Rapport de stage

Apprentissage par renforcement d'un analyseur syntaxique en transitions avec retour-arrière

Petit Maxime

Introduction

Est-il possible d'apprendre à un analyseur à annuler des décisions qu'il a prises une fois qu'il est en possession d'informations suffisantes pour corriger ses erreurs ?

Introduction

Est-il possible d'apprendre à un analyseur à annuler des décisions qu'il a prises une fois qu'il est en possession d'informations suffisantes pour corriger ses erreurs ?

Les petits ruisseaux

Introduction

Est-il possible d'apprendre à un analyseur à annuler des décisions qu'il a prises une fois qu'il est en possession d'informations suffisantes pour corriger ses erreurs ?

Les	petits	ruisseaux
Det		

Introduction

Est-il possible d'apprendre à un analyseur à annuler des décisions qu'il a prises une fois qu'il est en possession d'informations suffisantes pour corriger ses erreurs ?

Les	petits	ruisseaux
Det	Adj	

Introduction

Est-il possible d'apprendre à un analyseur à annuler des décisions qu'il a prises une fois qu'il est en possession d'informations suffisantes pour corriger ses erreurs ?

Les	petits	ruisseaux
Det	Adj	Nom

Introduction

Est-il possible d'apprendre à un analyseur à annuler des décisions qu'il a prises une fois qu'il est en possession d'informations suffisantes pour corriger ses erreurs ?

Les	petits	ruisseaux
Det	Adj	Nom

La

Introduction

Est-il possible d'apprendre à un analyseur à annuler des décisions qu'il a prises une fois qu'il est en possession d'informations suffisantes pour corriger ses erreurs ?

Les	petits	ruisseaux
Det	Adj	Nom

La
Det

Introduction

Est-il possible d'apprendre à un analyseur à annuler des décisions qu'il a prises une fois qu'il est en possession d'informations suffisantes pour corriger ses erreurs ?

Les	petits	ruisseaux
Det	Adj	Nom

La	petite
Det	

Introduction

Est-il possible d'apprendre à un analyseur à annuler des décisions qu'il a prises une fois qu'il est en possession d'informations suffisantes pour corriger ses erreurs ?

Les	petits	ruisseaux
Det	Adj	Nom

La	petite
Det	Adj

Introduction

Est-il possible d'apprendre à un analyseur à annuler des décisions qu'il a prises une fois qu'il est en possession d'informations suffisantes pour corriger ses erreurs ?

Les	petits	ruisseaux
Det	Adj	Nom

La	petite	mange
Det	Adj	

Introduction

Est-il possible d'apprendre à un analyseur à annuler des décisions qu'il a prises une fois qu'il est en possession d'informations suffisantes pour corriger ses erreurs ?

Les	petits	ruisseaux
Det	Adj	Nom

La	petite	mange
Det	Adj	Verbe

Introduction

Est-il possible d'apprendre à un analyseur à annuler des décisions qu'il a prises une fois qu'il est en possession d'informations suffisantes pour corriger ses erreurs ?

Les	petits	ruisseaux
Det	Adj	Nom

La	petite	mange
Det	~~Adj~~	Verbe
	Nom	

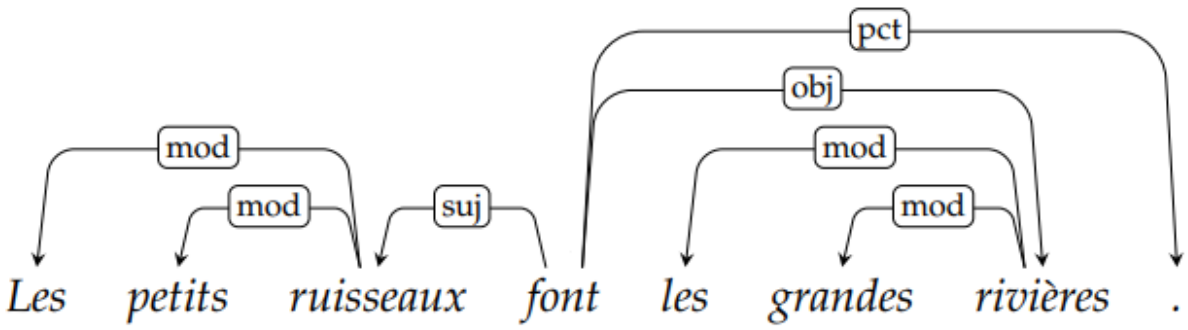
Sommaire

- Analyse en dépendance
- Analyse en transition
- Retour-arrière
- Apprentissage par renforcement
- Expériences
- Résultats

Analyse en dépendance

Les petits ruisseaux font les grandes rivières .

Analyse en dépendance

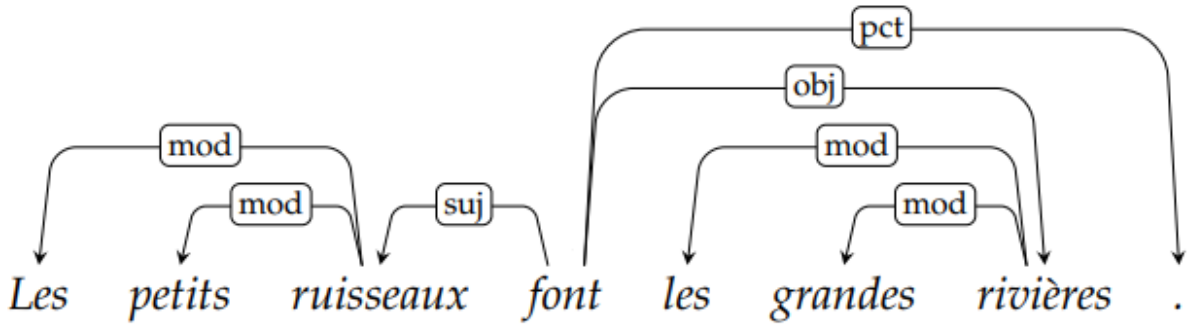


Analyse en transition

Deux objets fondamentaux :

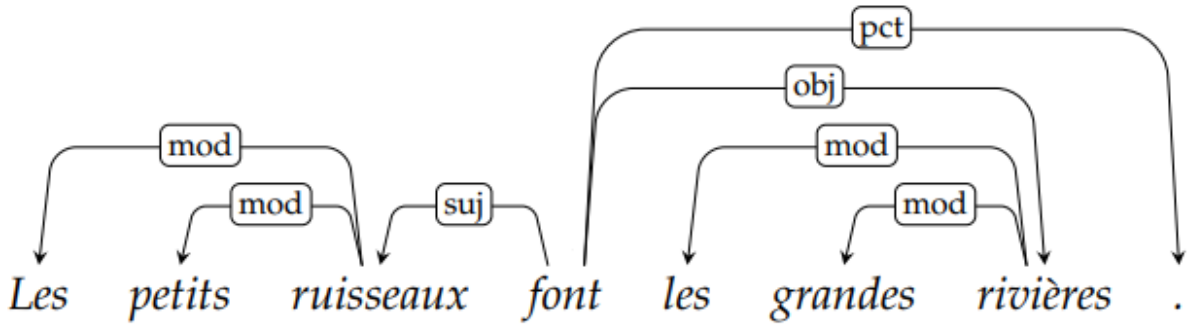
- Les configurations : représentation de l'analyseur à un instant de l'analyse.
- Les transitions : actions permettant de passer d'une configuration à une autre.

Exemple



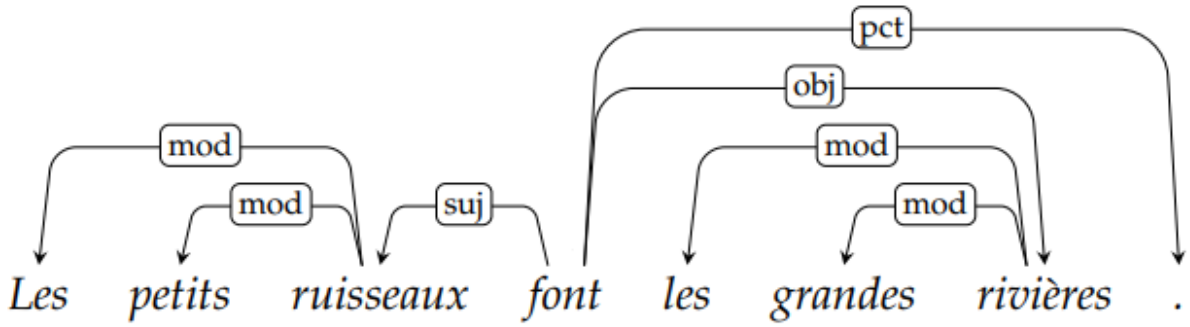
Pile σ	File β	Transition	Dépendance
	Les petits ruisseaux font les grandes rivières .		

Exemple



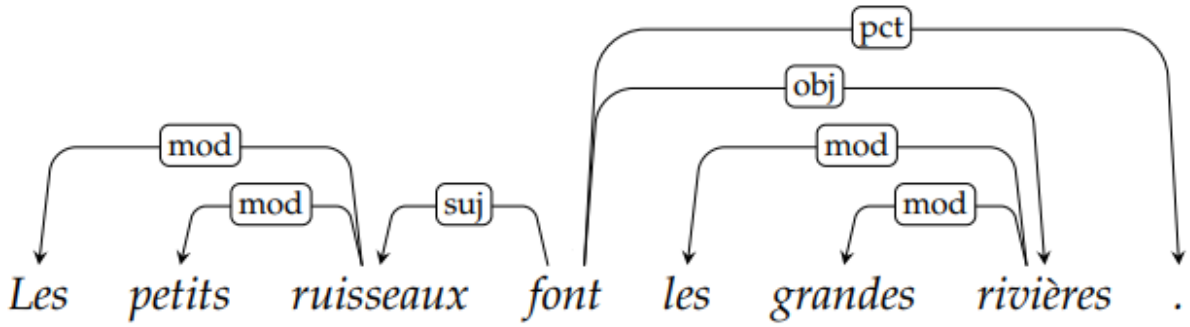
Pile σ	File β	Transition	Dépendance
	Les petits ruisseaux font les grandes rivières .	Shift	

Exemple



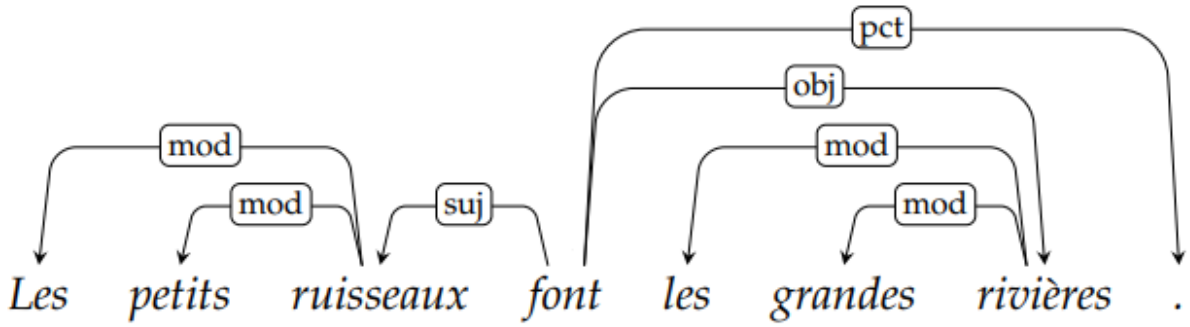
Pile σ	File β	Transition	Dépendance
	Les petits ruisseaux font les grandes rivières .	Shift	
Les	Les petits ruisseaux font les grandes rivières .		

Exemple



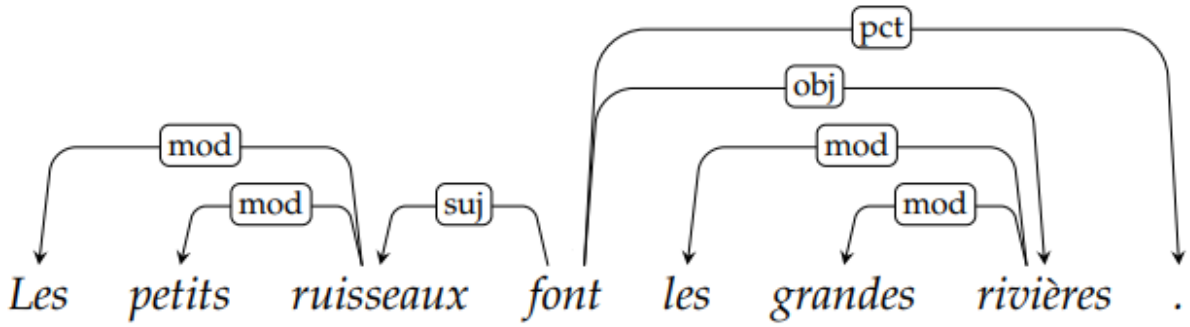
Pile σ	File β	Transition	Dépendance
	Les petits ruisseaux font les grandes rivières .	Shift	
Les	Les petits ruisseaux font les grandes rivières .	Shift	

Exemple



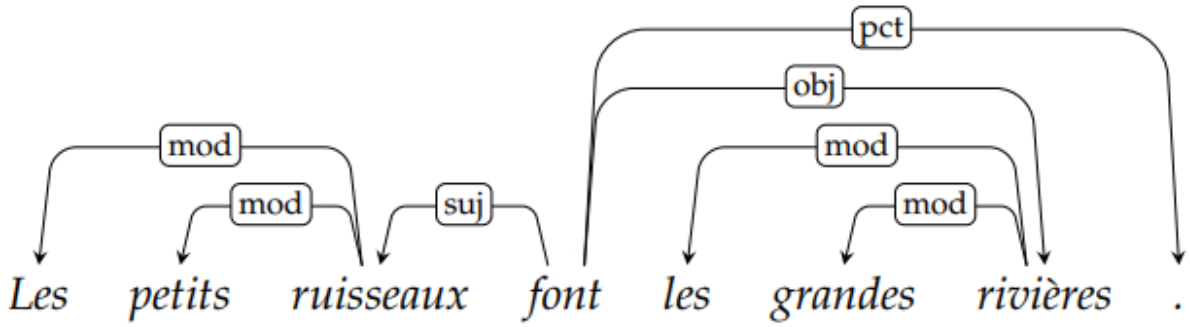
Pile σ	File β	Transition	Dépendance
	Les petits ruisseaux font les grandes rivières .	Shift	
Les	Les petits ruisseaux font les grandes rivières .	Shift	
Les petits	Les petits ruisseaux font les grandes rivières .		

Exemple



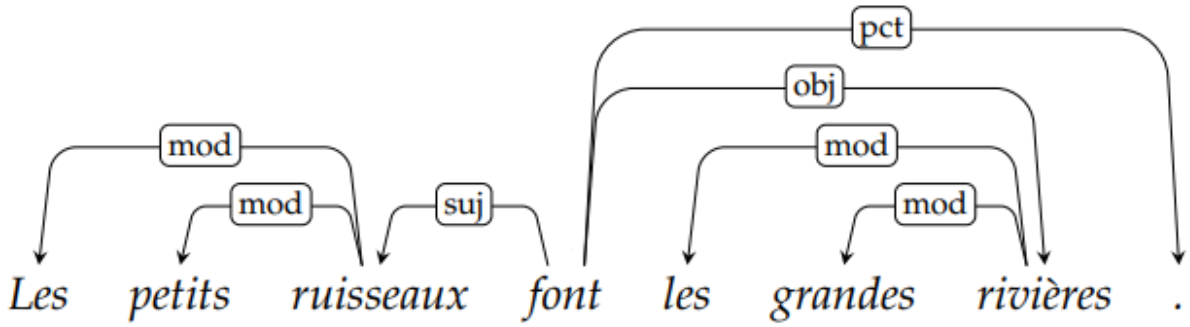
Pile σ	File β	Transition	Dépendance
	Les petits ruisseaux font les grandes rivières .	Shift	
Les	Les petits ruisseaux font les grandes rivières .	Shift	
Les petits	Les petits ruisseaux font les grandes rivières .	Arc-gauche$_{mod}$	(ruisseaux, mod, petits)

Exemple



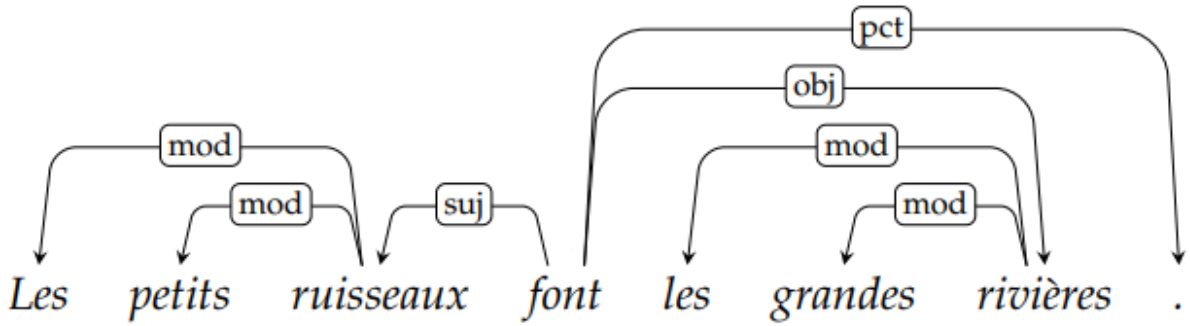
Pile σ	File β	Transition	Dépendance
	Les petits ruisseaux font les grandes rivières .	Shift	
Les	Les petits ruisseaux font les grandes rivières .	Shift	
Les petits	Les petits ruisseaux font les grandes rivières .	Arc-gauche$_{mod}$	(ruisseaux, mod, petits)
Les	Les petits ruisseaux font les grandes rivières .		

Exemple



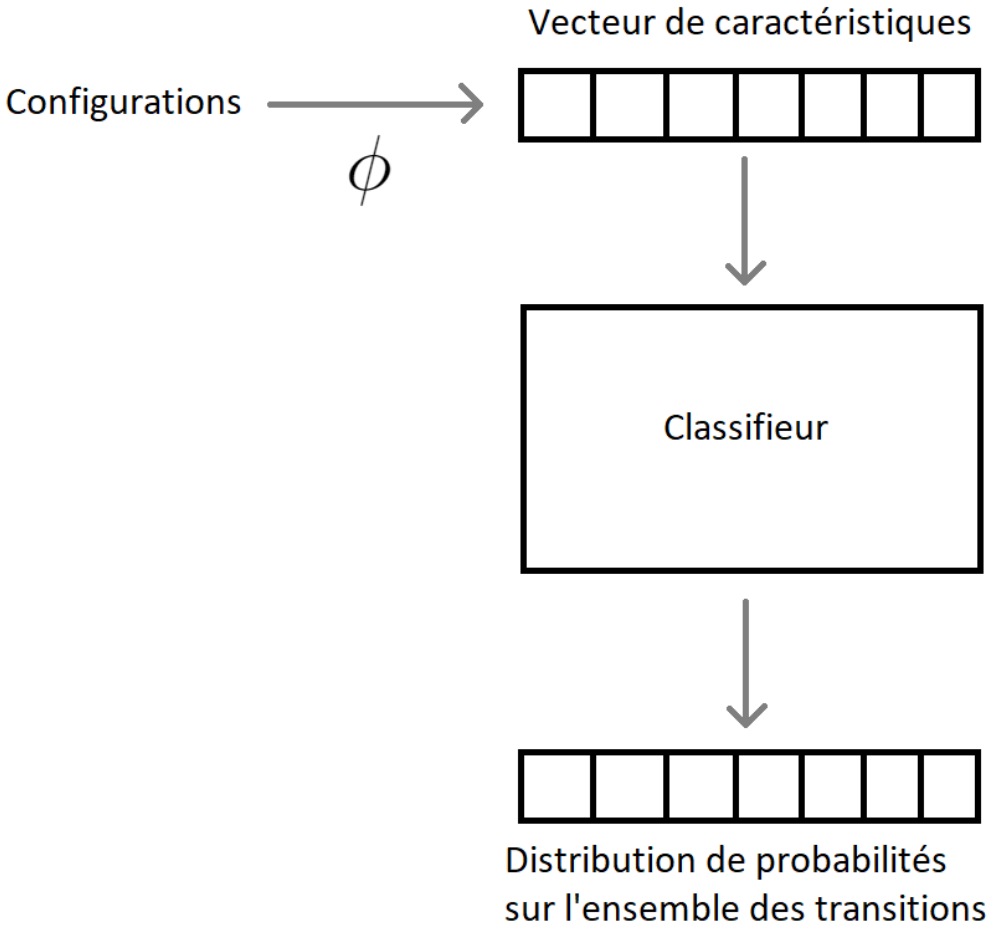
Pile σ	File β	Transition	Dépendance
	Les petits ruisseaux font les grandes rivières .	Shift	
Les	Les petits ruisseaux font les grandes rivières .	Shift	
Les petits	Les petits ruisseaux font les grandes rivières .	Arc-gauche$_{mod}$	(ruisseaux, mod, petits)
Les	Les petits ruisseaux font les grandes rivières .	Arc-gauche$_{mod}$	(ruisseaux, mod, Les)

Exemple



Pile σ	File β	Transition	Dépendance
	Les petits ruisseaux font les grandes rivières .	Shift	
Les	Les petits ruisseaux font les grandes rivières .	Shift	
Les petits	Les petits ruisseaux font les grandes rivières .	Arc-gauche$_{mod}$	(ruisseaux, mod, petits)
Les	Les petits ruisseaux font les grandes rivières .	Arc-gauche$_{mod}$	(ruisseaux, mod, Les)
	Les petits ruisseaux font les grandes rivières .	Shift	
ruisseaux	Les petits ruisseaux font les grandes rivières .	Arc-gauche$_{suj}$	(font, suj, ruisseaux)
	Les petits ruisseaux font les grandes rivières .	Shift	
font	Les petits ruisseaux font les grandes rivières .	Shift	
font les	Les petits ruisseaux font les grandes rivières .	Shift	
font les grandes	Les petits ruisseaux font les grandes rivières .	Arc-gauche$_{mod}$	(rivières, mod, grandes)
font les	Les petits ruisseaux font les grandes rivières .	Arc-gauche$_{mod}$	(rivières, mod, les)
font	Les petits ruisseaux font les grandes rivières .	Arc-droit$_{obj}$	(font, obj, rivières)
font rivières	Les petits ruisseaux font les grandes rivières .	Reduce	
font	.	Arc-droit$_{pct}$	(font, pct, .)
font .			

Classifieur



Entraînement supervisé

- Jeu d'entraînement créé par un oracle.
- Principe : Décomposer l'arbre de dépendance d'une phrase en une séquence de configurations et de transitions.
- Contient seulement des configurations correctes.
- En inférence, le classifieur peut analyser des configurations incorrectes suite à une erreur.

Suite

Défauts :

- Algorithme glouton : décision locale
- Propagation d'erreur

Cadre incrémental : Simulation de la lecture humaine.

- Pas de vision sur le contexte droit du mot analysé.

Retour-arrière

Retour-arrière : Nouvelle transition permettant au modèle d'annuler la dernière action qu'il a effectuée.

Configuration	Pile σ	File β	Transition	Dépendance
1		La	Shift	
2	La	La petite	Shift	
3	La petite	La petite mange	Retour-arrière	
4	La	La petite mange	Arc-gauche$_{det}$	(petite, det, La)

Difficile à apprendre en apprentissage supervisé.

Apprentissage par renforcement

Processus de décision Markovien : MDP (S, A, T, r) où

- S : Ensemble des états.
- A : Ensemble des actions.
- T : Fonction de transition $T : S \times A \to S$.
- r : Fonction de récompense $r : S \times A \to \mathbb{R}$

Politique

Politique $\pi : S \rightarrow A$

- Pas d'ensemble d'exemples (s_i, a_i) d'états s_i étiquetés par des actions optimales a_i.
- L'agent découvre des exemples (s_t, a_t, r_t, s_{t+1}) de récompense r_t et de nouvel état s_{t+1} lorsqu'il choisit une action a_t à partir d'un état s_t.
- L'agent doit faire un compromis entre :
  - exploitation : choisir une action qui parait optimale afin de maximiser ses récompenses à court terme.
  - exploration : choisir une action qui parait sous-optimale afin d'en apprendre davantage sur l'environnement et découvrir de nouveaux exemples.

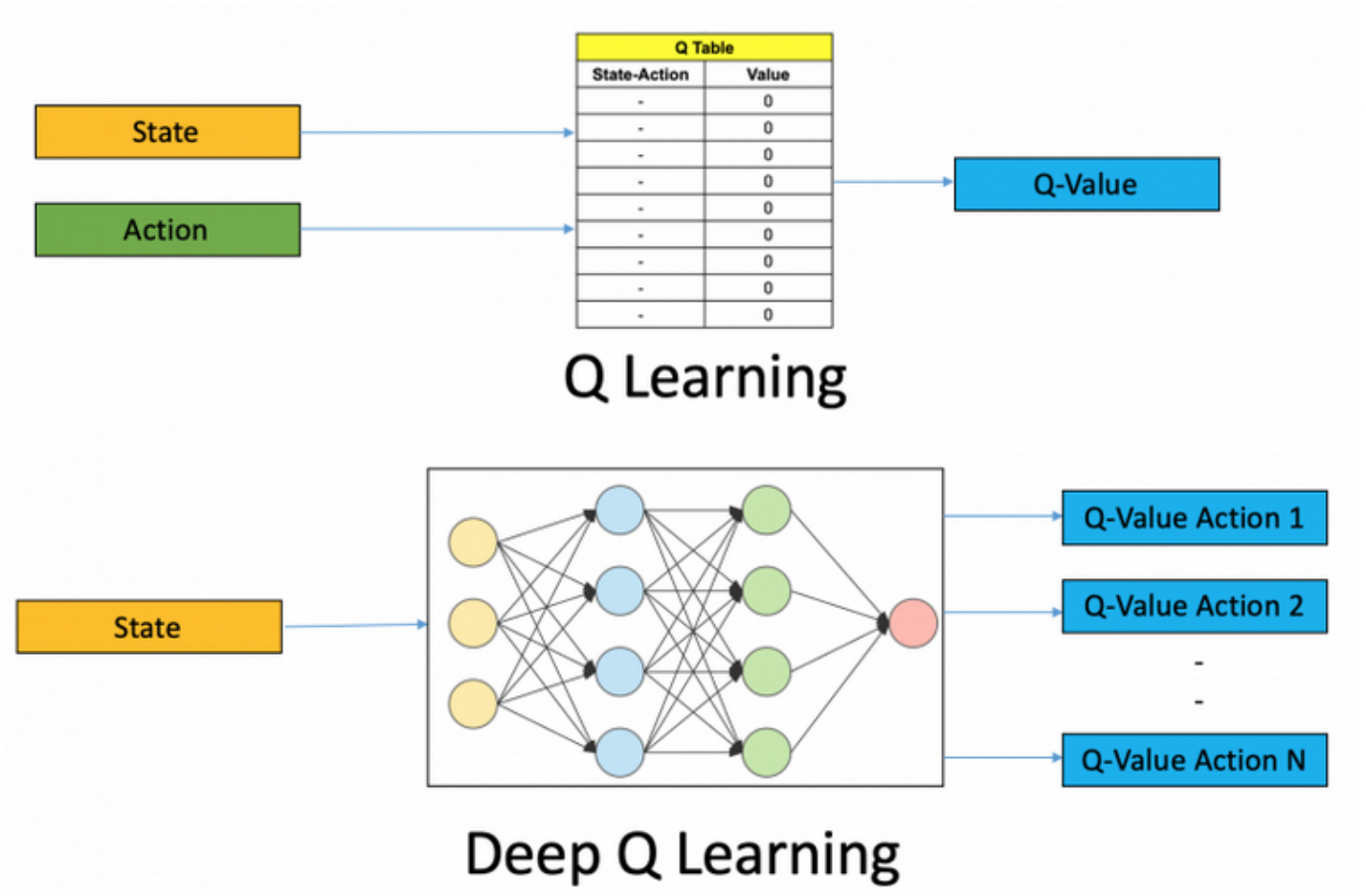
Récompense à long terme

Maximiser la récompense à long terme : $R_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k}$

Espérance du gain à long terme : $Q^{\pi}(i, a)$

$$\begin{aligned} Q^{\pi}(i, a) &= \mathbb{E}\Big[R_t | s^t = i, a^t = a\Big] \\ Q^{\pi}(i, a) &= r(i, a) + \gamma \max_b Q^{\pi}(j, b) \end{aligned}$$

Deep Q-learning



https://www.analyticsvidhya.com/blog/2019/04/introduction-deep-q-learning-python/

Expériences

- Jeu de données : Universal Dependencies : UD_French-GSD. Dont 80% d'entraînement, 10% de validation et 10% de test.
- Modèle : MLP à deux couches cachées. Une couche non-linéaire et une couche linéaire.
- Entrée du classifieur :
  - Information : FORM et UPOS.
  - Mots :
    - Les trois mots en sommet de pile.
    - Le mot en tête de file.
    - Les deux mots précédent la tête de file.
  - Historique : Les cinq dernières actions utilisées.

Politique

Politique : Epsilon-Greedy modifié :
Pour $\epsilon \in [0, 1]$ et $\delta \in [0, 1]$ tels que $\epsilon + \delta \leq 1$:

- Avec une probabilité ϵ, on applique une action aléatoire.
- Avec une probabilité δ, on applique l'action optimale dans la configuration actuelle.
- Avec une probabilité $1 - \epsilon - \delta$, on applique l'action prédite par le modèle.

Politique

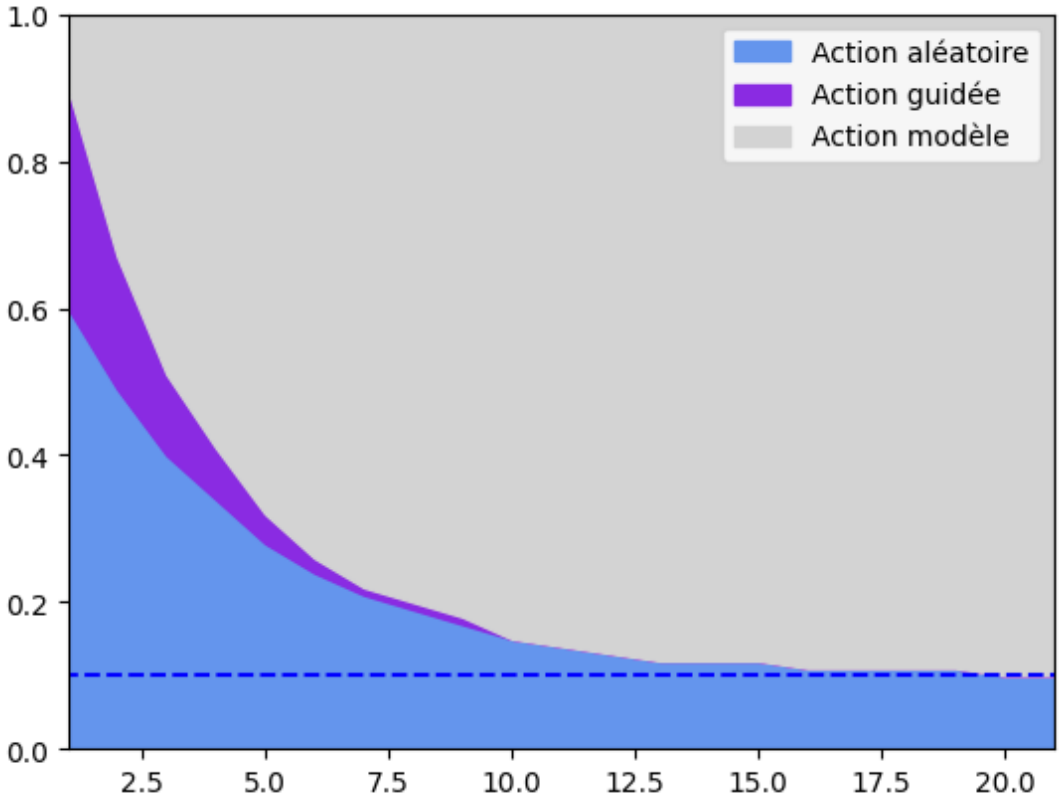


FIGURE – Évolution des probabilités au cours des 20 premières époques.

Fonction de récompense

La récompense d'une action de base est l'opposé du nombre de dépendances correctes que cette action va empêcher de créer dans l'arbre.

	récompense lorsque l'action annulée était optimale ($R_p = 0$)	récompense lorsque l'action annulée était erronée ($R_p < 0$)
Baseline	-1	0
Somme	0	$-R_p$
Somme Shiftée	-1	$-R_p - 1$
Log Somme	-1	$ln(1 - R_p)$

FIGURE – Tableau récapitulatif des récompenses du retour-arrière.

Résultats

Récompense	Score
LS	88.28 (89.01)
supervised	88.24
SS	87.94
B	87.40
NoRA	87.38

FIGURE – Résultats sur 100 époques classées par score.

	récompense lorsque l'action annulée était optimale ($R_p = 0$)	récompense lorsque l'action annulée était erronée ($R_p < 0$)
Baseline	-1	0
Somme Shiftée	-1	$-R_p - 1$
Log Somme	-1	$ln(1 - R_p)$

FIGURE – Tableau récapitulatif des récompenses du retour-arrière.

Résultats

	Action optimale ↓ Action optimale	Action optimale ↓ Action erronée	Action erronée ↓ Action optimale	Action erronée ↓ Action erronée	Total
LS	504	107	617	452	1680
SS	78	21	142	96	337
B	10	0	13	0	23

FIGURE – Répartition des retours-arrière selon l'action précédente et l'action suivante.

Conclusion

- Le retour-arrière a été appris et est utilisé.
- Il permet d'améliorer les performances et de corriger les erreurs.
- Il n'est pas parfait, il crée des erreurs.

Pour la suite

- Créer un modèle incrémental capable de prédire la partie de discours en plus de l'arbre syntaxique.
- Affiner les hyper-paramètres.
- Potentiellement tester d'autres méthodes d'apprentissage par renforcement.

Fin

Merci pour votre attention.