

Un algorithme d'analyse sémantique fondée sur les graphes via le problème de l'arborescence généralisée couvrante

Alban Petit Caio Corro

Université Paris-Saclay, CNRS, LISN, 91400, Orsay, France
{alban.petit, caio.corro}@lisn.upsaclay.fr

RÉSUMÉ

Nous proposons un nouvel algorithme pour l'analyse sémantique fondée sur les graphes via le problème de l'arborescence généralisée couvrante.

ABSTRACT

A graph-based semantic parsing algorithm via the generalized spanning arborescence problem

We propose a novel algorithm for graph-based semantic parsing via the generalized spanning arborescence problem.

MOTS-CLÉS : Analyse sémantique, analyse fondée sur les graphes, optimisation combinatoire.

KEYWORDS: Semantic parsing, graph-based parsing, combinatorial optimization.

1 Introduction

Dans cet article, nous nous intéressons au problème de l'analyse sémantique avec des contraintes de bonne formation, c'est-à-dire respectant les contraintes de valence et de typage des arguments. Pour ces problèmes, les analyseurs modernes utilisent des architectures neuronales de type séquence-à-séquence avec attention (Jia & Liang, 2016; Dong & Lapata, 2016; Wang *et al.*, 2020, entre autres), de façon similaire à ce qui est fait en traduction automatique (Bahdanau *et al.*, 2015). Récemment, Herzig & Berant (2021) ont proposé d'utiliser une représentation du contenu sémantique formellement similaire aux constituants syntaxiques, permettant d'utiliser les modèles et algorithmes développés pour ces derniers. En d'autres termes, le même contenu sémantique est représenté différemment ce qui ouvre la voie à l'utilisation d'autres méthodes de prédiction, un principe bien connu en analyse syntaxique (Rambow, 2010). Dans la même lignée, nous proposons de prédire l'arbre de syntaxe abstraite (ASA) associé à un programme sémantique (voir Figure 1b), ce qui ouvre la voie aux méthodes de prédiction structurée fondées sur les graphes. Celles-ci comportent deux étapes (McDonald *et al.*, 2005) :

1. la construction d'un graphe pondéré en utilisant un modèle statistique ;
2. le calcul du sous-graphe (défini par un choix d'arcs) contraint de poids maximum.

Dans le cas de l'analyse en dépendances syntaxiques, de nombreux algorithmes ont été proposés pour la seconde étape. Ceux-ci diffèrent par leur espace de recherche (c'est-à-dire la classe de sous-graphes considérés) et par leur complexité computationnelle.

Nous nous focalisons sur des représentations sémantiques dont les ASA n'ont pas de réentrance. Nous posons les hypothèses suivantes :

1. chaque prédicat ou argument est ancré sur exactement un mot de la phrase ;
2. chaque mot ne peut être associé qu'à un prédicat ou un argument, au plus.

Nous qualifions de sémantiquement « vides » les mots qui ne sont l'ancre d'aucun prédicat ou argument. Le problème du calcul du sous-graphe contraint se réduit alors à un problème joint d'étiquetage et d'analyse en dépendances bi-lexicales : une étiquette (correspondant à un prédicat, un argument ou un vide sémantique) doit être choisie pour chaque mot et les dépendances bi-lexicales identifient les arguments des prédicats. En terme de graphe, cela correspond à une arborescence couvrante généralisée (Myung *et al.*, 1995), représentation déjà utilisée en traitement automatique des langues pour l'analyse en constituants discontinus (Corro *et al.*, 2017).

Nos contributions peuvent être résumées comme suit :

1. nous proposons une nouvelle approche pour l'analyse sémantique fondée sur les graphes ;
2. nous proposons un nouvel algorithme pour calculer l'arborescence couvrante généralisée de poids maximum capable de prendre en compte les contraintes de la grammaire sémantique ;
3. nous proposons une approche d'apprentissage par approximations variationnelles afin de pouvoir utiliser des données où l'ancrage des prédicats et arguments est absent des annotations ;
4. nous évaluons notre approche sur deux jeux de données.

2 Analyse sémantique et arborescence généralisée couvrante

(Motivations) Herzig & Berant (2021) proposent de réduire le problème de l'analyse sémantique à un problème similaire à celui de l'analyse en constituants. Cette approche nécessite d'utiliser un analyseur capable de produire des structures discontinues. En particulier, Herzig & Berant (2021) utilisent un des algorithme de Corro (2020). L'espace de recherche est ainsi limité, il n'est pas possible de produire une structure où un constituant discontinu a également son parent discontinu, voir Figure 1a. Bien qu'il soit possible de considérer des espaces de recherche plus grand au prix d'une complexité en temps plus élevé, l'analyse de structures de degré de bloc limité (dont la limite viendrait ici de la grammaire sémantique) sans contrainte de bonne imbrication est un problème NP-complet (Satta, 1992). Corro *et al.* (2017) ont proposé de réduire le problème de l'analyse en constituant discontinu au problème de l'arborescence généralisée couvrante. Bien que ce problème soit NP-difficile, cette formulation sous forme de problème de graphe permet l'utilisation de méthodes d'optimisation combinatoire. Nous suivons une approche similaire, c'est-à-dire que nous proposons un algorithme fondé sur les graphes qui n'a pas les problèmes d'espace de recherche de l'approche de Herzig & Berant (2021).

(Graphes) Soit $G = (V, A)$ un graphe dirigé composé des noeuds V et des arcs $A \subseteq V \times V$. Pour tout sous-ensemble d'arcs $U \subseteq A$, nous dénotons $\sigma_G^+(U)$ (respectivement $\sigma_G^-(U)$) l'ensemble des arcs ayant comme source un noeud de U et comme destination un noeud de $V \setminus U$ (respectivement ayant comme source un noeud de $V \setminus U$ et comme destination un noeud de U) dans le graphe G . Soit $B \subseteq A$ un sous-ensemble d'arcs. Nous dénotons $V[B]$ la couverture de B , c'est-à-dire l'ensemble des noeuds qui sont des extrémités d'au moins un arc de B . Un graphe $G = \langle V, A \rangle$ est une arborescence de racine $u \in V$ si et seulement si il contient $|V| - 1$ arcs et qu'il existe un chemin dirigé depuis u vers chaque noeud de $G \setminus \{u\}$. Soient $\pi = \{V_0, \dots, V_n\}$ une partition de V en $n + 1$ groupes et $B \subseteq A$ un sous-ensemble d'arcs. Le graphe $G' = \langle V[B], B \rangle$ est une arborescence généralisée couvrante sur

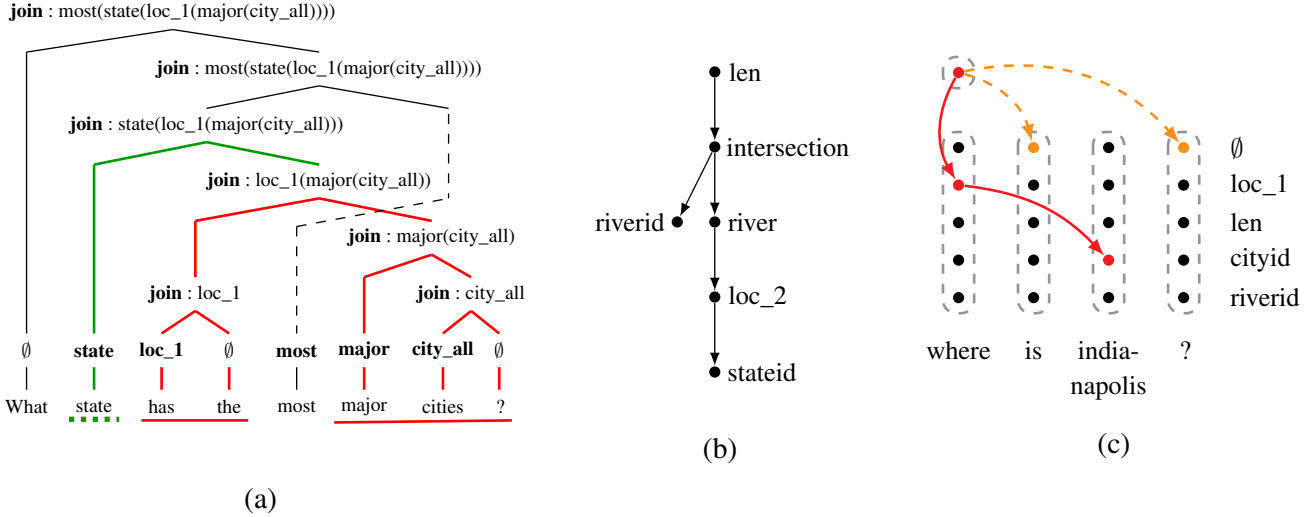


FIGURE 1 – (a) Exemple de la réduction de l’analyse sémantique à une analyse en constituants. L’analyseur de Herzig & Berant (2021) ne peut pas produire cette structure car elle est en dehors de l’espace de recherche : le constituant formé par les mots soulignés par un trait plein rouge a un parent également discontinu via l’ajout du mot souligné avec un trait pointillé vert (Corro, 2020, Section 3.6). (b) Exemple d’un arbre de syntaxe abstraite. (c) Exemple d’analyse sémantique d’une phrase sous forme d’une arborescence généralisée couvrante.

la partition π de G si et seulement si G' est une arborescence et que $V[B]$ contient exactement un nœud par groupe de π .

(Grammaire sémantique et arbre de syntaxe abstraite) Nous définissons une grammaire sémantique comme un tuple $\mathcal{G} = \langle E, T, f_{\text{TYPE}}, f_{\text{ARGS}} \rangle$ où :

- E est l’ensemble des prédicats et arguments, que nous nommerons ensemble des étiquettes ;
- T est un ensemble de types ;
- $f_{\text{TYPE}} : E \rightarrow T$ est un fonction de typage associant un type à chaque étiquette ;
- $f_{\text{ARGS}} : E \times T \rightarrow \mathbb{N}$ est une fonction de valence associant à chaque étiquette le nombre d’arguments attendus d’un certain type.

Une étiquette $e \in E$ est une entité si et seulement si $\forall t \in T : f_{\text{ARGS}}(e, t) = 0$. Soit $G = \langle V, A, l \rangle$ un graphe étiqueté, c’est-à-dire un graphe où la fonction $l : V \rightarrow E$ associe une étiquette à chaque nœud. Les arcs associent alors les arguments du programme, voir Figure 1b. Le graphe G est un arbre de syntaxe abstraite appartenant au langage généré par la grammaire sémantique \mathcal{G} si et seulement si G est une arborescence et que la valence et le typage des étiquettes sont respectés :

$$\forall u \in V, t \in T : f_{\text{ARGS}}(l(u), t) = \left| \left\{ u \rightarrow v \in \sigma_G^+(\{u\}) \mid f_{\text{TYPE}}(l(v)) = t \right\} \right|$$

(Analyse sémantique fondée sur les graphes) Nous proposons de réduire le problème de l’analyse sémantique au calcul d’une arborescence couvrante généralisée dans un graphe. Soit $w = w_1, \dots, w_n$ une phrase de n mots. Nous construisons un graphe complet $G = \langle V, A, \pi, l \rangle$ de la façon suivante :

- La partition $\pi = \{V_0, \dots, V_n\}$ contient $n + 1$ groupes où V_0 est un groupe racine et les autres groupes sont associés aux mots de la phrase ;
- Le groupe racine contient un unique nœud $V_0 = \{0\}$;
- Nous supposons l’existence d’une étiquette $\emptyset \in E$ afin de prendre en compte l’existence de mots sémantiquement vides.

- Chaque groupe $V_i, i \neq 0$, contient $|E|$ nœuds et aucun nœud dans un groupe n'a la même étiquette, c-à-d $\forall u, v \in V_i : l(u) = l(v) \implies u = v$;

Soit $B \subseteq A$ un sous-ensemble d'arcs. Le graphe $G' = \langle V[B], B, l \rangle$ définit une analyse sémantique pour la phrase w si et seulement si :

- G' est une arborescence généralisée couvrante de G ayant pour racine le nœud 0 ;
- il y a exactement un seul arc sortant de 0 ayant pour destination un nœud d'étiquette non-nulle, c-à-d que $|\{ 0 \rightarrow u \in \sigma_{G'}^+(\{0\}) \mid l(u) \neq \emptyset \}| = 1$;
- la sous-arborescence ayant pour racine le descendant direct de 0 avec une étiquette non-nulle est un arbre de syntaxe abstraite valide dans le langage \mathcal{G} .

Un exemple est donné sur la Figure 1c.

En pratique, l'objectif est de calculer l'analyse sémantique la plus probable d'après une fonction de coût paramétrisée par un réseau de neurones. Malheureusement, le problème de l'arborescence couvrante généralisée de poids maximum est un problème NP-difficile (Myung *et al.*, 1995). Nous faisons l'hypothèse que les contraintes du langage sémantique que nous ajoutons n'en changent pas la complexité.

3 Programme mathématique et méthode de résolution

Nous proposons une modélisation du problème d'analyse syntaxique décrit dans la section précédente sous forme d'un programme linéaire en nombre entiers. Notre formulation est fondée sur des flots multi-commodités mais adaptés à notre problème particulier (Martins *et al.*, 2009; Corro *et al.*, 2017). L'intuition est la suivante : la racine 0 du graphe envoie une unité de flot à chaque groupe de nœuds dans π . S'il existe un chemin entre la racine et un nœud du groupe, alors le flot est consommé par celui-ci. De plus, en restreignant le nombre d'arcs sélectionnés à être égal à $|\pi| - 1$, seules les structures de type arborescence couvrante satisfont les contraintes.

Nous définissons trois vecteurs de variables binaires : $\mathbf{x} \in \{0, 1\}^V$ indiquant la présence ou l'absence d'un nœud dans l'arborescence généralisée couvrante ; $\mathbf{y} \in \{0, 1\}^A$ indiquant la présence ou l'absence d'un arc dans l'arborescence généralisée couvrante ; $\mathbf{z} \in \{0, 1\}^{A \times \pi}$ indiquant la quantité de flot à destination d'un groupe passant sur un arc. Les vecteurs $\boldsymbol{\mu} \in \mathbb{R}^V$ et $\boldsymbol{\phi} \in \mathbb{R}^A$ contiennent les poids associés aux nœuds et aux arcs, respectivement. Le problème de l'analyse sémantique via l'arborescence généralisée couvrante peut être formulé de la façon suivante :

$$\max_{\mathbf{x}, \mathbf{y}, \mathbf{z}} \quad \boldsymbol{\mu}^\top \mathbf{x} + \boldsymbol{\phi}^\top \mathbf{y}$$

$$\text{tel que} \quad \begin{aligned} (1) \quad & \forall c \in \{1, \dots, n\} : \sum_{a \in \sigma_G^+(\{0\})} z_{a,c} = 1 & (2) \quad & \sum_{a \in A} y_a = n \\ (3) \quad & \forall u \in V \setminus \{0\} : \sum_{a \in \sigma^-(\{u\})} y_a = x_u & (4) \quad & \forall a \in A, c \in \{1, \dots, n\} : z_{a,c} \leq y_a \\ (5) \quad & \forall c, c' \in \{1, \dots, n\} : \sum_{a \in \sigma^-(V_c)} z_{a,c'} - \sum_{a \in \sigma^+(V_c)} z_{a,c'} = \mathbb{1}[c = c'] \\ (6) \quad & \forall u \in V \setminus \{0\}, t \in T : \sum_{u \rightarrow v \in \{\sigma^+(u) \mid f_{\text{TYPE}}(l(v))=t\}} x_u = f_{\text{ARGS}}(l(u), t) \end{aligned}$$

La racine envoie une unité de flot par groupe dans $\pi \setminus \{V_0\}$ (contraintes 1) et seuls les arcs sélectionnés peuvent transporter des unités de flot (contraintes 4). Chaque groupe consomme l'unité de flot lui étant destiné et propage les autres (contraintes 5). En ajoutant la contrainte (2) qui impose qu'exactly $n = |\pi| - 1$ arcs soient sélectionnés, \mathbf{y} est contraint à être une arborescence sur les groupes « contractés », c'est-à-dire que cela ne garantit pas que tous les arcs sélectionnés incidents d'un groupe soient incidents au même nœud. Un nœud est sélectionné si et seulement s'il admet un arc entrant

(contrainte 3). Maximiser l'objectif sous les contraintes (1-5) est un problème polynomial : on peut utiliser l'algorithme de Chu-Liu-Edmonds sur un graphe dans lequel les groupes ont été contractés, comme observé par [Corro et al. \(2017\)](#). Finalement, les contraintes (6) garantissent qu'un nœud ne peut avoir des arcs sortants que s'il est sélectionné et que le type et la valence de la grammaire sémantique sont respectés dans le sous-graphe prédit. On observe que ces dernières contraintes rendent le problème difficile : il n'existe plus d'algorithme polynomial permettant de résoudre le programme linéaire en nombre entiers lorsque celles-ci sont introduites.

Nous proposons d'introduire les contraintes (6) comme des pénalités quadratiques dans l'objectif ([Nesterov, 2005](#)) et d'utiliser la méthode du gradient conditionnel (a.k.a Frank-Wolfe, [Frank & Wolfe, 1956](#); [Levitin & Polyak, 1966](#)) pour résoudre le problème résultant, approche introduite par [Yurtsever et al. \(2018\)](#). Cette méthode consiste à résoudre le problème de manière itérative par descente de gradient sur des approximations du premier ordre de l'objectif. Ces approximations sont produites par un oracle linéaire qui, dans notre cas, est simplement l'algorithme de Chu-Liu-Edmonds sur un graphe contracté. Un avantage de cette méthode d'optimisation est que si seules des égalités sont relâchées, la taille de pas optimale peut être calculée sous une forme analytique — si des inégalités sont également relâchées (voir Section 4), une taille de pas quasi-optimale peut être trouvée par bisection car l'espace de recherche est borné. Notons que notre algorithme retourne la relaxation continue du programme linéaire en nombres entiers. Si la solution optimale n'est pas binaire, nous utilisons CPLEX pour résoudre le programme ci-dessus en conservant uniquement les variables ayant une valeur non-nulle dans la solution de la relaxation continue. Ce dernier programme a donc un très petit nombre de variables et est rapide à résoudre.

4 Apprentissage sans les ancrages

Les jeux de données annotés en représentation sémantique ne contiennent pas nécessairement les ancrages des prédicats et arguments sur les mots de la phrase. C'est le cas des données GeoQuery ([Zelle & Mooney, 1996](#)) que nous utilisons dans nos expériences. Nous proposons donc un objectif probabiliste dans lequel nous marginalisons sur toutes les solutions induisant la représentation sémantique annotée dans les données d'entraînement.

Soit X et Y deux variables aléatoires dont les réalisations \mathbf{x} et \mathbf{y} sont définies de façon similaire aux variables du programme mathématique. La probabilité d'un couple $\langle \mathbf{x}, \mathbf{y} \rangle$ est définie comme :

$$p(\mathbf{x}, \mathbf{y}) = \exp(s(\mathbf{x}, \mathbf{y}) - c) \quad \text{où} \quad s(\mathbf{x}, \mathbf{y}) = \begin{cases} \boldsymbol{\mu}^\top \mathbf{x} + \boldsymbol{\phi}^\top \mathbf{y} & \text{si } \mathbf{x} \text{ et } \mathbf{y} \text{ sont une solution faisable} \\ -\infty & \text{sinon} \end{cases}$$

où nous définissons $\exp(-\infty) = 0$ et $c = \log \sum_{\langle \mathbf{x}, \mathbf{y} \rangle} \exp(s(\mathbf{x}, \mathbf{y}))$ est la fonction de log-partition. Soit \mathcal{C} l'ensemble des couples $\langle \mathbf{x}, \mathbf{y} \rangle$ induisant l'analyse sémantique annotée dans les données. La fonction de perte pour apprendre les paramètres de notre modèle est :

$$-\log \sum_{\langle \mathbf{x}, \mathbf{y} \rangle \in \mathcal{C}} p(\mathbf{x}, \mathbf{y}) = -\log \sum_{\langle \mathbf{x}, \mathbf{y} \rangle \in \mathcal{C}} \exp(s(\mathbf{x}, \mathbf{y}) - c) = - \left(\log \sum_{\langle \mathbf{x}, \mathbf{y} \rangle \in \mathcal{C}} \exp(s(\mathbf{x}, \mathbf{y})) \right) + c$$

Ces deux termes étant intractables, nous dérivons une borne supérieure tractable.

(Log-partition) Soit une matrice U contenant sur chaque ligne une des k solutions faisables et Δ^k le simplexe de dimension $k - 1$. Nous construisons une approximation de la log-partition en passant par la forme variationnelle de c (Wainwright & Jordan, 2008) :

$$c = \underbrace{\max_{\mathbf{p} \in \Delta^k} \mathbf{p}^\top \left(U \begin{bmatrix} \boldsymbol{\mu} \\ \boldsymbol{\phi} \end{bmatrix} \right) + H[\mathbf{p}]}_{\text{Forme variationnelle}} = \underbrace{\max_{\mathbf{m} \in \mathcal{M}} \mathbf{m}^\top \begin{bmatrix} \boldsymbol{\mu} \\ \boldsymbol{\phi} \end{bmatrix} + H_{\mathcal{M}}[\mathbf{m}]}_{\text{Reformulation via le polytope des marginaux}} \leq \underbrace{\max_{\mathbf{m} \in \mathcal{L}} \mathbf{m}^\top \begin{bmatrix} \boldsymbol{\mu} \\ \boldsymbol{\phi} \end{bmatrix} + H[\mathbf{m}]}_{\text{Borne supérieure}}$$

où \mathcal{M} est le polytope des marginaux, c'est-à-dire l'enveloppe convexe des solutions entières faisables. Le terme $H[\mathbf{p}] = -\sum_i p_i \log p_i$ est l'entropie de Shannon et $H_{\mathcal{M}}[\mathbf{m}]$ est définie tel que les deux premières formes variationnelles sont équivalentes (cette fonction d'entropie jointe n'a généralement pas de forme analytique). Nous dérivons une borne supérieure en maximisant sur un espace de recherche plus grand, c-à-d $\mathcal{L} \supseteq \mathcal{M}$, et en décomposant le terme d'entropie jointe (Cover, 1999, propriété 4 du Summary en page 41). En particulier, on observe que chaque solution faisable a exactement un arc entrant et un nœud sélectionné par groupe. Nous imposons uniquement cette contrainte, l'approximation de c étant alors un ensemble de log-partitions de distributions de Boltzmann indépendantes, deux par par groupe (une pour les arcs et une pour les nœuds).

(Alignement) Pour dériver une borne supérieure de la fonction de perte, nous devons obtenir une borne inférieure du premier terme. Soit $q(X, Y)$ une distribution de proposition qui assigne une probabilité nulle à toutes les réalisations $\langle \mathbf{x}, \mathbf{y} \rangle \notin \mathcal{C}$, on dérive la borne suivante via l'inégalité de Jensen :

$$\log \sum_{\langle \mathbf{x}, \mathbf{y} \rangle \in \mathcal{C}} \exp(\boldsymbol{\mu}^\top \mathbf{x} + \boldsymbol{\phi}^\top \mathbf{y}) = \log \mathbb{E}_{q(\mathbf{x}, \mathbf{y})} \left[\frac{\exp(\boldsymbol{\mu}^\top \mathbf{x} + \boldsymbol{\phi}^\top \mathbf{y})}{q(\mathbf{x}, \mathbf{y})} \right] \geq \mathbb{E}_{q(\mathbf{x}, \mathbf{y})} [\boldsymbol{\mu}^\top \mathbf{x} + \boldsymbol{\phi}^\top \mathbf{y}] + H[q(X, Y)]$$

Plutôt que de calculer la distribution q qui maximise cette borne inférieure, nous cherchons la distribution qui assigne une probabilité de 1 au couple $\langle \mathbf{x}, \mathbf{y} \rangle$ de poids maximum. Ceci est équivalent à trouver un alignement entre les nœuds de l'ASA sémantique et les mots de la phrase¹. Notons que si nous n'avons aucune contrainte d'alignement, c'est-à-dire qu'un mot peut être l'ancrage de plusieurs nœuds de l'ASA, alors on peut calculer l'alignement de poids maximum par programmation dynamique, où l'algorithme remonte les nœuds de l'ASA dans l'ordre topologique inverse, d'une façon similaire à l'algorithme d'étiquetage d'arbre de dérivation de Corro & Le Roux (2017). Si nous ajoutons la contrainte que chaque mot ne peut être aligné qu'avec au plus un nœud, alors le problème devient intractable. Nous remplaçons donc cette contrainte par une pénalité quadratique dans l'objectif et utilisons l'algorithme du gradient conditionnel où notre programme dynamique est l'oracle linéaire.

5 Expériences

(Données) Nous évaluons notre modèle sur SCAN (Lake & Baroni, 2018) et la variante FunQL de GEOQUERY (Zelle & Mooney, 1996; Kate et al., 2005). Nous utilisons les données pré-traitées par Herzig & Berant (2021). Pour tester la robustesse de notre approche, nous utilisons trois découpages

1. Ce problème d'alignement diffère des problèmes usuels car le poids de l'alignement dépend des vecteurs $\boldsymbol{\mu}$ et $\boldsymbol{\phi}$. Si un prédicat est aligné au mot w_i et un de ses arguments est aligné au mot w_j de la phrase, alors le poids de l'alignement inclut le poids de l'arc $i \rightarrow j$.

	SCAN			GEOQUERY		
	IID	RIGHT	ARIGHT	IID	TPL	LENGTH
SEQ2SEQ	99.9	11.6	0	78.5	46.0	24.3
+ ELMO	100	54.9	41.6	79.3	50.0	25.7
BERT2SEQ	100	77.7	95.3	81.1	49.6	26.1
GRAMMAR	100	0.0	4.2	72.1	54.0	24.6
BART	100	50.5	100	87.1	67.0	19.3
Constituants	100	100	100	86.1	82.2	63.6
Ce travail	100	99.5	99.6	88.6	76.5	51.6
+ projectivité	100	100	100	66.8	47.8	45.5

FIGURE 2 – (c) % de prédictions exactes sur SCAN et GEOQUERY. Tous les modèles sauf le nôtre sont des implémentations de [Herzig & Berant \(2021\)](#).

différents par jeu de données. Les découpages IID sont des découpages aléatoires. Pour SCAN, les découpages RIGHT et ARIGHT permettent de tester la robustesse de généralisation à la compositionnalité du modèle ([Loula et al., 2018](#)). Pour GEOQUERY, nous utilisons les découpages LENGTH et TPL. Le premier permet de tester la généralisation à des phrases plus longues et le second ne contient pas d’ASA identique dans les données de test et d’entraînement ([Finegan-Dollak et al., 2018](#)).

(Architecture) Sur SCAN, nous utilisons des plongements lexicaux pour représenter les mots de la phrase. Sur GEOQUERY, un modèle BERT est utilisé pour calculer ces représentations ([Devlin et al., 2019](#)). Ces représentations sont ensuite fournies à un LSTM bi-directionnel ([Hochreiter & Schmidhuber, 1997](#)). Pour obtenir la pondération des étiquettes pour les différents mots, nous appliquons une projection linéaire standard aux sorties du LSTM. La pondération des arcs est obtenue en fournissant ces sorties à une couche bi-affine ([Dozat & Manning, 2017](#)).

(Résultats) Les résultats sont reproduits sur la Figure 2. Nous observons que notre modèle obtient des résultats à l’état de l’art pour les données IID. Sur SCAN, nous avons besoin de forcer la projectivité pour ne pas faire d’erreur sur RIGHT et ARIGHT. Nous pouvons facilement ajouter ces contraintes en utilisant l’algorithme de [Eisner \(1996\)](#) comme oracle linéaire. Sur GEOQUERY, notre modèle produit de moins bons résultats que l’approche de [Herzig & Berant \(2021\)](#) mais est cependant bien meilleur que tous les modèles précédents de la littérature. On note que forcer la projectivité sur GEOQUERY implique une dégradation importante des résultats.

6 Conclusion

Dans cet article, nous proposons une nouvelle approche pour l’analyse sémantique fondée sur les graphes par réduction au problème de l’arborescence généralisée couvrante. Cette approche est différente de ce qui a été exploré dans la littérature et permet, entre autre, d’explorer un espace de recherche plus grand que l’approche de [Herzig & Berant \(2021\)](#). Des travaux futurs pourront s’intéresser à l’utilisation de notre analyseur pour d’autres jeux de données et à améliorer les résultats sur les découpages non-aléatoires.

Références

- BAHDANAU D., CHO K. & BENGIO Y. (2015). Neural machine translation by jointly learning to align and translate. In Y. BENGIO & Y. LECUN, Édts., *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.
- CORRO C. (2020). Span-based discontinuous constituency parsing : a family of exact chart-based algorithms with time complexities from $O(n^6)$ down to $O(n^3)$. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, p. 2753–2764, Online : Association for Computational Linguistics. DOI : [10.18653/v1/2020.emnlp-main.219](https://doi.org/10.18653/v1/2020.emnlp-main.219).
- CORRO C. & LE ROUX J. (2017). Transforming dependency structures to LTAG derivation trees. In *Proceedings of the 13th International Workshop on Tree Adjoining Grammars and Related Formalisms*, p. 112–121, Umeå, Sweden : Association for Computational Linguistics.
- CORRO C., LE ROUX J. & LACROIX M. (2017). Efficient discontinuous phrase-structure parsing via the generalized maximum spanning arborescence. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, p. 1644–1654, Copenhagen, Denmark : Association for Computational Linguistics. DOI : [10.18653/v1/D17-1172](https://doi.org/10.18653/v1/D17-1172).
- COVER T. M. (1999). *Elements of information theory*. John Wiley & Sons.
- DEVLIN J., CHANG M.-W., LEE K. & TOUTANOVA K. (2019). BERT : Pre-training of deep bidirectional transformers for language understanding. In *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](https://doi.org/10.18653/v1/N19-1423).
- DONG L. & LAPATA M. (2016). Language to logical form with neural attention. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, p. 33–43, Berlin, Germany : Association for Computational Linguistics. DOI : [10.18653/v1/P16-1004](https://doi.org/10.18653/v1/P16-1004).
- DOZAT T. & MANNING C. D. (2017). Deep biaffine attention for neural dependency parsing. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings* : OpenReview.net.
- EISNER J. M. (1996). Three new probabilistic models for dependency parsing : An exploration. In *COLING 1996 Volume 1 : The 16th International Conference on Computational Linguistics*.
- FINEGAN-DOLLAK C., KUMMERFELD J. K., ZHANG L., RAMANATHAN K., SADASIVAM S., ZHANG R. & RADEV D. (2018). Improving text-to-SQL evaluation methodology. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, p. 351–360, Melbourne, Australia : Association for Computational Linguistics. DOI : [10.18653/v1/P18-1033](https://doi.org/10.18653/v1/P18-1033).
- FRANK M. & WOLFE P. (1956). An algorithm for quadratic programming. *Naval Research Logistics Quarterly*, **3**(1-2), 95–110. DOI : <https://doi.org/10.1002/nav.3800030109>.
- HERZIG J. & BERANT J. (2021). Span-based semantic parsing for compositional generalization. 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. 908–921, Online : Association for Computational Linguistics. DOI : [10.18653/v1/2021.acl-long.74](https://doi.org/10.18653/v1/2021.acl-long.74).
- HOCHREITER S. & SCHMIDHUBER J. (1997). Long short-term memory. *Neural Comput.*, **9**(8), 1735–1780. DOI : [10.1162/neco.1997.9.8.1735](https://doi.org/10.1162/neco.1997.9.8.1735).

- JIA R. & LIANG P. (2016). Data recombination for neural semantic parsing. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, p. 12–22, Berlin, Germany : Association for Computational Linguistics. DOI : [10.18653/v1/P16-1002](https://doi.org/10.18653/v1/P16-1002).
- KATE R. J., WONG Y. W. & MOONEY R. J. (2005). Learning to transform natural to formal languages. In *Proceedings of the 20th National Conference on Artificial Intelligence - Volume 3*, AAAI'05, p. 1062–1068 : AAAI Press.
- LAKE B. & BARONI M. (2018). Generalization without systematicity : On the compositional skills of sequence-to-sequence recurrent networks. In J. DY & A. KRAUSE, Édts., *Proceedings of the 35th International Conference on Machine Learning*, volume 80 de *Proceedings of Machine Learning Research*, p. 2873–2882 : PMLR.
- LEVITIN E. & POLYAK B. (1966). Constrained minimization methods. *USSR Computational Mathematics and Mathematical Physics*, **6**(5), 1–50. DOI : [https://doi.org/10.1016/0041-5553\(66\)90114-5](https://doi.org/10.1016/0041-5553(66)90114-5).
- LOULA J., BARONI M. & LAKE B. (2018). Rearranging the familiar : Testing compositional generalization in recurrent networks. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP : Analyzing and Interpreting Neural Networks for NLP*, p. 108–114, Brussels, Belgium : Association for Computational Linguistics. DOI : [10.18653/v1/W18-5413](https://doi.org/10.18653/v1/W18-5413).
- MARTINS A., SMITH N. & XING E. (2009). Concise integer linear programming formulations for dependency parsing. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, p. 342–350, Suntec, Singapore : Association for Computational Linguistics.
- MCDONALD R., PEREIRA F., RIBAROV K. & HAJIĆ J. (2005). Non-projective dependency parsing using spanning tree algorithms. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, p. 523–530, Vancouver, British Columbia, Canada : Association for Computational Linguistics.
- MYUNG Y.-S., LEE C.-H. & TCHA D.-W. (1995). On the generalized minimum spanning tree problem. *Networks*, **26**(4), 231–241. DOI : <https://doi.org/10.1002/net.3230260407>.
- NESTEROV Y. (2005). Smooth minimization of non-smooth functions. *Mathematical programming*, **103**(1), 127–152.
- RAMBOW O. (2010). The simple truth about dependency and phrase structure representations : An opinion piece. In *Human Language Technologies : The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, p. 337–340, Los Angeles, California : Association for Computational Linguistics.
- SATTA G. (1992). Recognition of linear context-free rewriting systems. In *30th Annual Meeting of the Association for Computational Linguistics*, p. 89–95, Newark, Delaware, USA : Association for Computational Linguistics. DOI : [10.3115/981967.981979](https://doi.org/10.3115/981967.981979).
- WAINWRIGHT M. J. & JORDAN M. I. (2008). *Graphical models, exponential families, and variational inference*. Now Publishers Inc.
- WANG B., SHIN R., LIU X., POLOZOV O. & RICHARDSON M. (2020). RAT-SQL : Relation-aware schema encoding and linking for text-to-SQL parsers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, p. 7567–7578, Online : Association for Computational Linguistics. DOI : [10.18653/v1/2020.acl-main.677](https://doi.org/10.18653/v1/2020.acl-main.677).
- YURTSEVER A., FERCOQ O., LOCATELLO F. & CEVHER V. (2018). A conditional gradient framework for composite convex minimization with applications to semidefinite programming. In J. DY & A. KRAUSE, Édts., *Proceedings of the 35th International Conference on Machine Learning*, volume 80 de *Proceedings of Machine Learning Research*, p. 5727–5736 : PMLR.

ZELLE J. M. & MOONEY R. J. (1996). Learning to parse database queries using inductive logic programming. In *AAAI/IAAI*, p. 1050–1055, Portland, OR : AAAI Press/MIT Press.