche d’analyse AMR translingue a pour objectif de prédire des graphes AMR pour une langue cible alors que cette langue est absente des langues sources dans les données d’apprentissage. Les données d’entraînement AMR, des paires composées d’une phrase¹ et de son graphe AMR, ne sont disponibles qu’en anglais. Par conséquent, les approches précédentes ont soit cherché à créer des données d’entraînement artificielles dans la langue cible, soit à entraîner le modèle à l’aide de données AMR anglaises, puis à l’évaluer dans la langue cible (*zéro-shot*).

Damonte & Cohen (2018) traduisent automatiquement les données d’entraînement AMR en anglais vers la langue cible. Xu *et al.* (2021) et Bliloshi *et al.* (2020) utilisent des corpus parallèles (anglais - langue cible) et utilisent un analyseur AMR anglais pour obtenir des graphes AMR de la partie anglaise du corpus. Ils obtiennent finalement une nouvelle paire de texte cible et son graphe AMR correspondant. Inversement, dans l’approche *zéro-shot*, la tâche AMR en anglais est considérée

1. Le graphe AMR peut être utilisé au-delà du niveau de la phrase (O’Gorman *et al.*, 2018).

comme une tâche pivot, et la traduction multilingue entre l’anglais et la langue cible est ajoutée en tant que tâche auxiliaire (Procopio *et al.*, 2021; Xu *et al.*, 2021). La tâche auxiliaire permet à un modèle d’analyser les graphes AMR de la langue cible en *zéro-shot*.

Cependant, ces approches se concentrent sur un petit ensemble de langues pour lesquelles des données d’entraînement ou d’évaluation sont disponibles, ce qui motive nos efforts pour l’évaluation sur de nouvelles langues. Pour obtenir des données d’entraînement dans différentes langues, nous utilisons la traduction automatique comme Damonte & Cohen (2018). Nous évaluons ensuite notre modèle de manière *zéro-shot / few-shot* sur cinq langues : Chinois (sino-tibétain), Coréen (coréanique), et trois langues de trois branches de la famille des langues indo-européennes : le français (romane), le farsi (indo-iranienne) et le croate (slave).

Méta-apprentissage Le méta-apprentissage est un paradigme d’apprentissage qui permet à un modèle d’apprendre rapidement une nouvelle tâche avec seulement quelques exemples. Cela est possible grâce aux connaissances préalables que le modèle a acquises au cours d’une série de tâches différentes. Parmi les différentes approches du méta-apprentissage, la méthode basée sur l’optimisation est très utilisée dans les applications du TAL (Dingliwal *et al.*, 2021; Hua *et al.*, 2020; Bansal *et al.*, 2020) en raison de son efficacité. En particulier, le méta-apprentissage agnostique envers les modèles (MAML pour *model-agnostic meta learning* Finn *et al.*, 2017) est très utilisé (Nooralahzadeh *et al.*, 2020; Gu *et al.*, 2018; Singh *et al.*, 2022; Langedijk *et al.*, 2022).

L’idée derrière MAML est de trouver de bons paramètres initiaux θ qui peuvent être ajustés à de nouvelles tâches avec seulement quelques étapes d’optimisation et quelques exemples d’entraînement. Pour cela, MAML procède en *simulant l’entraînement et l’évaluation avec peu d’exemples* sur des tâches d’entraînement. En outre, le modèle est entraîné avec différentes tâches afin qu’il puisse apprendre à s’adapter rapidement à toutes les tâches similaires². Dans les applications translingues, chaque tâche correspond à une langue différente, ce qui est l’objet de notre étude (cf section 3).

L’approche la plus proche de la nôtre est celle de Langedijk *et al.* (2022), qui adoptent MAML pour l’analyse syntaxique en dépendances translingue. Ils appliquent MAML pour apprendre de bons paramètres initiaux à partir de langues sources, puis s’évaluent sur un ensemble disjoint de langues cibles. Dans nos travaux, nous nous concentrons plutôt sur une tâche d’analyse *sémantique*. De plus, ils disposent de données d’entraînement multilingues, alors que nos données n’existent qu’en anglais. AMR est un formalisme non ancré donc une phrase anglaise et sa traduction devraient avoir le même graphe AMR. Par conséquent, nous générons nos données multilingues par la traduction automatique des données anglaises. Une autre approche similaire à la nôtre est celle de Sherborne & Lapata (2023) qui ont appliqué le méta-apprentissage à l’analyse sémantique SQL translingue. Bien qu’utile pour représenter (et exécuter) des requêtes de base de données exprimées en langage naturel, SQL n’est pas un formalisme sémantique polyvalent comme AMR. À notre connaissance, notre travail est donc le premier à utiliser MAML dans le cadre de l’analyse AMR translingue.

3 Meta X-AMR

Analyse AMR seq2seq Trois approches d’analyse AMR sont très utilisées : l’analyse AMR par transitions (Damonte *et al.*, 2017), l’analyse AMR par graphe (*graph-based*, Zhang *et al.*, 2019; Cai

2. Tâches cibles avec une distribution similaire à celle des tâches sources.

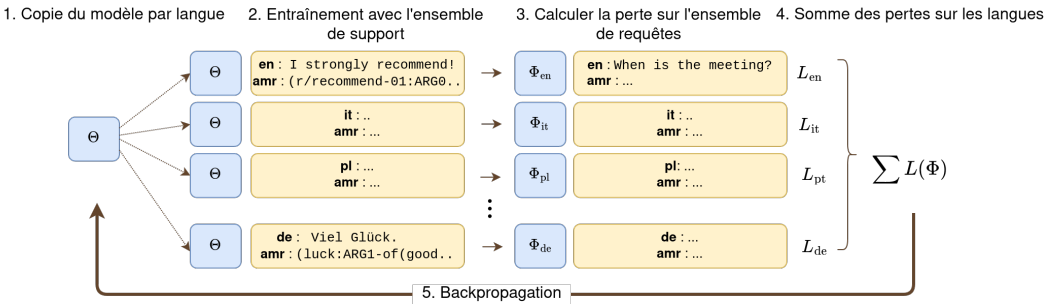


FIGURE 1 – Une étape d’entraînement MAML pour l’analyse AMR translingue.

& Lam, 2019), et l’analyse AMR seq2seq (Bevilacqua *et al.*, 2021). Nous utilisons cette dernière qui considère l’analyse AMR comme la génération d’un graphe AMR à partir de textes d’entrée à l’aide d’un modèle seq2seq. Dans cette approche, les graphes AMR doivent d’abord être linéarisés, c’est-à-dire représentés sous la forme d’une simple chaîne de caractères. Nous adoptons l’algorithme de parcours en profondeur pour la linéarisation comme Bevilacqua *et al.* (2021).

Nous utilisons le modèle mBart (Tang *et al.*, 2020)³ pour entraîner notre analyseur AMR multilingue, comme Procopio *et al.* (2021). Le modèle mBart est un *transformer* (Vaswani *et al.*, 2017) pré-entraîné qui se compose de plusieurs couches d’encodeurs et de décodeurs. Comme le résultat de ce modèle est un graphe linéarisé, nous restructurons le graphe AMR par des étapes de post-traitement pour l’évaluation. Nous utilisons le code de van Noord & Bos (2017)⁴ pour la linéarisation et la délinéarisation. Nous renvoyons les lecteurices à van Noord & Bos (2017) pour une description complète du processus.

MAML pour l’analyse AMR translingue Nous utilisons MAML (Finn *et al.*, 2017) pour entraîner notre analyseur AMR. L’objectif est d’entraîner un modèle qui s’adapte rapidement aux langues cibles avec aucun ou avec quelques exemples. La procédure d’entraînement est décrite ci-dessous et illustrée dans la Figure 1.

Étape 1 - Boucle interne : À chaque étape d’entraînement, le modèle initial (Θ) est copié une fois par langue i . Pour chaque langue i , $2 \times K$ exemples sont échantillonnés aléatoirement à partir de D_i^{train} et divisés en ensemble de support (*support set*) et ensemble de requête (*query set*) : K exemples pour chacun. En utilisant l’ensemble de support, le modèle est temporairement mis à jour avec une descente de gradient stochastique avec un taux d’apprentissage α (Eq. 1). Cette étape est répétée pendant P étapes d’adaptation afin d’obtenir Φ_i :

$$\Phi_i \leftarrow \Theta - \alpha \nabla_{\Theta} \mathcal{L}(\Theta_i). \quad (1)$$

Ensuite, la perte est calculée pour évaluer le modèle temporaire Φ_i sur l’ensemble de requêtes. La perte $\mathcal{L}_i(\Phi_i)$ est conservée pour l’étape suivante. L’ensemble de l’étape est appelé "boucle interne" et elle est répétée sur l’ensemble des tâches, c’est-à-dire pour le nombre de langues d’apprentissage. Ce nombre est désigné par I .

3. Nous utilisons le modèle facebook/mbart-large-50 via la plateforme de *transformers* (Wolf *et al.*, 2020).

4. <https://github.com/RikVN/AMR>

Étape 2 - Boucle externe : La somme de $\mathcal{L}_i(\Phi_i)$ est calculée sur l'ensemble des langues d'apprentissage afin de mettre à jour le modèle initial Θ par descente de gradient stochastique avec un taux d'apprentissage β . L'ensemble de cette étape est appelé "boucle externe"⁵ :

$$\Theta \leftarrow \Theta - \beta \sum_i \nabla_{\Phi_i} \mathcal{L}_i(\Phi_i). \quad (2)$$

Étape 3 : Répéter les étapes 1 et 2 jusqu'à ce que le nombre total d'étapes d'entraînement soit atteint.

Étape 4 : Une fois l'entraînement terminé, nous évaluons le modèle sur les langues cibles en *zéro-shot* ou *few-shot*. C'est-à-dire que le modèle est évalué sur de nouvelles langues cibles qui n'ont pas été vues pendant l'entraînement, sans affinage ou avec un affinage utilisant quelques exemples.

4 Expérimentations

4.1 Données

Notre méthode est similaire à celle de [Langedijk et al. \(2022\)](#) dans l'application du méta-apprentissage pour une tâche d'analyse translingue en *few-shot*. Pourtant, les données d'entraînement d'AMR ne sont disponibles qu'en anglais, alors qu'ils disposent de données d'entraînement multilingues pour l'analyse syntaxique en dépendance. Pour créer des données d'entraînement d'AMR multilingues, nous appliquons la traduction automatique comme dans les approches précédentes ([Damonte & Cohen, 2018](#); [Xu et al., 2021](#); [Blloshmi et al., 2020](#)). Nous utilisons DeepL⁶ pour la traduction automatique et traduisons les données d'entraînement AMR anglaises (LDC2020T02 [Knight et al., 2020](#)) en 13 langues : allemand, italien, roumain, finnois, russe, turc, japonais, tchèque, néerlandais, polonais, suédois, estonien et indonésien. Les 13 langues ont été choisies pour leur diversité linguistique et couvrent 5 familles de langues : indo-européen (germanique, roman, slave), ouralien, turcique, japonique et austronésien. Nous utilisons un total de 14 langues, dont l'anglais, pour nos données d'entraînement.

Pour sauvegarder le meilleur modèle pendant l'entraînement, nous évaluons notre modèle en *k-shot*. Pour cela, nous avons besoin des données de validation ainsi que de k exemples d'affinage dans la même langue. Pour les données de validation, nous utilisons les données de test en **espagnol** du corpus AMR 2.0 ([Damonte & Cohen, 2020](#)). Pour les données d'affinage, nous traduisons k exemples aléatoires issus des données de validation anglaises vers l'espagnol. Une fois l'entraînement terminée, nous évaluons notre modèle sur les langues cibles : le français, le chinois, le farsi, le coréen, et le croate. Pour les trois premières langues, nous utilisons le corpus AMR du Petit Prince annoté dans chaque langue, respectivement à partir de [Kang et al. \(2023\)](#), <https://amr.isi.edu/> et [Takhshid et al. \(2022\)](#)⁷. Pour le croate et le coréen, nous créons nos données de test en alignant manuellement le corpus du Petit Prince dans chaque langue sur les graphes AMR anglais correspondants. Nous mettons

5. Notez que dans l'équation 2, nous utilisons $\nabla_{\Phi_i} \mathcal{L}_i(\Phi_i)$ au lieu de $\nabla_{\theta} \mathcal{L}_i(\Phi_i)$ parce que nous appliquons MAML du premier ordre (*first-order MAML*) pour éviter le calcul très coûteux de la dérivée seconde.

6. <https://www.deepl.com>

7. L'ensemble de données original en farsi consiste en des graphes AMR dont les nœuds sont en farsi. Étant donné que nous utilisons des graphes AMR avec des nœuds en anglais, nous n'utilisons que les textes d'entrée du corpus et les graphes du corpus AMR en anglais.

ces données de test à la disposition du public ⁸.

4.2 Méta-entraînement et évaluation

Nous adoptons le modèle `mbart-large-50` (Tang *et al.*, 2020) de la bibliothèque `transformers` (Wolf *et al.*, 2020) pour entraîner notre analyseur AMR multilingue. Pour mettre en œuvre MAML, nous utilisons la bibliothèque `learn2learn` (Arnold *et al.*, 2020). Nous entraînons notre modèle pendant 30 000 étapes et l'évaluons toutes les 500 étapes avec les données de validation en espagnol. Nous mettons fin à l'entraînement si le score SMATCH (Cai & Knight, 2013) de validation ne s'améliore pas pendant plus de 7 500 étapes (interruption précoce de l'entraînement, *early stopping*). Pour la validation et le test, nous utilisons l'évaluation en k -shot, où le modèle est affiné avec k exemples avant d'être évalué sur l'ensemble de test/validation. Le nombre de cycles d'affinage, appelé étape d'adaptation, est noté P . Sauf indication contraire, nous fixons $P = 0$ et $k = 0$ (évaluation en 0 -shot). MAML nécessite deux taux d'apprentissage, un pour la boucle interne (α) et un pour la boucle externe (β). Nous avons effectué une recherche par quadrillage (*grid search*) pour identifier un ensemble optimal de taux d'apprentissage et avons utilisé $\alpha = 1 \times 10^{-5}$, $\beta = 3 \times 10^{-5}$ tout au long des expériences. Pour β , nous utilisons un taux d'apprentissage qui croît linéairement jusqu'à β sur les 1 500 premières étapes. Sauf indication contraire, nous appliquons 1×10^{-5} pour affiner le modèle pendant la validation et le test. À chaque étape d'itération pendant l'entraînement, $2 \times K$ sont échantillonnés pour former un lot (*batch*) à partir des ensembles de requêtes et de support. Par conséquent, la taille de lot N est égale à $2 \times K \times I$, où I représente le nombre de langues d'apprentissage. Par défaut, nous attribuons $K = 8$ et $I = 14$, sauf indication contraire. Nous présentons les scores d'évaluation en utilisant SMATCH (Cai & Knight, 2013), une métrique d'évaluation pour les graphes AMR.

4.3 Modèle de base avec apprentissage conjoint

Nous entraînons un modèle de base avec une méthode d'apprentissage conjoint où plusieurs tâches sont apprises simultanément afin d'améliorer les performances globales du modèle. Le modèle mBart est utilisé comme décrit à la section 4.2. Pour l'entraînement, nous utilisons les données d'entraînement AMR en 14 langues listées à la section 4.1. Nous concaténons ces données multilingues et à chaque étape d'itération, nous sélectionnons aléatoirement N exemples d'entraînement à partir de ces données pour calculer la perte et optimiser le modèle en conséquence. Le modèle est évalué en 0 -shot ou k -shot selon le cadre de l'expérience (les détails sont décrits dans chaque paragraphe de la section 5). Il convient de noter que notre objectif est de réaliser une étude comparative avec l'approche du méta-apprentissage. Par conséquent, sauf mention contraire, nous appliquons les mêmes hyperparamètres et la même méthode de test/évaluation pour les deux approches (par exemple, la taille des lots, la taille de k -shot). Cependant, alors que le méta-apprentissage nécessite deux taux d'apprentissage pour une boucle interne et une boucle externe, le modèle de base ne nécessite qu'un seul taux d'apprentissage pendant l'entraînement. Nous utilisons un taux d'apprentissage uniforme pour l'apprentissage 3×10^{-5} avec une croissance linéaire pendant les 1 500 premières étapes d'entraînement.

8. <https://github.com/Emvista/Meta-XAMR-2024>

	fr	zh	ko	fa	hr	avg
base_DeepL	56.3	45.6	42.1	46.3	51.4	48.3
base_mBart	56.2	44.5	41.2	46.1	51.3	47.8
MAML_DeepL	56.5	46.1	42.2	46.7	50.8	48.4
MAML_mBart	55.6	45.1	40.8	46.1	48.9	47.3

TABLE 1 – SMATCH en fonction de la source de traduction.

5 Questions de recherche et discussions

Nous examinons les points forts et les points faibles de notre méthode en répondant aux questions de recherche ci-dessous. Pour l'évaluation, nous faisons varier systématiquement cinq hyperparamètres individuellement tout en gardant les autres paramètres fixes et évaluons leur influence sur la performance du modèle. Nous évaluons chaque modèle en le comparant à son modèle de base adverse et apportons des éléments de réponses à 6 questions. Les questions Q1 et Q2 portent sur la manière dont les deux modèles réagissent à des facteurs spécifiques pendant la phase d'apprentissage, tandis que les questions Q3 à Q5 concernent les phases d'affinage fin et d'évaluation. Les discussions sur les questions conduisent à une discussion finale Q6 sur la question : le méta-apprentissage est-il l'approche optimale pour l'analyse AMR translingue ?

Q1 : Quelle est la robustesse du modèle en ce qui concerne la qualité de la traduction ? Pour évaluer l'impact de la traduction sur notre méthode, nous utilisons un autre modèle de traduction pour traduire nos données d'apprentissage. Plus précisément, nous utilisons les modèles de traduction mBart, provenant du hub Huggingface⁹, pour traduire nos données d'entraînement en 13 langues. Ensuite, nous utilisons ces données traduites pour entraîner les modèles MAML et les modèles de base. Enfin, nous comparons les résultats de l'évaluation de ces modèles avec ceux entraînés à l'aide du système de traduction automatique DeepL.

Resultats Pour les modèles MAML et les modèles de base, l'utilisation d'un modèle de traduction open-source mBart entraîne une baisse des performances (voir tableau 1). Dans les deux cas, le score SMATCH du coréen diminue le plus lors de l'utilisation du modèle de traduction mBart. Le modèle MAML est plus affecté par ce changement. En moyenne, le modèle de base perd 0,9 %, tandis que le modèle MAML perd 2,3 %. Ce résultat montre que le modèle de méta-apprentissage est plus sensible à la qualité des textes traduits que le modèle de base.

Q2 : Le modèle apprend-il efficacement dans des environnements avec moins de ressources ? Nous évaluons la robustesse de notre méthode dans les environnements avec moins de ressources où seule une petite partie des données d'apprentissage est disponible. À cette fin, nous échantillonnons au hasard 1 000 exemples pour chaque langue (les mêmes exemples pour toutes les langues) et utilisons uniquement ces données échantillonnées comme données d'entraînement.

Résultats Le tableau 2 illustre les scores SMATCH obtenus par les modèles MAML et les modèles de base dans différentes conditions d'entraînement : en utilisant les données complètes (base_full, MAML_full) ou en utilisant seulement 1 000 exemples (base_1000, MAML_1000). Sans surprise, les

9. <https://huggingface.co/facebook/mbart-large-50-many-to-many-mmt>

	fr	zh	ko	fa	hr	avg
base_full	56.3	45.6	42.1	46.3	51.4	48.3
base_1000	41.4	35.1	33.3	36.9	38.5	37.0
MAML_full	56.5	46.1	42.2	46.7	50.8	48.4
MAML_1000	38.9	33.9	32.8	36.1	35.0	35.3

TABLE 2 – SMATCH en fonction de la taille des données d’apprentissage.

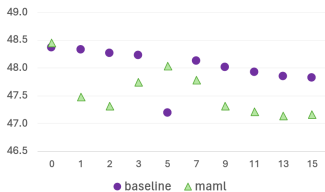


FIGURE 2 – Scores moyens de SMATCH sur les langues cibles en fonction des étapes d’adaptation.

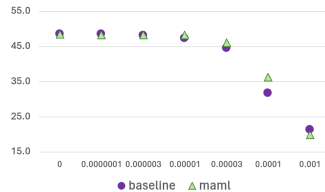


FIGURE 3 – Scores moyens de SMATCH sur les langues cibles en fonction du taux d’apprentissage de l’affinage fin.

performances des deux modèles ont considérablement diminué lorsqu’ils ont été entraînés sur un petit ensemble de données. Plus précisément, le modèle MAML connaît une baisse plus importante du score SMATCH, de 27 %, par rapport au modèle de base, qui a enregistré une baisse de 23,3 %. Cette différence suggère que le modèle MAML est plus sensible à la dégradation des performances dans les scénarios à faibles ressources.

Q3 : De combien d’étapes d’adaptation le modèle a-t-il besoin pour apprendre efficacement une nouvelle tâche ? Nous affinons nos modèles avec 32 exemples sur les langues cibles, puis nous les évaluons en langues cibles, pour l’évaluation *k-shot*. Étant donné que les données d’affinage ne sont pas disponibles pour les langues cibles, nous utilisons DeepL pour traduire les données de validation anglaises afin d’obtenir des données. Le modèle est ajusté de manière itérative avec les données de l’affinage et le nombre d’étapes d’adaptation est le nombre d’époques lors de l’affinage finale *k-shot*. Pour évaluer l’influence des étapes d’adaptation sur les performances du modèle, nous augmentons le nombre d’itérations et évaluons le modèle en conséquence. Nous échantillonons 32 exemples de manière aléatoire à trois reprises et utilisons le score moyen des trois processus d’évaluation. Le taux d’apprentissage de l’affinage est fixé à 1×10^{-5} dans toutes les expériences.

Résultats La figure 2¹⁰ représente visuellement les scores moyens des tests SMATCH dans les langues cibles. Lorsque l’étape d’adaptation est égale à 0, le modèle est évalué en mode *zéro-shot*. De manière surprenante, les résultats indiquent que les modèles MAML et celui de base sont moins efficaces après l’adaptation. Nous émettons l’hypothèse que le modèle pré-entraîné mBart possède déjà une connaissance suffisante de nos langues cibles, et qu’affiner le modèle avec seulement quelques exemples dans chaque langue peut nuire à la capacité du modèle. Cela peut également être attribué à la différence de domaine entre les données de d’affinage et les données de test. Le premier comprend du contenu provenant de domaines généraux tels que des actualités, des forums en ligne, et des journaux, tandis que les données de test se composent du roman écrit dans les années 1940, *Le Petit*

10. Les données numériques seront également présentées dans l’annexe en version finale

k_size	baseline	MAML
0	48.3	48.4
32	48.2	47.3
64	48.2	47.7
128	48.5	48.5

TABLE 3 – Scores SMATCH du modèle de base et du modèle MAML en fonction de k

	fr	zh	ko	fa	hr	avg
baseline	56.4	45.6	42.1	46.3	51.4	48.4
MAML	56.5	46.1	42.2	46.7	50.8	48.5

TABLE 4 – Scores SMATCH du modèle de base et du modèle MAML (évaluation à zéro-shot).

Prince. Par conséquent, le changement de domaine entre les 2 jeux de données peut avoir contribué à l’incapacité du modèle à s’adapter efficacement au domaine de test. Une autre hypothèse est la taille restreinte des données d’affinage, qui peut avoir entravé la performance du modèle, ou un taux d’apprentissage inadéquat conduisant aux résultats d’affinage indésirables. Nous approfondissons les hypothèses sur le taux d’apprentissage et la taille de k dans les questions suivantes.

Q4 : Taux d’apprentissage élevé ou faible pour l’affinage ? Pour examiner les performances du modèle en fonction des différents taux d’apprentissage, nous affinons notre modèle avec différents taux d’apprentissage. Nous appliquons les mêmes paramètres que dans Q4, tels que l’échantillonnage des données trois fois avec une taille k égale à 32.

Résultats La figure 3¹¹ présente une représentation visuelle des scores SMATCH moyens du test dans les différentes langues cibles. Le modèle de base et le modèle MAML présentent un comportement similaire, à savoir qu’un taux d’apprentissage plus faible permet d’obtenir de meilleurs résultats. Lorsque le taux d’apprentissage est égal à 0, c’est-à-dire lorsque le modèle n’est pas affiné, les deux modèles affichent les meilleures performances. Cela correspond aux résultats de Q3, mais on peut se demander pourquoi l’affinage dans les langues cibles ne conduit pas à un gain de performance. Cela peut être dû à la petite taille de k et dans la question suivante, nous évoquerons les résultats avec une plus grande taille de k .

Q5 : k -shot, quel est l’effet de la taille de k ? Pour répondre à cette question, nous utilisons différentes tailles de données d’affinage $k = 0, 32, 64, 128$. Comme pour Q3 et Q4, les données d’affinage sont échantillonnées trois fois et nous utilisons le score moyen. Nous appliquons le taux d’apprentissage 1×10^{-5} pour affiner les modèles.

Résultats Le tableau 3 montre que pour des valeurs de $32 \leq k \leq 128$, plus k est grand, plus le score est élevé. Cependant, à l’exception des modèles avec 128 exemples d’affinage, la plupart des modèles ne présentent pas d’amélioration par rapport à l’évaluation 0-shot. Il semble paradoxal qu’un modèle affiné soit moins performant qu’un modèle non affiné. Le modèle MAML est particulièrement affecté par l’étape d’affinage et présente une baisse de performance plus importante que le modèle de base. La baisse la plus importante est observée entre le modèle à 0 shot et le modèle à 32 shot, avec une différence de 2,3 %, alors que le modèle de base à 32 shot ne se dégrade que de 0,2 % par rapport au modèle à 0 shot. Par conséquent, cela nous amène à revoir les hypothèses discutées dans Q3 concernant la connaissance préalable du modèle mBart dans nos langues cibles et le changement de domaine entre les données d’affinage et celles du test.

11. Les données numériques seront également présentées dans l’annexe en version finale

Q6 : Comment l’analyse AMR translingue doit-elle être mise en place ? Le tableau 4 résume les scores SMATCH les plus élevés obtenus par les modèles de base et MAML lors de l’évaluation zéro-shot. La différence de performance entre ces modèles est marginale et varie en fonction de la langue cible. Par conséquent, il est difficile de tirer une conclusion définitive quant à la supériorité de l’une des méthodes. Cependant, notre examen nous a permis de constater que les modèles MAML présentent une plus grande sensibilité aux changements dans les types d’entrée et les tailles des données d’apprentissage. Notamment, leurs performances se détériorent de manière significative dans les scénarios avec peu de ressources ou lors de l’utilisation de différents modèles de traduction pour les entrées. En outre, des incohérences apparaissent lors de l’affinage du modèle avec différentes étapes d’adaptation, ce qui complique l’interprétation des résultats et rend difficile l’identification des perspectives d’amélioration. À l’inverse, nos observations indiquent qu’une approche simple d’apprentissage conjoint permet d’obtenir des performances comparables à celles du modèle MAML. Cela montre que la méthode d’apprentissage conjoint reste un point de départ solide pour l’analyse AMR translingue. Par conséquent, MAML n’apparaît pas comme la solution optimale pour cette tâche, compte tenu de ses performances instables.

Limites Notre modèle n’est pas plus performant qu’un simple modèle monolingue entraîné avec les données AMR dans la langue cible traduite par un système de traduction automatique. Cependant, notre approche peut être utilisée pour des langues peu dotées en ressources. En outre, nous n’avons pas appliqué de recherche par quadrillage (*grid-search*) pour trouver les meilleurs taux d’apprentissage pour les modèles de base et avons utilisé le même taux d’apprentissage que [Procopio et al. \(2021\)](#), qui a également utilisé mBart pour l’analyse AMR translingue en seq2seq. Cela a pu affecter les résultats en faveur du méta-apprentissage. Néanmoins, cela n’affecte pas notre conclusion de l’étude empirique qui révèle la faiblesse de l’approche du méta-apprentissage pour l’analyse AMR translingue.

6 Conclusion

Cette étude examine l’efficacité du méta-apprentissage par rapport à l’apprentissage conjoint dans l’analyse AMR translingue. Nous évaluons nos modèles dans des langues peu ou pas étudiées pour l’analyse AMR, notamment le français, le chinois, le coréen, le farsi et le croate. Pour faciliter l’évaluation, nous développons de nouveaux jeux de tests pour le coréen et le croate et publions les données pour promouvoir la diversité des langues d’évaluation pour l’analyse AMR translingue. Nous explorons différents contextes afin d’effectuer une analyse approfondie du méta-apprentissage par rapport à l’apprentissage conjoint. Nos résultats suggèrent que la méthode d’apprentissage conjoint est une approche de base robuste, tandis que le méta-apprentissage semble être une approche non optimale pour l’analyse AMR translingue en raison de ses performances peu robustes aux variations de configurations expérimentales.

Remerciements

Nous remercions les relecteurices anonymes pour leurs retours. Ces travaux ont bénéficié d’un accès aux moyens de calcul de l’IDRIS au travers de l’allocation de ressources 2023-AD011012853R1 attribuée par GENCI.

Références

- ARNOLD S. M. R., MAHAJAN P., DATTA D., BUNNER I. & ZARKIAS K. S. (2020). learn2learn : A library for Meta-Learning research. *arXiv*, **abs/2008.12284**.
- BANARESCU L., BONIAL C., CAI S., GEORGESCU M., GRIFFITT K., HERMIJAKOB U., KNIGHT K., KOEHN P., PALMER M. & SCHNEIDER N. (2013). Abstract Meaning Representation for sembanking. In A. PAREJA-LORA, M. LIAKATA & S. DIPPER, Éd., *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, p. 178–186, Sofia, Bulgaria : Association for Computational Linguistics.
- BANSAL T., JHA R. & MCCALLUM A. (2020). Learning to few-shot learn across diverse natural language classification tasks. In D. SCOTT, N. BEL & C. ZONG, Éd., *Proceedings of the 28th International Conference on Computational Linguistics*, p. 5108–5123, Barcelona, Spain (Online) : International Committee on Computational Linguistics. DOI : [10.18653/v1/2020.coling-main.448](https://doi.org/10.18653/v1/2020.coling-main.448).
- BEVILACQUA M., BLOSHMI R. & NAVIGLI R. (2021). One spring to rule them both : Symmetric amr semantic parsing and generation without a complex pipeline. *Proceedings of the AAAI Conference on Artificial Intelligence*, **35**(14), 12564–12573. DOI : [10.1609/aaai.v35i14.17489](https://doi.org/10.1609/aaai.v35i14.17489).
- BLOSHMI R., TRIPODI R. & NAVIGLI R. (2020). XL-AMR : Enabling cross-lingual AMR parsing with transfer learning techniques. In B. WEBBER, T. COHN, Y. HE & Y. LIU, Éd., *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, p. 2487–2500, Online : Association for Computational Linguistics. DOI : [10.18653/v1/2020.emnlp-main.195](https://doi.org/10.18653/v1/2020.emnlp-main.195).
- CAI D. & LAM W. (2019). Core semantic first : A top-down approach for AMR parsing. In K. INUI, J. JIANG, V. NG & X. WAN, Éd., *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, p. 3799–3809, Hong Kong, China : Association for Computational Linguistics. DOI : [10.18653/v1/D19-1393](https://doi.org/10.18653/v1/D19-1393).
- CAI D., LI X., HO J. C.-S., BING L. & LAM W. (2021). Multilingual AMR parsing with noisy knowledge distillation. In M.-F. MOENS, X. HUANG, L. SPECIA & S. W.-T. YIH, Éd., *Findings of the Association for Computational Linguistics : EMNLP 2021*, p. 2778–2789, Punta Cana, Dominican Republic : Association for Computational Linguistics. DOI : [10.18653/v1/2021.findings-emnlp.237](https://doi.org/10.18653/v1/2021.findings-emnlp.237).
- CAI S. & KNIGHT K. (2013). Smatch : an evaluation metric for semantic feature structures. In H. SCHUETZE, P. FUNG & M. POESIO, Éd., *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2 : Short Papers)*, p. 748–752, Sofia, Bulgaria : Association for Computational Linguistics.
- DAMONTE M. & COHEN S. (2020). Abstract meaning representation 2.0 - four translations ldc2020t07.
- DAMONTE M. & COHEN S. B. (2018). Cross-lingual Abstract Meaning Representation parsing. In M. WALKER, H. JI & A. STENT, Éd., *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, Volume 1 (Long Papers)*, p. 1146–1155, New Orleans, Louisiana : Association for Computational Linguistics. DOI : [10.18653/v1/N18-1104](https://doi.org/10.18653/v1/N18-1104).
- DAMONTE M., COHEN S. B. & SATTI G. (2017). An incremental parser for Abstract Meaning Representation. In M. LAPATA, P. BLUNSOM & A. KOLLER, Éd., *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics : Volume 1, Long Papers*, p. 536–546, Valencia, Spain : Association for Computational Linguistics.

DINGLIWAL S., GAO S., AGARWAL S., LIN C.-W., CHUNG T. & HAKKANI-TUR D. (2021). Few shot dialogue state tracking using meta-learning. In P. MERLO, J. TIEDEMANN & R. TSARFATY, Édts., *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics : Main Volume*, p. 1730–1739, Online : Association for Computational Linguistics. DOI : [10.18653/v1/2021.eacl-main.148](https://doi.org/10.18653/v1/2021.eacl-main.148).

FINN C., ABBEEL P. & LEVINE S. (2017). Model-agnostic meta-learning for fast adaptation of deep networks.

GU J., WANG Y., CHEN Y., LI V. O. K. & CHO K. (2018). Meta-learning for low-resource neural machine translation. In E. RILOFF, D. CHIANG, J. HOCKENMAIER & J. TSUJII, Édts., *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, p. 3622–3631, Brussels, Belgium : Association for Computational Linguistics. DOI : [10.18653/v1/D18-1398](https://doi.org/10.18653/v1/D18-1398).

HUA Y., LI Y.-F., HAFFARI G., QI G. & WU T. (2020). Few-shot complex knowledge base question answering via meta reinforcement learning. In B. WEBBER, T. COHN, Y. HE & Y. LIU, Édts., *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, p. 5827–5837, Online : Association for Computational Linguistics. DOI : [10.18653/v1/2020.emnlp-main.469](https://doi.org/10.18653/v1/2020.emnlp-main.469).

KANG J., COAVOUX M., SCHWAB D. & LOPEZ C. (2023). Analyse sémantique AMR pour le français par transfert translingue. In C. SERVAN & A. VILNAT, Édts., *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*, p. 55–62, Paris, France : ATALA.

KNIGHT K., BADARAU B., BARANESCU L., BONIAL C., BARDOCZ M., GRIFFITT K., HERMIAKOB U., MARCU D., PALMER M., O’GORMAN T. & SCHNEIDER N. (2017). Abstract meaning representation (amr) annotation release 2.0 - linguistic data consortium.

KNIGHT K., BADARAU B., BARANESCU L., BONIAL C., BARDOCZ M., GRIFFITT K., HERMIAKOB U., MARCU D., PALMER M., O’GORMAN T. & SCHNEIDER N. (2020). Abstract meaning representation (amr) annotation release 3.0 - linguistic data consortium.

LANGEDIJK A., DANKERS V., LIPPE P., BOS S., CARDENAS GUEVARA B., YANNAKOUKAKIS H. & SHUTOVA E. (2022). Meta-learning for fast cross-lingual adaptation in dependency parsing. In S. MURESAN, P. NAKOV & A. VILLAVICENCIO, Édts., *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, p. 8503–8520, Dublin, Ireland : Association for Computational Linguistics. DOI : [10.18653/v1/2022.acl-long.582](https://doi.org/10.18653/v1/2022.acl-long.582).

NOORALAHZADEH F., BEKOULIS G., BJERVA J. & AUGENSTEIN I. (2020). Zero-shot cross-lingual transfer with meta learning. In B. WEBBER, T. COHN, Y. HE & Y. LIU, Édts., *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, p. 4547–4562, Online : Association for Computational Linguistics. DOI : [10.18653/v1/2020.emnlp-main.368](https://doi.org/10.18653/v1/2020.emnlp-main.368).

O’GORMAN T., REGAN M., GRIFFITT K., HERMIAKOB U., KNIGHT K. & PALMER M. (2018). AMR beyond the sentence : the multi-sentence AMR corpus. In E. M. BENDER, L. DERCZYNSKI & P. ISABELLE, Édts., *Proceedings of the 27th International Conference on Computational Linguistics*, p. 3693–3702, Santa Fe, New Mexico, USA : Association for Computational Linguistics.

PROCOPIO L., TRIPODI R. & NAVIGLI R. (2021). SGL : Speaking the graph languages of semantic parsing via multilingual translation. In K. TOUTANOVA, A. RUMSHISKY, L. ZETTLEMOYER, D. HAKKANI-TUR, I. BELTAGY, S. BETHARD, R. COTTERELL, T. CHAKRABORTY & Y. ZHOU, Édts., *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, p. 325–337, Online : Association for Computational Linguistics. DOI : [10.18653/v1/2021.naacl-main.30](https://doi.org/10.18653/v1/2021.naacl-main.30).

SHERBORNE T. & LAPATA M. (2023). Meta-learning a cross-lingual manifold for semantic parsing. *Transactions of the Association for Computational Linguistics*, **11**, 49–67. DOI : [10.1162/tacl_a_00533](https://doi.org/10.1162/tacl_a_00533).

SHETH J., LEE Y.-S., FERNANDEZ ASTUDILLO R., NASEEM T., FLORIAN R., ROUKOS S. & WARD T. (2021). Bootstrapping multilingual AMR with contextual word alignments. In P. MERLO, J. TIEDEMANN & R. TSARFATY, Édts., *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics : Main Volume*, p. 394–404, Online : Association for Computational Linguistics. DOI : [10.18653/v1/2021.eacl-main.30](https://doi.org/10.18653/v1/2021.eacl-main.30).

SINGH S., WANG R. & HOU F. (2022). Improved meta learning for low resource speech recognition. In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, p. 4798–4802. DOI : [10.1109/ICASSP43922.2022.9746899](https://doi.org/10.1109/ICASSP43922.2022.9746899).

TAKHSHID R., SHOJAEI R., AZIN Z. & BAHRANI M. (2022). Persian abstract meaning representation. *arXiv*, **abs/2205.07712**.

TANG Y., TRAN C., LI X., CHEN P.-J., GOYAL N., CHAUDHARY V., GU J. & FAN A. (2020). Multilingual translation with extensible multilingual pretraining and finetuning. *arXiv*, **abs/2008.00401**.

VAN NOORD R. & BOS J. (2017). Neural semantic parsing by character-based translation : Experiments with abstract meaning representations. *Computational Linguistics in the Netherlands Journal*, **7**, 93–108.

VASWANI A., SHAZEER N., PARMAR N., USZKOREIT J., JONES L., GOMEZ A. N., KAISER L. U. & POLOSUKHIN I. (2017). Attention is all you need. In I. GUYON, U. V. LUXBURG, S. BENGIO, H. WALLACH, R. FERGUS, S. VISHWANATHAN & R. GARNETT, Édts., *Advances in Neural Information Processing Systems*, volume 30 : Curran Associates, Inc.

WOLF T., DEBUT L., SANH V., CHAUMOND J., DELANGUE C., MOI A., CISTAC P., RAULT T., LOUF R., FUNTOWICZ M., DAVISON J., SHLEIFER S., VON PLATEN P., MA C., JERNITE Y., PLU J., XU C., SCAO T. L., GUGGER S., DRAME M., LHOEST Q. & RUSH A. M. (2020). Huggingface’s transformers : State-of-the-art natural language processing.

XU D., LI J., ZHU M., ZHANG M. & ZHOU G. (2021). XLPT-AMR : Cross-lingual pre-training via multi-task learning for zero-shot AMR parsing and text generation. In C. ZONG, F. XIA, W. LI & R. NAVIGLI, Édts., *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1 : Long Papers)*, p. 896–907, Online : Association for Computational Linguistics. DOI : [10.18653/v1/2021.acl-long.73](https://doi.org/10.18653/v1/2021.acl-long.73).

ZHANG S., MA X., DUH K. & VAN DURME B. (2019). AMR parsing as sequence-to-graph transduction. In A. KORHONEN, D. TRAUM & L. MÀRQUEZ, Édts., *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, p. 80–94, Florence, Italy : Association for Computational Linguistics. DOI : [10.18653/v1/P19-1009](https://doi.org/10.18653/v1/P19-1009).