Actes de CORIA-TALN-RJCRI-RECITAL 2025
Marseille, France, 30 juin au 4 juillet 2025

Alignements entre attention et sémantique dans des modèles de langues pré-entraînés

Frédéric Charpentier[1, 2] Jairo Cugliari Duhalde[1] Adrien Guille[1]
(1) Laboratoire Eric, Université Lumière Lyon 2
(2) Cabot Financial, Lyon
jairo.cugliari@univ-lyon2.fr, adrien.guille@univ-lyon2.fr,
frederic.charpentier@cabotfinancial.fr

RÉSUMÉ

Les AMR (Abstract Meaning Representation) sont une structure destinée à coder la sémantique de phrases sous forme de graphes. Les mots des phrases correspondantes peuvent être alignés avec les sommets de l'AMR, de telle sorte que les relations sémantiques entre les mots puissent être mises en correspondance avec les rôles sémantiques lus sur les arcs de l'AMR. Le mécanisme d'attention d'un modèle de langue (ML) peut être modélisé comme le calcul de vecteurs descripteurs pour les arêtes d'un graphe complet dont les sommets sont les mots d'une phrase ou d'un paragraphe entier. Dans cet article, nous projetons les graphes AMR sur les graphes d'attention et concevons des méthodes supervisées pour détecter les relations sémantiques étiquetant les arêtes à partir des poids d'attention. Pour cela, nous mettons en œuvre des méthodes opérant soit sur les arêtes une à une, soit sur le graphe d'attention entier afin de comparer les capacités sémantiques de ML pré-entraînés. Il ressort de cette étude que l'encodeur bidirectionnel RoBERTA-base est meilleur que les décodeurs causaux, jusqu'à Llama 3 8B.

ABSTRACT

Aligning attention with semantics in pre-trained LLMs

Abstract Meaning Representations (AMRs) encode the semantics of sentences in the form of graphs. Words in the corresponding sentences can be aligned to vertices in the AMR, in such a way that semantic relations between words can be mapped from semantic roles read on the arcs of the AMR. The attention mechanism of a Language Model (LM) can be modelled as the computation of vectors describing edges on a complete graph whose vertices are words in a sentence or a whole paragraph. In this work, we map AMR graphs to Attention Graphs and devise supervised methods to detect the semantic relations labelling the edges from the attention weights. To do so, we implement methods operating either on single edges or on the whole attention graph in order to compare semantic capacities of several pretrained LMs. This study shows that the RoBERTA-base bidirectional encoder outperforms causal decoders up to Llama 3 8B.

MOTS-CLÉS : Sémantique, étiquetage sémantique, Représentation Abstraite de Signification, Attention, Réseaux de neurones sur graphes.

KEYWORDS: Semantics, Semantic Role Labeling, Abstract Meaning Representation, Attention, Graph Neural Networks.

Actes des 32ème Conférence sur le Traitement Automatique des Langues Naturelles (TALN), volume 1 : articles scientifiques originaux, pages 100–116.

1 Introduction

L'étiquetage des rôles sémantiques (SRL : Semantic Role Labelling Gildea & Jurafsky (2002)) est une tâche cruciale dans le domaine du traitement automatique du langage naturel (TALN) qui vise à identifier de manière univoque les rôles sémantiques des constituants d'une phrase, tels que qui a fait quoi à qui, quand et où. Par exemple, dans la phrase *Jean a cassé la fenêtre avec un marteau*, le SRL identifierait *Jean* comme l'agent, *fenêtre* comme le patient et *marteau* comme l'instrument. Les graphes AMR (Abstract meaning representation, Banarescu *et al.* (2013)) constituent un formalisme qui s'appuie sur le SRL pour encoder le sens d'une phrase dans un graphe enraciné, orienté et acyclique, dont les sommets représentent des instances de concepts (*Jean*, *casser*, *fenêtre*, *marteau*) et les arcs dirigés sont étiquetés avec les relations sémantiques entre eux (*agent*, *patient*, *instrument*).

Les modèles de langue basés sur des transformeurs, introduits par Vaswani *et al.* (2017), ont fait leurs preuves dans la résolution de divers problèmes liés au traitement automatique des langues. L'interprétabilité des calculs qu'ils mettent en œuvre fait encore l'objet de recherches actives, tant sur la syntaxe que sur la sémantique.

La résolution de tâches sémantiques est cruciale pour de nombreuses applications NLP. Dans cet article, nous présentons une étude visant à démontrer la capacité intrinsèque d'un modèle de langue de type transformeur à réaliser des tâches de SRL.

Nous commençons dans un cadre où nous classifions des paires de symboles [1] d'un transformeur en prédisant une relation sémantique, de façon similaire à l'approche de Charpentier *et al.* (2024). Puis, observant que le mécanisme d'attention dans un transformeur peut être considéré comme un moyen de construire le graphe complet d'une phrase, que nous appelons « graphe d'attention »,nous développons plus avant notre méthode dans un réseau neuronal sur graphe (Graph Neural Network : GNN). Nous décrivons la construction d'un GNN capable de classer des arêtes sélectionnées dans le graphe d'attention, en prenant en compte d'autres arêtes pour aider à la classification. Cette méthode ne repose pas sur la manipulation des plongements vectoriels des mots considérés, mais uniquement sur le mécanisme d'attention d'un grand modèle de langue. Il s'agit plus précisément d'une classification sur le graphe adjoint du graphe d'attention, où les sommets sont décrits par les vecteurs du mécanisme d'attention.

Dans la section 4, nous présentons les résultats des expériences menées sur notre ensemble de données et sur plusieurs modèles de langue. Nous mettons notre code à disposition [2] pour reproduire nos expériences.

1.1 Travaux associés

La capacité des têtes d'attention à saisir les caractéristiques syntaxiques a été examinée par Clark *et al.* (2019), qui ont étudié la capacité de certaines têtes d'attention dans le réseau BERT (Devlin *et al.*, 2019) à classer plusieurs relations syntaxiques et de coréférence sans réajuster [3] ce réseau pour une tâche spécifique. Luo (2021) a étudié la manière dont la grammaire des constituants est prise en compte par différentes têtes d'attention dans BERT. Ces travaux s'inscrivent dans le cadre de l'approche de "probing", présentée par Tenney *et al.* (2019b), que ces mêmes auteurs ont utilisée dans

1. symbole : *token*
2. `https://anonymous.4open.science/r/GAS-2DE3/`
3. réajustement : *fine-tuning*

[2019a] pour montrer les capacités du réseau BERT à effectuer en série et dans un ordre interprétable toutes les étapes traditionnelles du TALN, parmi lesquelles le SRL et la résolution de coréférence. Conia & Navigli (2022) ont également utilisé la méthode de probing spécifiquement dans l'étude du SRL, en comparant les capacités de BERT et RoBERTa à prédire le sens d'un prédicat et à identifier les rôles sémantiques qui lui sont liés.

L'utilisation de transformeurs pour construire automatiquement des graphes AMR a été marquée par Xu *et al.* (2020), qui les ont entraînés dans un modèle *séquence-à-séquence* pour construire des AMR à partir de phrases. Bevilacqua *et al.* (2021) ont conçu par la suite une approche symétrique pour effectuer la transduction AMR-texte et texte-AMR dans une architecture *séquence-à-séquence* basée sur des transformeurs, ce qui leur a permis de se débarrasser de la recatégorisation des graphes, c'est-à-dire d'heuristiques spécifiques conçues pour le jeu de données étudié. Zhou *et al.* (2021) ont présenté StructBART, une approche améliorée qui combine l'utilisation de modèles de transformeurs *séquence-à-séquence* avec une analyse syntaxique basée sur une machine à transitions, ce qui a permis d'obtenir de meilleures garanties quant à la forme des graphes.

Qorib *et al.* (2024) ont évalué la capacité de compréhension sémantique des modèles de langue à travers les tâches de désambiguïsation lexicale et des tâches "mots en contexte", afin de comparer les capacités des modèles de langue décodeurs à celles des encodeurs. Charpentier *et al.* (2024) ont étudié les capacités de RoBERTa et de GPT-2 à encoder les étiquettes sémantiques des graphes AMR dans leurs têtes d'attention, en utilisant des méthodes non supervisées et supervisées. Chizhikova *et al.* (2022) ont construit un modèle d'extraction de relations sémantiques pour évaluer les capacités de BERT dans une étude où seuls les scores d'attention sont pris en compte.

2 Structure du jeu de données

Dans cette section, nous décrivons notre méthode de classification des relations sémantiques entre paires de mots dans une phrase. Suivant Charpentier *et al.* (2024), nous avons utilisé les graphes AMR (Banarescu *et al.*, 2013) comme point de départ.

Un graphe AMR est un graphe acyclique orienté visant à représenter globalement le sens d'une phrase indépendamment des idiocrasies syntaxiques de sorte que différentes phrases de même sens puissent recevoir la même représentation. Les sommets d'un graphe AMR sont étiquetés avec des instances de concepts et chaque arc tracé entre deux sommets est étiqueté avec la relation sémantique induite par la phrase entre les concepts correspondants.

Un modèle de transformeur (Vaswani *et al.*, 2017) peut être décrit comme un réseau de neurones sur graphe opérant sur le graphe complet d'une phrase. En effet, le mécanisme d'attention opère sur chaque paire de symboles dans un texte, où les symboles sont des sous-unités de la phrase, représentant tout ou partie d'un mot. Nous décrivons donc une phrase comme un graphe non orienté complet de ses mots, appelé « graphe d'attention », où les arêtes sont décrites par des vecteurs descripteurs de $\mathbb{R}^d$ composés de tous les poids d'attention entre ces deux symboles. Pour un transformeur à m couches et n têtes d'attention par couche, la dimension de nos vecteurs d'attention est donc $d = m \times n$. Il convient de noter que, puisque le mécanisme d'attention est un calcul directionnel, d'un symbole requête à un symbole clé, chaque arête du graphe complet est en fait décrite par deux vecteurs de $\mathbb{R}^d$, afin de tenir compte de l'une ou l'autre direction dans le calcul de l'attention.

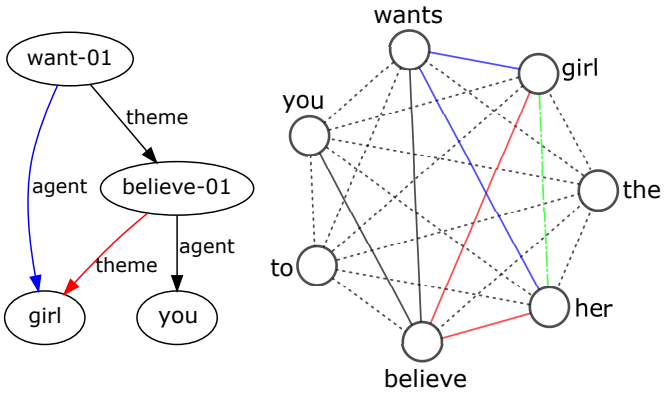


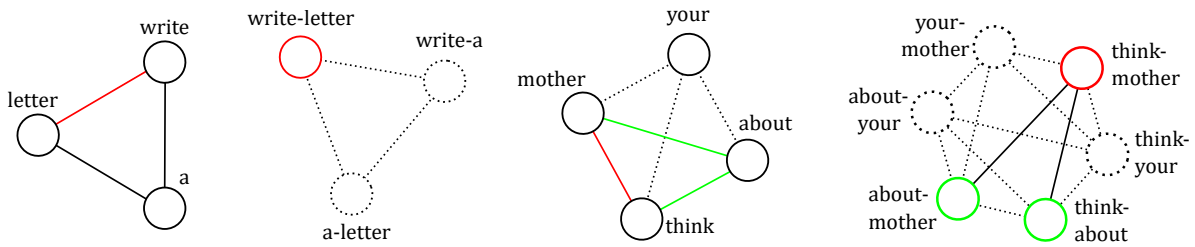
FIGURE 1 – Graphe AMR (à gauche) et graphe d'attention (à droite) pour la phrase "The Girl Wants You To Believe Her". Les symboles « her » et « girl » sont des coréférences du sommet AMR « girl ». L'arête verte entre les deux symboles est donc étiquetée avec le label spécial `{idem}`. Le sommet AMR « want-01 » est aligné sur le symbole « want ». L'arc bleu entre les sommets « want-01 » et « girl » est dupliqué à droite et associé aux arêtes (« wants », « girl ») et (« wants », « her »). De même, l'arête (« believe-01 », « girl ») de l'AMR est associée aux arêtes (« believe », « girl ») et (« believe », « her »). Les deux arcs noirs de l'AMR sont mis en correspondance avec les deux arêtes noires pleines du graphique de l'attention. Chaque arête pleine du graphe d'attention est étiquetée avec le rôle sémantique de l'arc correspondant dans l'AMR. Chaque arête reçoit également une autre étiquette indiquant la direction de l'arc, à l'exception de l'arête verte, qui n'en reçoit pas. Les arêtes en pointillé n'ont aucune étiquette.

2.1 Projection des graphes AMR sur les graphes d'attention

Contrairement à Charpentier *et al.* (2024), dont le travail s'est concentré sur l'extraction de paires de symboles des phrases, le calcul de l'attention et l'étiquetage avec les arcs alignés dans le graphe AMR, nous avons plutôt construit une correspondance complète entre les AMR et les graphes d'attention. Pour ce faire, nous avons eu recours à un travail antérieur de Blodgett & Schneider (2021), qui ont publié un algorithme basé sur la méthode EM permettant d'aligner les mots ou groupes de mots d'une phrase aux sommets ou sous-graphes du graphe AMR correspondant, et qui ont mis à disposition les données d'alignement calculées sur le jeu de données LDC2020T02 (Knight *et al.*, 2020). (Un jeu de près de 60 000 phrases anglaises et leurs représentations AMR)

Les étiquettes des arcs d'un graphe AMR sont un mélange hétéroclite d'étiquettes sémantiques de différents paradigmes. Certains rôles dits « centraux » sont étiquetés à partir des rôles de PropBank (Palmer *et al.*, 2005), un étiquetage qui a été critiqué par Di Fabio *et al.* (2019) pour son interprétabilité obscure : il consiste en effet en une simple énumération des arguments (`ARG0`, `ARG1`, etc.), qui ne marque pas explicitement le type de relation sémantique. Pour pallier cette limitation, Di Fabio *et al.* (2019) ont publié VerbAtlas, un inventaire manuel de verbes et de structures d'arguments avec des rôles sémantiques explicites inter-domaines.

— Nous avons réétiqueté les rôles centraux numérotés de PropBank dans l'AMR en utilisant les rôles de VerbAtlas, grâce à la table *PropBank to VerbAtlas*, publiée par Di Fabio *et al.* (2019).
— Entre deux symboles correspondant à deux mots alignés à deux sommets de l'AMR, nous avons étiqueté l'arête avec la relation sémantique dans l'AMR et ajouté une autre étiquette directionnelle pour noter la direction de l'arc.
— Entre deux symboles faisant partie du même mot, nous utilisons une étiquette spéciale



(a) Write A Letter (graphe direct) (b) Write A Letter (graphe adjoint) (c) Think About Your Mother (graphe direct) (d) Think About Your Mother (graphe adjoint)

FIGURE 2 – (a) et (b) : Graphes direct et adjoint pour la phrase *write a letter*, où la relation à classer (en rouge) se trouve entre les mots *write* et *letter*. (c) et (d) : Graphes d'attention directe et adjointe pour la phrase *Think about your mother*. La relation à classer est entre *think* et *mother* (en rouge). Le mot *about* semble être un pivot utile. Les arêtes utiles (en vert) du graphe direct relient le mot *about* aux mots *think* et *mother*. Dans le graphe adjoint, il s'agit des nœuds voisins du nœud à classer.

`{group}`, sans étiquette directionnelle. Nous avons distribué la relation sémantique de toutes les arêtes incidentes à ces deux mots.

— Parfois, en raison de la coréférence, deux mots sont alignés sur le même sommet ou sous-graphe AMR. Nous avons utilisé l'étiquette spéciale `{idem}` et aucune étiquette directionnelle entre les symboles concernés, et nous avons distribué les relations incidentes.

— Les AMR décrivent la conjonction et la disjonction à l'aide d'un sommet spécial étiqueté par *and* ou *or*. Ce sommet pointe vers les concepts combinés à l'aide de relations syntaxiques spéciales (`:op1`, `:op2`, ...). Nous avons choisi de redistribuer les relations sémantiques associées à ce sommet spécial et d'utiliser l'étiquette spéciale `{and}` ou `{or}` entre les concepts combinés. Là encore, aucune étiquette directionnelle n'est utilisée dans ce cas.

Le résultat est un ensemble phrases représentées dans des graphes d'attention entre symboles, où chaque arête est décrite par deux vecteurs dans $\mathbb{R}^d$, et porte deux types d'étiquettes : L'une pour représenter la relation sémantique (ou spéciale) entre les deux symboles ou les deux mots, l'autre pour représenter la direction de la relation, le cas échéant. Les deux étiquettes sont facultatives et de nombreuses arêtes du graphe complet n'en ont pas. (Voir la figure 1 pour une illustration).

Notre solution pour l'étiquetage sémantique des rôles se résume aux étapes suivantes : **1.** Pour chaque phrase, calculer les poids d'attention de chaque symbole par rapport aux autres en utilisant un modèle de langage, et utiliser ces poids pour décrire les arêtes d'un graphe d'attention. **2.** En utilisant l'alignement entre les sommets de l'AMR et les symboles de la phrase, et en appliquant les transformations décrites ci-dessus, attribuer des étiquettes sémantiques et directionnelles à certaines des arêtes du graphe d'attention. **3.** Entraîner les modèles décrits dans la section suivante à prédire les étiquettes des arêtes.

3 Modèles

Chaque arête du graphe d'attention porte au plus deux étiquettes, il y a donc deux problèmes à résoudre : **1.** Prédire l'étiquette sémantique de l'arête, **2.** Prédire l'étiquette directionnelle de l'arête, ce qui correspond à prédire quel sommet est la source et quel sommet est la cible de l'arc dans l'AMR

projeté. [4]

3.1 Prédiction en fonction des descripteurs des arêtes

Dans cette section, nous décrivons la manière la plus simple de résoudre les deux problèmes mentionnés ci-dessus. Supposons que nous voulions prédire la relation sémantique particulière entre les mots *write* et *letter* dans la phrase simple « *write a letter* ». L'idée de base est d'utiliser les données calculées par le modèle de langage représentant tous les poids d'attention du symbole *write* au symbole *letter*, ainsi que les poids d'attention du symbole *letter* au symbole *write*. Cela revient à n'utiliser que les deux vecteurs descripteurs de l'arête (*write*, *letter*) dans le graphe d'attention pour classer son étiquette, sans tenir compte du reste du graphe. Dans ce contexte, la structure du graphe n'est pas encore utilisée.

Pour résoudre le problème **1.**, on utilise un classificateur symétrique bi-affine de rang faible, décrit dans les annexes A.1 et A.2. Pour résoudre le problème **2.**, on utilise un classificateur antisymétrique bi-affine de rang faible, décrit dans les annexes A.3 et A.4.

3.2 Modèles sur graphe

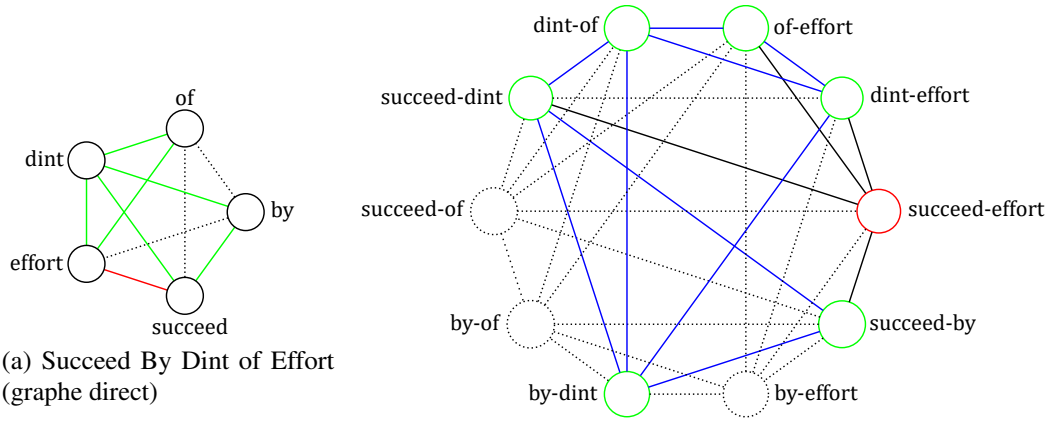
Ce paragraphe décrit notre idée principale qui consiste à utiliser les relations d'ordre supérieur à deux entre les symboles pour prédire les étiquettes.

Si nous devions prédire la relation sémantique entre les mots *think* et *mother* dans la phrase *think about your mother*, nous pourrions alimenter les modèles biaffines de la section 3.1 avec les vecteurs représentant l'attention entre les symboles *mother* et *think*. L'intuition suggère cependant que l'utilisation de la préposition *about* (agissant comme un pivot) pourrait aider le processus de prédiction. Par conséquent, l'idée principale est de prédire la relation sémantique entre *think* et *mother* en utilisant les vecteurs décrivant non seulement l'attention dans la paire (*think*, *mother*), mais aussi les paires (*think*, *about*) et (*about*, *mother*), utilisant ainsi la relation d'ordre trois entre les symboles *think*, *about*, et *mother*. En outre, dans le cas plus réaliste où le symbole pivot n'est pas connu à l'avance, il faut considérer chaque sommet du graphe d'attention complet et essayer les vecteurs descripteurs des arêtes reliant ce sommet et les deux sommets *think* et *mother*. Ce procédé peut être exprimé de manière élégante en considérant le graphe adjoint (Harary & Norman (1960)).

Le graphe adjoint d'un graphe non orienté $G_0 = (\mathcal{V}_0, \mathcal{E}_0)$ est un graphe $G_1 = (\mathcal{V}_1, \mathcal{E}_1)$, où $\mathcal{V}_1 = \mathcal{E}_0$, et où deux nœuds sont adjacents si et seulement s'ils partagent un sommet dans $\mathcal{V}_0$. Dans ce cadre, les arêtes reliant *think*, ou *mother* à un symbole pivot candidat sont exactement les voisins de (*think*, *mother*) dans le graphe adjoint du graphe complet. (voir figure 2). Dans certains cas plus complexes, il peut être utile de considérer des arêtes plus éloignées de l'arête à classer. Si nous voulons classer l'arête (*succeed*, *efforts*) dans la phrase *Succeed by dint of efforts*, nous pourrions utiliser les informations données par les arêtes (*succeed*, *by*), (*dint*, *effort*), (*succeed*, *dint*), mais aussi considérer les deux mots (*by*, *dint*), ne partageant aucun sommet avec l'arête à classer. Dans le graphe adjoint, cela correspond à prendre en compte les voisins de second ordre du nœud considéré. (voir figure 3.)

Pour choisir et prendre en compte les arêtes candidates, nous utilisons un GNN sur le graphe adjoint du graphe d'attention. Il se compose d'une ou plusieurs couches GATv2 (Brody *et al.*, 2022). Pour

4. Pour cette tâche, on a pris soin de redresser les étiquettes inversées, du type `ARG0-of`



(a) Succeed By Dint of Effort (graphe direct)

(b) Succeed By Dint of Effort (graphe adjoint)

FIGURE 3 – Graphes d'attention direct et adjoint pour la phrase *Succeed by dint of effort*. L'arête à classer est (*succeed*, *effort*) (en rouge). Des informations intéressantes peuvent être recueillies à partir des arêtes en vert. (Nœuds verts dans le graphe adjoint). Certains sont des voisins de second ordre du nœud considéré dans le graphe linéaire, comme (*by*, *dint*).

chaque sommet d'un graphe, une telle couche calcule une combinaison linéaire des caractéristiques de ce sommet et des caractéristiques de ses voisins en utilisant un mécanisme d'attention pour dériver les coefficients linéaires. Pour mieux décrire notre idée, nous commençons par donner un formalisme rapide du concept de couche GATv2 dans le cas particulier d'un *graphe régulier*.

3.2.1 Couche GATv2 dans un graphe régulier

Un *graphe régulier* est un graphe dont tous les sommets ont le même degré. Le graphe adjoint d'un graphe complet à n sommets est un graphe régulier de degré $2n-4$.

Si chaque nœud du graphe linéaire $(\mathcal{V}, \mathcal{E})$ est décrit par un vecteur descripteur de $\mathbb{R}^d$, le mécanisme GATv2 qui transforme les descripteurs d'un nœud peut être conceptualisé comme une fonction descriptrice π, paramétrée par θ, telle que :

$$\pi_\theta : \quad \mathbb{R}^d \times (\mathbb{R}^d)^{2n-4} \to \mathbb{R}^{d'}, \quad (\boldsymbol{v}, \boldsymbol{w}_1, \cdots, \boldsymbol{w}_{2n-4}) \mapsto \pi_\theta(\cdots) = \boldsymbol{h} \in \mathbb{R}^{d'},$$

où $\boldsymbol{v}$ est le vecteur de caractéristiques du nœud à transformer, et $\boldsymbol{w}_1 \cdots \boldsymbol{w}_{2n-4}$ sont les vecteurs de caractéristiques de ses voisins. La fonction π_θ est invariante par rapport à toute permutation des vecteurs voisins $\boldsymbol{w}_1 \cdots \boldsymbol{w}_{2n-4}$. L'ensemble de la couche GATv2 est le résultat de l'application de π_θ à chaque nœud du graphe adjoint $(\mathcal{V}, \mathcal{E})$. Nous pouvons conceptualiser ce processus à l'aide d'une fonction $\phi : \mathcal{V} \to \mathbb{R}^d,\ \kappa \mapsto \phi(\kappa)$, qui agit comme un paramètre pour la couche GATv2 tel que :

$$\mathrm{gat}_{\theta,\phi} : \quad \mathcal{V} \longrightarrow \mathbb{R}^{d'}, \quad \kappa \longmapsto \pi_\theta(\phi(\kappa), \phi(\mathcal{N}(\kappa))),$$

où $\mathcal{N}(\kappa)$ est l'ensemble des voisins de κ.

3.2.2 Classificateur sur graphe d'attention

Dans notre problème particulier, les nœuds du graphe adjoint de notre graphe d'attention ne sont pas décrits par un seul vecteur de $\mathbb{R}^d$, mais par deux. Plus formellement, nous pouvons définir deux fonctions descriptrices $\phi_1 : \mathcal{V} \longrightarrow \mathbb{R}^d$ et $\phi_2 : \mathcal{V} \longrightarrow \mathbb{R}^d$, qui associent chaque sommet κ à deux vecteurs $\phi_1(\kappa)$ et $\phi_2(\kappa)$.

La première couche de notre modèle est une couche symétrique : $\psi_1 : \mathcal{V} \longrightarrow \mathbb{R}^{d_1}$, où $\psi_1(\kappa) = \big((\phi_1(\kappa) + \boldsymbol{b}_i)^\top \cdot M_i \cdot (\phi_2(\kappa) + \boldsymbol{b}_i)\big)_{i\in\{1,\cdots,d_1\}}$. Cette couche permet d'attribuer un seul vecteur descripteur à chaque nœud, indépendamment de toute permutation de $\phi_1(\kappa)$ et $\phi_2(\kappa)$ pour tout nœud particulier κ. La deuxième couche est une couche GATv2 :

$$\psi_2 : \quad \mathcal{V} \longrightarrow \mathbb{R}^{d_2}, \quad \kappa \longmapsto \psi_2(\kappa) = \mathrm{gat}_{\theta_2,\psi_1}(\kappa),$$

où $\mathrm{gat}_{\theta_2,\psi_1}(s) = \pi_{\theta_2}(\psi_1(s), \psi_1(\mathcal{N}(s)))$. Les couches suivantes sont des couches GATv2, paramétrées par la fonction ψ de la couche précédente :

$$\psi_i : \quad \mathcal{V} \longrightarrow \mathbb{R}^{d_i}, \quad \kappa \longmapsto \mathrm{gat}_{\theta_i,\psi_{i-1}}(\kappa).$$

La dernière couche est un classificateur linéaire, et l'ensemble du modèle est insensible à la permutation de $\phi_1(\kappa)$ et $\phi_2(\kappa)$ pour tout $\kappa \in \mathcal{V}$. Pour nos expériences décrites ci-après, nous avons mis en œuvre exactement deux couches GATv2.

4 Expériences

Nous avons testé nos méthodes sur quelques modèles de langue : RoBERTa-base (Liu *et al.*, 2019), GPT-2 (Radford *et al.*, 2019), ModernBERT base, et trois modèles de la suite Llama 3 (Grattafiori *et al.*, 2024) : Llama-3.2-1B, Llama-3.2-3B, et Llama-3.1-8B. Nous avons choisi 15 des rôles sémantiques les plus utiles de notre ensemble de données, qui correspondent à la projection sur VerbAtlas des rôles choisis par Charpentier *et al.* (2024). La liste des rôles et leurs fréquences relatives sont disponibles dans le tableau de l'annexe B. Contrairement à Charpentier *et al.* (2024), nous n'avons pas restreint nos modèles à classer les relations entre symboles correspondant à des mots entiers. La tâche est ainsi plus difficile, puisqu'on tente de classer des relations entre certains symboles qui ne sont que des fractions de mots, certains pouvant être très peu porteurs de sens.

Comme nous l'avons décrit dans la section 2.1, nous avons ajouté des étiquettes spéciales pour tenir compte de la séparation des mots entre plusieurs symboles, ainsi que la coréférence, la conjonction et la disjonction. Ces étiquettes sont au nombre de six : `{and-or}`, `{and}`, `{or}`, `{group}`, `{idem}`, `{inter}`, (la dernière étant utilisée pour les concepts associés à des prépositions telles que « entre » ou « parmi »). En les incluant, le nombre de classes sélectionnées atteint 21. Notre entraînement s'effectue sur un jeu d'entraînement, en surveillant l'exactitude équilibrée (*macro-averaged accuracy*) calculée périodiquement sur les 21 classes, sur un jeu de validation.

La détection des six étiquettes est parfois un problème trivial, (notamment l'étiquette `{group}`, qui relie deux symboles du même mot, c'est pourquoi nous ne mesurons pas la performance de nos modèles sur ces six étiquettes supplémentaires, ce qui donnerait des résultats trop optimistes. Les performances, reportées dans le tableau 1 indiquent l'exactitude équilibrée calculée sur un jeu de test sur la restriction aux quinze classes sémantiques décrites à l'annexe B.

Modèle de langue	Dim. attention	Rang	Exac. équil. biaffine	Exac. équil. GNN	Gain biaff.→GNN	Nb params GNN
GPT-2	144	8	49,9%	63,9%	1,28	275×10^3
RoBERTa base	144	8	54,5%	**69,7%**	**1,28**	275×10^3
ModernBERT base	264	8	58,2%	65,7%	1,13	916×10^3
ModernBERT base	264	16	58,1%	**67,3%**	**1,16**	$1{,}48 \times 10^6$
Llama 1B	512	16	62,5%	66,1%	1,06	$2{,}50 \times 10^6$
Llama 3B	672	8	62,4%	68,9%	1,10	$2{,}50 \times 10^6$
Llama 8B	1024	16	64,4%	66,0%	1,02	$2,71 \times 10^6$

TABLE 1 – Résultats des expériences. Voir §4

4.1 Classificateur bi-affine

Cette section présente les résultats obtenus sur les modèles symétriques biaffines de faible rang. Pour les deux plus petits modèles, GPT-2 et RoBERTa base, Nous avons observé une augmentation de l' exactitude lorsque nous avons augmenté jusqu'à 8 le rang des matrices. Après $r = 8$, les métriques plafonnent. Toutefois, nous avons préféré utiliser un rang un peu plus élevé $r = 16$ pour les modèles plus grands, afin de mieux capturer l'information diluée dans des vecteurs descripteurs de plus en plus grands.

Les résultats sont lisibles dans la colonne « exac. équil. biaffine » du tableau 1. On constate que l'exactitude équilibrée est une fonction croissante de la dimension des vecteurs d'attention, qui dépend directement de la taille des modèles. On voit aussi qu'il y a une grande différence entre les deux plus petits modèles, GPT-2 et RoBERTa base, en faveur du second. Nous supposons que ce phénomène est dû à l'attention causale du modèle décodeur GPT-2 qui n'exploite que le calcul d'attention d'un symbole sur les symboles antérieurs, alors qu'un modèle bidirectionnel comme RoBERTa base exploite l'attention dans les deux directions.

4.2 Classificateur sur graphe

Nous avons comparé ces résultats à ceux que donne le GNN que nous avons décrit à la section 3.2.1. Le tableau 1 donne les résultats dans la colonne « Exac. équil GNN ». Pour chaque modèle de langue, il s'agit d'un réseau dont la première couche partage avec l'expérience de classification biaffine le rang indiqué dans la colonne « rang ». Le réseau est pourvu de deux couches GAT-v2 dont la dimension cachée est indiquée dans la colonne « dim cachée ».

On constate que notre approche permet d'obtenir des gains importants dans les performances. (gains calculés entre les performances du GNN rapportées aux performances du classificateur biaffine de même rang.) On obtient un gain de 1,28 pour les plus petits modèles, ce qui prouve l'utilité d'examiner des relations d'ordre supérieur que permet notre approche. Toutefois le gain diminue avec les plus grands réseaux, prouvant ainsi expérimentalement que les plus grands réseaux encodent déjà suffisamment la sémantique dans les relations du premier ordre entre symboles. (Llama-8B, le modèle le plus grand, montre un gain de performance de 1,02 seulement.)

Il est remarquable de constater que les deux meilleurs résultats de tout le tableau sont ceux obtenus

avec les modèles RoBERTa base et ModernBERT base, qui sont les seuls réseaux à tirer parti du calcul bidirectionnel de l'attention, alors que tous les autres emploient un masquage causal qui interdit de tirer parti de l'attention sur des symboles ultérieurs dans la phrase. Notre expérience tend à prouver que ce masquage constitue un obstacle à l'encodage de la sémantique dans l'attention d'un modèle transformeur causal.

4.3 GNN à zéro couche GATv2

L'écart de performance entre nos modèles à deux couches GATv2 et nos classificateurs symétriques biaffines ne s'explique pas uniquement par la présence des couches GATv2 : Pour prédire parmi c classes avec un classificateur symétrique, il suffit de c matrices de rang r directement connectées à une couche softmax. Dans notre GNN, la première couche est composée de h matrices de rang r, suivie de deux couches GATv2, suivies d'un classificateur linéaire. Même sans connecter les couches GATv2, notre GNN a beaucoup plus de paramètres que le classificateur symétrique. Pour prouver l'efficacité des couches GAT, nous avons mené une étude d'ablation, dans laquelle nous avons remplacé les deux couches GATv2 par une simple activation reLU, sans paramètres. Nous avons entraîné notre modèle ainsi amputé sur les deux meilleurs modèles de langue pour l'expérience de classification biaffine : RoBERTA base et ModernBERT base en rang 16.

On obtient une exactitude équilibrée de 52,0% sur RoBERTa base, bien en deçà de la valeur de 69,7% obtenue avec le modèle GATv2 normal, et même un peu en dessous des performances du classificateur biaffine. Pour ModernBERT base, l'exactitude équilibrée est de 58,7%, ce qui constitue un niveau comparable au classificateur biaffine, et perd tout le gain dû aux couches GATv2.

4.4 Classificateur d'orientation

Évoquons brièvement encore le classificateur d'orientation des arcs de l'AMR, évoqué dans la section 3.1, et décrit dans les annexes A.3 et A.4. Il convient de noter la possibilité particulière de s'entraîner sur un ensemble de données équilibré, en permutant d'abord de manière aléatoire les deux vecteurs pour chaque nœud, ainsi que l'étiquette indiquant la direction de l'arête. Les meilleurs résultats ont été obtenus avec le rang 7, avec une exactitude de 86% sur le modèle RoBERTa base, et de 87% sur le modèle GPT-2. Ces résultats montrent que le problème de la classification de la direction d'un arc dans un AMR n'est pas un problème très difficile, et ne nous ont pas incité à concevoir un classificateur sur graphes pour l'aborder.

5 Conclusion

Dans cette étude, nous avons introduit une méthode basée sur les graphes pour évaluer la capacité intrinsèque de différents modèles de langue à effectuer des tâches de SRL. Notre méthode exploite les poids d'attention calculés par les modèles de langue utilisés sans réajustement, et nous n'avons pas recours aux plongements des mots. On a montré expérimentalement d'une part qu'une classification simple basée sur les vecteurs d'attention entre les deux symboles d'une paire est d'autant plus efficace que le modèle de langue utilisé est grand ; d'autre part, en utilisant le graphe attentionnel complet entre les symboles du texte pour classer les paires, notre méthode est capable de meilleures prédictions

à partir de relations entre plus de deux mots dans le texte. Bien que l'avantage de notre méthode soit moins sensible sur de grands modèles de langue, qui semblent encoder la sémantique directement dans des relations d'ordre deux, notre approche a permis les meilleurs résultats sur le plus petit modèle RoBERTa base. Nos expériences ont également confirmé le fait que les modèles de langue bidirectionnels montrent une meilleure capacité à distinguer les relations sémantiques que les modèles causaux. De plus, nous avons présenté une méthode générale qui peut être utilisée de manière plus générale dans des problèmes impliquant des paires de mots, pour d'autres tâches que la sémantique.

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 A. PAREJA-LORA, M. LIAKATA & S. DIPPER, Éds., *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, p. 178–186, Sofia, Bulgaria : Association for Computational Linguistics.

BEVILACQUA M., BLLOSHMI 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.

BLODGETT A. & SCHNEIDER N. (2021). Probabilistic, Structure-Aware Algorithms for Improved Variety, Accuracy, and Coverage of AMR Alignments. In *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. 3310–3321, Online : Association for Computational Linguistics. DOI : 10.18653/v1/2021.acl-long.257.

BRODY S., ALON U. & YAHAV E. (2022). How Attentive are Graph Attention Networks ?

CHARPENTIER F., CUGLIARI J. & GUILLE A. (2024). Exploring Semantics in Pretrained Language Model Attention. In D. BOLLEGALA & V. SHWARTZ, Éds., *Proceedings of the 13th Joint Conference on Lexical and Computational Semantics (*SEM 2024)*, p. 326–333, Mexico City, Mexico : Association for Computational Linguistics. DOI : 10.18653/v1/2024.starsem-1.26.

CHIZHIKOVA A., MURZAKHMETOV S., SERIKOV O., SHAVRINA T. & BURTSEV M. (2022). Attention understands semantic relations. In N. CALZOLARI, F. BÉCHET, P. BLACHE, K. CHOUKRI, C. CIERI, T. DECLERCK, S. GOGGI, H. ISAHARA, B. MAEGAARD, J. MARIANI, H. MAZO, J. ODIJK & S. PIPERIDIS, Éds., *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, p. 4040–4050, Marseille, France : European Language Resources Association.

CLARK K., KHANDELWAL U., LEVY O. & MANNING C. D. (2019). What does BERT look at ? An analysis of BERT's attention. In T. LINZEN, G. CHRUPAŁA, Y. BELINKOV & D. HUPKES, Éds., *Proceedings of the 2019 ACL workshop BlackboxNLP : Analyzing and interpreting neural networks for NLP*, p. 276–286, Florence, Italy : Association for Computational Linguistics. DOI : 10.18653/v1/W19-4828.

CONIA S. & NAVIGLI R. (2022). Probing for Predicate Argument Structures in Pretrained Language Models. In S. MURESAN, P. NAKOV & A. VILLAVICENCIO, Éds., *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, p. 4622–4632, Dublin, Ireland : Association for Computational Linguistics. DOI : 10.18653/v1/2022.acl-long.316.

DEVLIN J., CHANG M.-W., LEE K. & TOUTANOVA K. (2019). BERT : Pre-training of deep bidirectional transformers for language understanding. In J. BURSTEIN, C. DORAN & T. SOLORIO,

Éds., *Proceedings of the 2019 conference of the north American chapter of the association for computational linguistics : Human language technologies, volume 1 (long and short papers)*, p. 4171–4186, Minneapolis, Minnesota : Association for Computational Linguistics. DOI : 10.18653/v1/N19-1423.

DI FABIO A., CONIA S. & NAVIGLI R. (2019). VerbAtlas : A Novel Large-Scale Verbal Semantic Resource and Its Application to Semantic Role Labeling. In K. INUI, J. JIANG, V. NG & X. WAN, Éds., *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. 627–637, Hong Kong, China : Association for Computational Linguistics. DOI : 10.18653/v1/D19-1058.

GILDEA D. & JURAFSKY D. (2002). Automatic labeling of semantic roles. *Computational Linguistics*, **28**(3), 245–288. DOI : 10.1162/089120102760275983.

GRATTAFIORI A., DUBEY A., JAUHRI A., PANDEY A., KADIAN A., AL-DAHLE A., LETMAN A., MATHUR A., SCHELTEN A., VAUGHAN A., YANG A., FAN A., GOYAL A., HARTSHORN A., YANG A., MITRA A., SRAVANKUMAR A., KORENEV A., HINSVARK A., RAO A., ZHANG A., RODRIGUEZ A., GREGERSON A., SPATARU A., ROZIERE B., BIRON B., TANG B., CHERN B., CAUCHETEUX C., NAYAK C., BI C., MARRA C., MCCONNELL C., KELLER C., TOURET C., WU C., WONG C., FERRER C. C., NIKOLAIDIS C., ALLONSIUS D., SONG D., PINTZ D., LIVSHITS D., WYATT D., ESIOBU D., CHOUDHARY D., MAHAJAN D., GARCIA-OLANO D., PERINO D., HUPKES D., LAKOMKIN E., ALBADAWY E., LOBANOVA E., DINAN E., SMITH E. M., RADENOVIC F., GUZMÁN F., ZHANG F., SYNNAEVE G., LEE G., ANDERSON G. L., THATTAI G., NAIL G., MIALON G., PANG G., CUCURELL G., NGUYEN H., KOREVAAR H., XU H., TOUVRON H., ZAROV I., IBARRA I. A., KLOUMANN I., MISRA I., EVTIMOV I., ZHANG J., COPET J., LEE J., GEFFERT J., VRANES J., PARK J., MAHADEOKAR J., SHAH J., VAN DER LINDE J., BILLOCK J., HONG J., LEE J., FU J., CHI J., HUANG J., LIU J., WANG J., YU J., BITTON J., SPISAK J., PARK J., ROCCA J., JOHNSTUN J., SAXE J., JIA J., ALWALA K. V., PRASAD K., UPASANI K., PLAWIAK K., LI K., HEAFIELD K., STONE K., EL-ARINI K., IYER K., MALIK K., CHIU K., BHALLA K., LAKHOTIA K., RANTALA-YEARY L., VAN DER MAATEN L., CHEN L., TAN L., JENKINS L., MARTIN L., MADAAN L., MALO L., BLECHER L., LANDZAAT L., DE OLIVEIRA L., MUZZI M., PASUPULETI M., SINGH M., PALURI M., KARDAS M., TSIMPOUKELLI M., OLDHAM M., RITA M., PAVLOVA M., KAMBADUR M., LEWIS M., SI M., SINGH M. K., HASSAN M., GOYAL N., TORABI N., BASHLYKOV N., BOGOYCHEV N., CHATTERJI N., ZHANG N., DUCHENNE O., ÇELEBI O., ALRASSY P., ZHANG P., LI P., VASIC P., WENG P., BHARGAVA P., DUBAL P., KRISHNAN P., KOURA P. S., XU P., HE Q., DONG Q., SRINIVASAN R., GANAPATHY R., CALDERER R., CABRAL R. S., STOJNIC R., RAILEANU R., MAHESWARI R., GIRDHAR R., PATEL R., SAUVESTRE R., POLIDORO R., SUMBALY R., TAYLOR R., SILVA R., HOU R., WANG R., HOSSEINI S., CHENNABASAPPA S., SINGH S., BELL S., KIM S. S., EDUNOV S., NIE S., NARANG S., RAPARTHY S., SHEN S., WAN S., BHOSALE S., ZHANG S., VANDENHENDE S., BATRA S., WHITMAN S., SOOTLA S., COLLOT S., GURURANGAN S., BORODINSKY S., HERMAN T., FOWLER T., SHEASHA T., GEORGIOU T., SCIALOM T., SPECKBACHER T., MIHAYLOV T., XIAO T., KARN U., GOSWAMI V., GUPTA V., RAMANATHAN V., KERKEZ V., GONGUET V., DO V., VOGETI V., ALBIERO V., PETROVIC V., CHU W., XIONG W., FU W., MEERS W., MARTINET X., WANG X., WANG X., TAN X. E., XIA X., XIE X., JIA X., WANG X., GOLDSCHLAG Y., GAUR Y., BABAEI Y., WEN Y., SONG Y., ZHANG Y., LI Y., MAO Y., COUDERT Z. D., YAN Z., CHEN Z., PAPAKIPOS Z., SINGH A., SRIVASTAVA A., JAIN A., KELSEY A., SHAJNFELD A., GANGIDI A., VICTORIA A., GOLDSTAND A., MENON A., SHARMA A., BOESENBERG A., BAEVSKI A., FEINSTEIN A., KALLET A., SANGANI A., TEO

A., YUNUS A., LUPU A., ALVARADO A., CAPLES A., GU A., HO A., POULTON A., RYAN A., RAMCHANDANI A., DONG A., FRANCO A., GOYAL A., SARAF A., CHOWDHURY A., GABRIEL A., BHARAMBE A., EISENMAN A., YAZDAN A., JAMES B., MAURER B., LEONHARDI B., HUANG B., LOYD B., PAOLA B. D., PARANJAPE B., LIU B., WU B., NI B., HANCOCK B., WASTI B., SPENCE B., STOJKOVIC B., GAMIDO B., MONTALVO B., PARKER C., BURTON C., MEJIA C., LIU C., WANG C., KIM C., ZHOU C., HU C., CHU C.-H., CAI C., TINDAL C., FEICHTENHOFER C., GAO C., CIVIN D., BEATY D., KREYMER D., LI D., ADKINS D., XU D., TESTUGGINE D., DAVID D., PARIKH D., LISKOVICH D., FOSS D., WANG D., LE D., HOLLAND D., DOWLING E., JAMIL E., MONTGOMERY E., PRESANI E., HAHN E., WOOD E., LE E.-T., BRINKMAN E., ARCAUTE E., DUNBAR E., SMOTHERS E., SUN F., KREUK F., TIAN F., KOKKINOS F., OZGENEL F., CAGGIONI F., KANAYET F., SEIDE F., FLOREZ G. M., SCHWARZ G., BADEER G., SWEE G., HALPERN G., HERMAN G., SIZOV G., GUANGYI, ZHANG, LAKSHMINARAYANAN G., INAN H., SHOJANAZERI H., ZOU H., WANG H., ZHA H., HABEEB H., RUDOLPH H., SUK H., ASPEGREN H., GOLDMAN H., ZHAN H., DAMLAJ I., MOLYBOG I., TUFANOV I., LEONTIADIS I., VELICHE I.-E., GAT I., WEISSMAN J., GEBOSKI J., KOHLI J., LAM J., ASHER J., GAYA J.-B., MARCUS J., TANG J., CHAN J., ZHEN J., REIZENSTEIN J., TEBOUL J., ZHONG J., JIN J., YANG J., CUMMINGS J., CARVILL J., SHEPARD J., MCPHIE J., TORRES J., GINSBURG J., WANG J., WU K., U K. H., SAXENA K., KHANDELWAL K., ZAND K., MATOSICH K., VEERARAGHAVAN K., MICHELENA K., LI K., JAGADEESH K., HUANG K., CHAWLA K., HUANG K., CHEN L., GARG L., A L., SILVA L., BELL L., ZHANG L., GUO L., YU L., MOSHKOVICH L., WEHRSTEDT L., KHABSA M., AVALANI M., BHATT M., MANKUS M., HASSON M., LENNIE M., RESO M., GROSHEV M., NAUMOV M., LATHI M., KENEALLY M., LIU M., SELTZER M. L., VALKO M., RESTREPO M., PATEL M., VYATSKOV M., SAMVELYAN M., CLARK M., MACEY M., WANG M., HERMOSO M. J., METANAT M., RASTEGARI M., BANSAL M., SANTHANAM N., PARKS N., WHITE N., BAWA N., SINGHAL N., EGEBO N., USUNIER N., MEHTA N., LAPTEV N. P., DONG N., CHENG N., CHERNOGUZ O., HART O., SALPEKAR O., KALINLI O., KENT P., PAREKH P., SAAB P., BALAJI P., RITTNER P., BONTRAGER P., ROUX P., DOLLAR P., ZVYAGINA P., RATANCHANDANI P., YUVRAJ P., LIANG Q., ALAO R., RODRIGUEZ R., AYUB R., MURTHY R., NAYANI R., MITRA R., PARTHASARATHY R., LI R., HOGAN R., BATTEY R., WANG R., HOWES R., RINOTT R., MEHTA S., SIBY S., BONDU S. J., DATTA S., CHUGH S., HUNT S., DHILLON S., SIDOROV S., PAN S., MAHAJAN S., VERMA S., YAMAMOTO S., RAMASWAMY S., LINDSAY S., LINDSAY S., FENG S., LIN S., ZHA S. C., PATIL S., SHANKAR S., ZHANG S., ZHANG S., WANG S., AGARWAL S., SAJUYIGBE S., CHINTALA S., MAX S., CHEN S., KEHOE S., SATTERFIELD S., GOVINDAPRASAD S., GUPTA S., DENG S., CHO S., VIRK S., SUBRAMANIAN S., CHOUDHURY S., GOLDMAN S., REMEZ T., GLASER T., BEST T., KOEHLER T., ROBINSON T., LI T., ZHANG T., MATTHEWS T., CHOU T., SHAKED T., VONTIMITTA V., AJAYI V., MONTANEZ V., MOHAN V., KUMAR V. S., MANGLA V., IONESCU V., POENARU V., MIHAILESCU V. T., IVANOV V., LI W., WANG W., JIANG W., BOUAZIZ W., CONSTABLE W., TANG X., WU X., WANG X., WU X., GAO X., KLEINMAN Y., CHEN Y., HU Y., JIA Y., QI Y., LI Y., ZHANG Y., ZHANG Y., ADI Y., NAM Y., YU, WANG, ZHAO Y., HAO Y., QIAN Y., LI Y., HE Y., RAIT Z., DEVITO Z., ROSNBRICK Z., WEN Z., YANG Z., ZHAO Z. & MA Z. (2024). The llama 3 herd of models.

HARARY F. & NORMAN R. Z. (1960). Some properties of line digraphs. *Rendiconti del Circolo Matematico di Palermo*, **9**(2), 161–168. DOI : 10.1007/BF02854581.

KNIGHT K., BADARAU B., BARANESCU L., BONIAL C., BARDOCZ M., GRIFFITT K., HERMJAKOB U., MARCU D., PALMER M., O'GORMAN T. & NATHAN S. (2020). Abstract meaning representation (amr) annotation release 3.0 ldc2020t02. *Linguistic Data Consortium*.

LIU Y., OTT M., GOYAL N., DU J., JOSHI M., CHEN D., LEVY O., LEWIS M., ZETTLEMOYER L. & STOYANOV V. (2019). RoBERTa : A Robustly Optimized BERT Pretraining Approach. arXiv :1907.11692 [cs].

LUO Z. (2021). Have Attention Heads in BERT Learned Constituency Grammar ? In I.-T. SORODOC, M. SUSHIL, E. TAKMAZ & E. AGIRRE, Éds., *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics : Student Research Workshop*, p. 8–15, Online : Association for Computational Linguistics. DOI : 10.18653/v1/2021.eacl-srw.2.

PALMER M., GILDEA D. & KINGSBURY P. (2005). The Proposition Bank : An Annotated Corpus of Semantic Roles. *Computational Linguistics*, **31**(1), 71–106. DOI : 10.1162/0891201053630264.

QORIB M., MOON G. & NG H. T. (2024). Are Decoder-Only Language Models Better than Encoder-Only Language Models in Understanding Word Meaning ? In L.-W. KU, A. MARTINS & V. SRIKUMAR, Éds., *Findings of the Association for Computational Linguistics : ACL 2024*, p. 16339–16347, Bangkok, Thailand : Association for Computational Linguistics. DOI : 10.18653/v1/2024.findings-acl.967.

RADFORD A., WU J., CHILD R., LUAN D., AMODEI D. & SUTSKEVER I. (2019). Language models are unsupervised multitask learners.

TENNEY I., DAS D. & PAVLICK E. (2019a). BERT Rediscovers the Classical NLP Pipeline. In A. KORHONEN, D. TRAUM & L. MÀRQUEZ, Éds., *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, p. 4593–4601, Florence, Italy : Association for Computational Linguistics. DOI : 10.18653/v1/P19-1452.

TENNEY I., XIA P., CHEN B., WANG A., POLIAK A., MCCOY R. T., KIM N., DURME B. V., BOWMAN S. R., DAS D. & PAVLICK E. (2019b). WHAT DO YOU LEARN FROM CONTEXT ? PROBING FOR SENTENCE STRUCTURE IN CONTEXTUALIZED WORD REPRESENTATIONS.

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

XU D., LI J., ZHU M., ZHANG M. & ZHOU G. (2020). Improving AMR Parsing with Sequence-to-Sequence Pre-training. arXiv :2010.01771 [cs].

ZHOU J., NASEEM T., FERNANDEZ ASTUDILLO R., LEE Y.-S., FLORIAN R. & ROUKOS S. (2021). Structure-aware fine-tuning of sequence-to-sequence transformers for transition-based AMR parsing. In M.-F. MOENS, X. HUANG, L. SPECIA & S. W.-T. YIH, Éds., *Proceedings of the 2021 conference on empirical methods in natural language processing*, p. 6279–6290, Online and Punta Cana, Dominican Republic : Association for Computational Linguistics. DOI : 10.18653/v1/2021.emnlp-main.507.

A Description des classificateurs biaffines

A.1 Classificateur symétrique bi-affine

Chaque arête est décrite par deux vecteurs $\boldsymbol{v}_1$ et $\boldsymbol{v}_2$, représentant l'attention d'un symbole sur l'autre, et inversement. Nous cherchons à déterminer la probabilité qu'un arc porte l'étiquette $k \in \{1, \cdots, c\}$

à partir de $\boldsymbol{v}_1$ et $\boldsymbol{v}_2$, à l'aide d'une fonction de trois variables :

$$\mathbb{P}(y = k \mid \boldsymbol{v}_1, \boldsymbol{v}_2) = f(k, \boldsymbol{v}_1, \boldsymbol{v}_2).$$

Comme nous ne savons pas lequel des deux vecteurs $\boldsymbol{v}_1$ et $\boldsymbol{v}_2$ représente l'attention du symbole source vers le symbole cible, (selon la direction de l'arc dans l'AMR projeté), il est raisonnable de poser $f(k, \boldsymbol{v}_1, \boldsymbol{v}_2) = f(k, \boldsymbol{v}_2, \boldsymbol{v}_1)$. On modélise la fonction f comme suit :

Soit $M_1, \cdots, M_c$ des matrices réelles symétriques carrées de $\mathbb{R}^{d\times d}$, et soient $\boldsymbol{b}_1, \cdots, \boldsymbol{b}_c$ des vecteurs de biais de $\mathbb{R}^d$. Nous calculons les c scalaires $\ell_i(\boldsymbol{v}_1, \boldsymbol{v}_2) = (\boldsymbol{v}_1 + \boldsymbol{b}_i)^\top \cdot M_i \cdot (\boldsymbol{v}_1 + \boldsymbol{b}_i)$. Puis, la fonction f est définie ainsi :

$$f(k, \boldsymbol{v}_1, \boldsymbol{v}_2) = \frac{\exp(\ell_k(\boldsymbol{v}_1, \boldsymbol{v}_2))}{\sum_{i=1}^{c} \exp(\ell_i(\boldsymbol{v}_1, \boldsymbol{v}_2))},$$

ce qui établit un classificateur bi-affine, symétrique quant à la permutation de $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$. Ce modèle requiert $\frac{d(d+3)}{2}$ paramètres par classe.

A.2 Classificateur symétrique de rang faible

Pour réduire la complexité du classificateur, on peut contraindre les paramètres matriciels à rester parmi les matrices de rang faible. En posant $r \in \mathbb{N},\ r \ll d$, on peut chercher les $(M_i)_i$ parmi les matrices carrées symétriques de $\mathbb{R}^{d\times d}$ dont le rang est inférieur ou égal à r. On montre facilement que pour de telles matrices, on a :

$$\exists U_i \in \mathbb{R}^{d\times r}, \exists (a_i^1, \cdots, a_i^r) \in \mathbb{R}^r, \quad M_i = U \cdot \mathrm{diag}(a_i^1, \cdots, a_i^r) \cdot U^\top,$$

où diag indique une matrice diagonale dont les coefficients sont $a_i^1, \cdots, a_i^r$. Il faut noter que U_i n'est pas nécessairement une matrice orthogonale, sa transposée n'est donc pas nécessairement sa pseudo-inverse, et les coefficients $a_i^1, \cdots, a_i^r$ ne sont pas nécessairement des valeurs propres de M_i. Nous laissons néanmoins ces coefficients libres afin de ne pas restreindre M_i à l'espace des matrices semi-définies positives. Ce modèle nécessite $(d+1)r + d$ paramètres par classe.

A.3 Classificateur bi-affine antisymétrique

Nous décrivons dans cette section un classificateur capable de résoudre le problème de classification binaire qui consiste à déterminer le sens de l'arc AMR associé à une arête du graphe d'attention. Nous désignons les deux classes par 0 et 1. Nous définissons f comme la probabilité conditionnelle :

$$f(\boldsymbol{v}_1, \boldsymbol{v}_2) = \mathbb{P}(y = 1 \mid \boldsymbol{v}_1, \boldsymbol{v}_2).$$

Comme le problème est antisymétrique (ce qui signifie que la classe devient la classe opposée si on échange $\boldsymbol{v}_1$ et $\boldsymbol{v}_2$), nous postulons que :

$$\forall \boldsymbol{v}_1 \in \mathbb{R}^d, \forall \boldsymbol{v}_2 \in \mathbb{R}^d, \quad f(\boldsymbol{v}_1, \boldsymbol{v}_2) = 1 - f(\boldsymbol{v}_2, \boldsymbol{v}_1).$$

Pour modéliser f, nous définissons une matrice antisymétrique $M \in \mathbb{R}^{d\times d}$ ($M^\top = -M$), un vecteur $\boldsymbol{b} \in \mathbb{R}^d$ et une fonction $\ell(\boldsymbol{v}_1, \boldsymbol{v}_2) = (\boldsymbol{v}_1 + \boldsymbol{b})^\top \cdot M \cdot (\boldsymbol{v}_2 + \boldsymbol{b}) - \boldsymbol{b}^\top \cdot M \cdot \boldsymbol{b}$. Nous posons ensuite

$$f(\boldsymbol{v}_1, \boldsymbol{v}_2) = \frac{1}{1 + \exp[-\ell(\boldsymbol{v}_1, \boldsymbol{v}_2)]}.$$

Ce modèle requiert $\frac{d(d-1)}{2}$ paramètres.

A.4 Classificateur antisymétrique de rang faible

Comme plus haut, la matrice M peut être approchée par une matrice de rang faible, puisqu'on remarque que si rang$(M) \leqslant r$, on a :

$$\exists U \in \mathbb{R}^{d \times r}, \exists A \in \mathrm{antisym}_r(\mathbb{R}), \quad M = U \cdot A \cdot U^\top.$$

Ce modèle requiert $dr + \frac{r(r-1)}{2}$ paramètres.

B Rôles sémantiques conservés

Relation	Freq	Relation	Freq
Theme	27.39%	Agent	23.15%
Mod	17.37%	Time	5.77%
Patient	5.35%	Location	4.96%
Topic	3.59%	Poss	2.96%
Experiencer	2.86%	Beneficiary	1.72%
Manner	1.51%	Purpose	1.09%
Degree	1.03%	Condition	0.96%
Cause	0.29%		

TABLE 2 – Liste des rôles sémantiques conservés pour nos expériences