

Etudier les incohérences des analyseurs syntaxiques

L'idée de ce TP est d'étudier la cohérence d'un analyseur syntaxique.

Les parseurs en dépendances obtiennent des résultats tout à fait satisfaisants quand ils ont accès à suffisamment de ressources (données d'entraînement, modèle de langue...). Et on a envie de faire confiance à des modèles avec des bons scores sur les benchmarks. Mais ils restent des boîtes noires, donc peut-on vraiment leur faire confiance ? et comment s'en assurer ?

L'approche suivie ici consiste à parser des phrases ayant la même structure syntaxique et à comparer les prédictions des parseurs. Il va donc nous falloir des phrases partageant la même structure et on aimerait avoir des phrases de longueur variée.

Une première idée est de prendre des phrases existantes déjà annotées et de changer des mots dedans en s'assurant de maintenir la structure intacte. Attention, la structure syntaxique est capricieuse et facilement ambiguë.

ex : J'ai mangé un gâteau avec une cuillère/cerise.

Dans le schéma UD et la majorité des autres aussi, « avec une cuillère » se rattacherait à « mangé » alors que « avec une cerise » se rattacherait à « gâteau ».

On va commencer par créer des variantes en proposant du vocabulaire alternatif pour des phrases déjà annotées :

[https://docs.google.com/spreadsheets/d/
12ranvKkW7Cat8nQv1xdXpDppXy3UO0Q-nACpcVoOhdw/edit?usp=sharing](https://docs.google.com/spreadsheets/d/12ranvKkW7Cat8nQv1xdXpDppXy3UO0Q-nACpcVoOhdw/edit?usp=sharing)

Ces phrases viennent du corpus UD French GSD train.

Après avoir ajouter des mots en douzième colonnes, séparés par des ; on va tester un parseur pré-entraîné : spacy_udpipe fr. (si vous en avez un autre sous la main, allez-y)

Procédure : pour chaque phrase, pour chaque mot pour lequel on a des alternatives, créer une nouvelle phrase avec cette alternative, la parser et comparer le résultat au parse de la phrase d'origine.

Les principaux points que l'on va étudier : quand les parses sont différents, s'agit-il du label de la relation de dépendance qui change ou aussi l'index du gouverneur ? Les différences portent-elles sur le mot qui a changé (en tant que dépendant ou gouverneur) ou sur d'autres mots ? Quand il y a des différences, une des alternatives est-elle en accord avec l'annotation gold ?

Quand on a plus que 2 variantes, combien de structures différentes apparaît-il ?

Une fois arrivé·e·s ici, il est temps de prendre un peu de recul. Les parseurs ne sont pas entraînés pour être cohérents, d'ailleurs ils ne traitent qu'une phrase à la fois (modulo le batch, mais c'est une autre histoire).

Comment feriez-vous pour améliorer la cohérence des parseurs ?

Notes sur le parsing en dépendances :

Objectif de la tâche :

Prédire la structure syntaxique d'une phrase dans une langue donnée sous la forme d'un arbre dont les nœuds sont les mots de la phrase.

Même si Universal Dependencies (UD) est le schéma d'annotation majoritaire aujourd'hui, il y en a eu et en existe encore d'autres, notamment le SUD de Kahane et Gerdès.

Les données :

Beaucoup de corpus ont été créés, soit par conversion depuis des formalismes en constituants (PennTreeBank), soit annotés nativement en dépendances (PDT)...

Universal Dependencies fait beaucoup pour rassembler les efforts de la communauté, et a entre autre proposé un format et un schéma commun pour l'annotation de nouveaux corpus ou la conversion de corpus existants.

Le format Conll-U :

```
# sent_id = fr-ud-dev_00972
# text = Tout allait changer.
```

1	Tout	tout	PRON	_ Gender=Masc Number=Sing Person=3 PronType=Ind	2	nsubj	_ wordform=tout
2	allait	aller	VERB	_ Mood=Ind Number=Sing Person=3 Tense=Imp VerbForm=Fin	0	root	_ _
3	changer	changer	VERB	_ VerbForm=Inf	2	xcomp	_ SpaceAfter=No
4	.	.	PUNCT	_ _	2	punct	_ _

Format tabulaire à 10 colonnes :

ID, l'indice du token en comptant à partir de 1 (0 est réservé pour la racine) ; FORM, la forme du mot ;

LEMMA, le lemme du mot ;

UPOS, la partie du discours prise dans un ensemble fixé par UD ;

XPOS, une partie du discours prise dans un autre ensemble pour les corpus pré-datant UD ;

FEATS, les traits morphosyntaxiques de la forme ;

HEAD, l'indice du gouverneur du mot actuel, 0 sert à marquer la racine ;

DEPREL, la relation de dépendance entre le mot et son gouverneur ;

EXTREL, d'autres relations de dépendance permettant de décrire des structures syntaxiques plus compliquées ;

MISC, une colonne pour des informations supplémentaires.

Les méthodes :

Des systèmes de règles au tout début, mais difficile comme toujours. Deux grandes familles historiques de méthodes et une nouvelle venue :

Les parseurs à transitions / shift-reduce :

L'analyse syntaxique est la trace d'un automate à états. Les transitions consistent à déplacer des mots ou des fragments d'arbres entre des buffers et des piles, et les états correspondent au contenu des différents buffers et piles de l'automate.

Les features sont complexes et représentent l'histoire de la machine et son état. Mais étant données un automate, il est souvent difficile de trouver la meilleure séquence de transitions correspondant à un arbre donné. Beaucoup de travail autour de la notion d'oracle.

Les parseurs graphiques :

L'analyse syntaxique est un arbre couvrant maximal (meilleur score). Problème de théorie des graphes bien étudié et pour lequel il existe des algorithmes efficaces quand le score d'un arbre et la somme du score de chacune des arêtes qui le compose. Algorithmes pour les cas projectif et non projectif.

Les features sont plus simples car le parse n'a pas « d'histoire ».

Les modèles de langues et leurs capacités de représentation ont beaucoup bénéficié aux parseurs graphiques.

Les taggeurs :

Plus récemment des parseurs sont apparus qui traitent l'analyse syntaxique comme une tâche de tagging/labeling, où chaque token de la phrase (et parfois les espaces entre les tokens) reçoit un tag.

Ex : Hexatagging : <https://arxiv.org/abs/2306.05477>

Le tagging est plus rapide que le décodage d'un arbre couvrant mais les features sont encore plus simples. (trade-off temps de calcul / qualité de l'analyse)

Les challenges :

Les grands modèles de langues n'ont pas l'air d'avoir besoin de représenter explicitement la syntaxe/la grammaire pour être capables de faire beaucoup de choses impressionnantes. Rôle du parsing dans le TAL du futur ?

Encore utile néanmoins pour les linguistes.

Langues peu/sous-dotées ;

De manière plus générale, problème de généralisation ; les humains sont bons, les machines pas encore.

Black-boxes : on ne sait pas vraiment pourquoi les parseurs font ce qu'ils font.
Les modèles ne sont pas entraînés pour être cohérents, mais pour analyser des phrases correctement...