Application de la réécriture de graphes au traitement de corpus linguistiques

Guy Perrier
LORIA / INRIA Nancy Grand Est / Université de Lorraine

travail réalisé avec
Bruno Guillaume (LORIA)

Séminaire IDL Grenoble
6 novembre 2020

Plan de la présentation

- Les corpus annotés vus comme graphes
- L'exploration de corpus comme « matching » de graphes
- La réécriture de graphes
- La transformation d'annotations comme réécriture de graphes

L'importance de l'annotation linguistique de corpus

- En **linguistique**, la **linguistique de corpus** occupe une place de plus en plus importante.

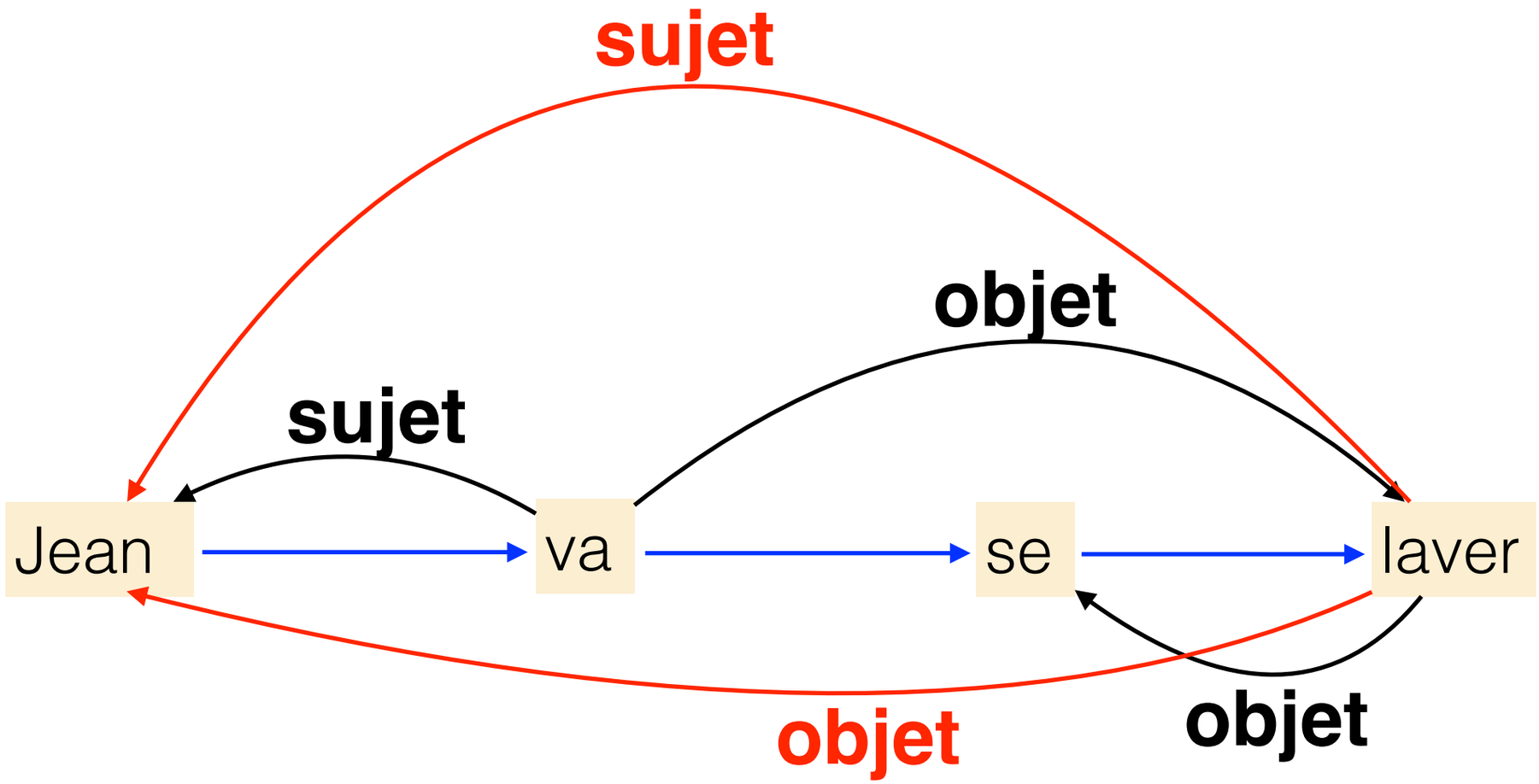
- En **TAL**, l'**apprentissage profond** occupe une place hégémonique et requiert des corpus annotés de qualité de plus en plus gros.

- Il est donc nécessaire de disposer d'**outils automatiques** performants pour annoter les corpus linguistiques.

Les corpus annotés linguistiquement comme collections de graphes

- Un corpus annoté linguistiquement est une collection de **graphes**.
- Ces graphes se réduisent parfois à des **arbres**, en syntaxe de surface par exemple.
- Ce sont des graphes au sens plein du terme en **syntaxe profonde** et en **sémantique**.
- Ce sont aussi des graphes au sens plein du terme quand on considère simultanément **plusieurs niveaux linguistiques** (phonologique et syntaxique par exemple).

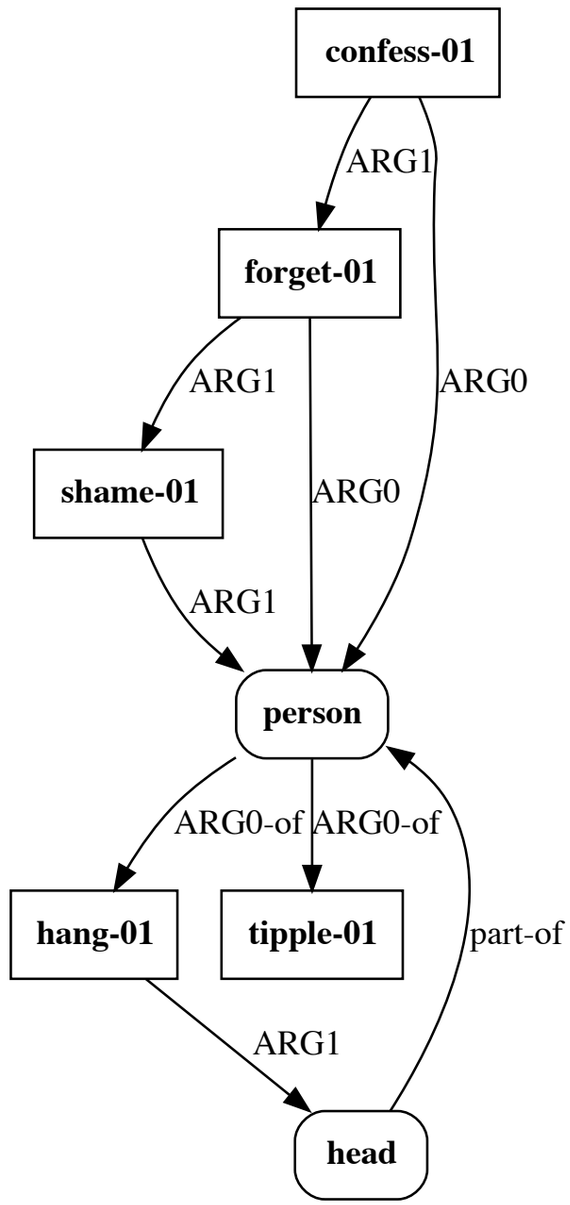
Exemple d'annotation vue comme graphe



Exemple d’annotation vue comme graphe

“Forget that I am ashamed,” the tippler confessed, hanging his head.

« Oubliez que j'ai honte », a avoué le buveur en baissant la tête

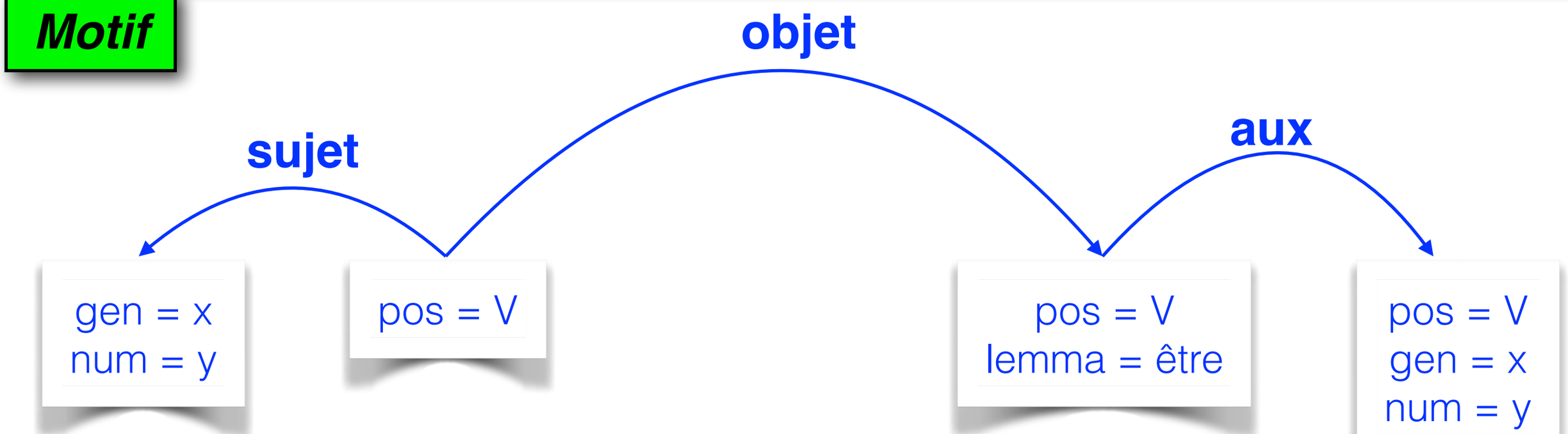


L'exploration de corpus par « matching » de graphes

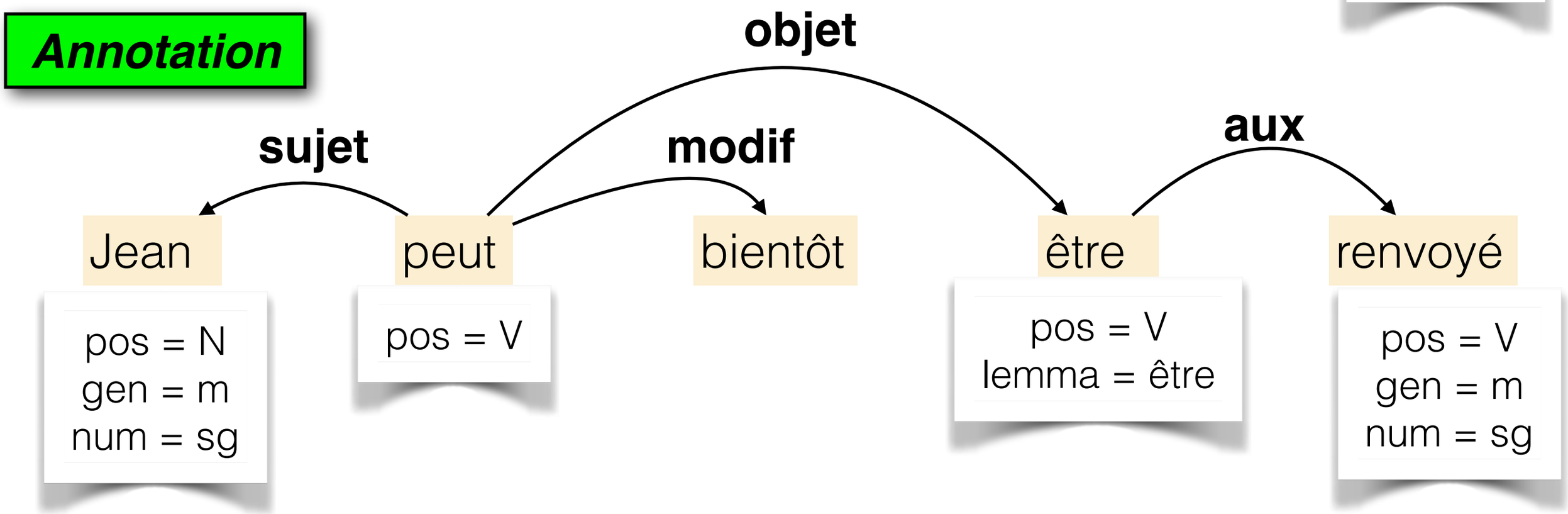
- Les linguistes ont souvent besoin d'étudier des occurrences de constructions particulières dans des corpus.
- Dans l'annotation de corpus, on a souvent besoin aussi de vérifier la cohérence et la conformité à certaines règles grammaticales d'une annotation (l'accord du sujet avec le verbe par exemple).
- Ces tâches peuvent être implémentées par « **matching** » d'un **motif** avec un **graphe**, le motif étant lui-même un graphe.

Exemple de « matching » d'un motif avec une annotation

Motif

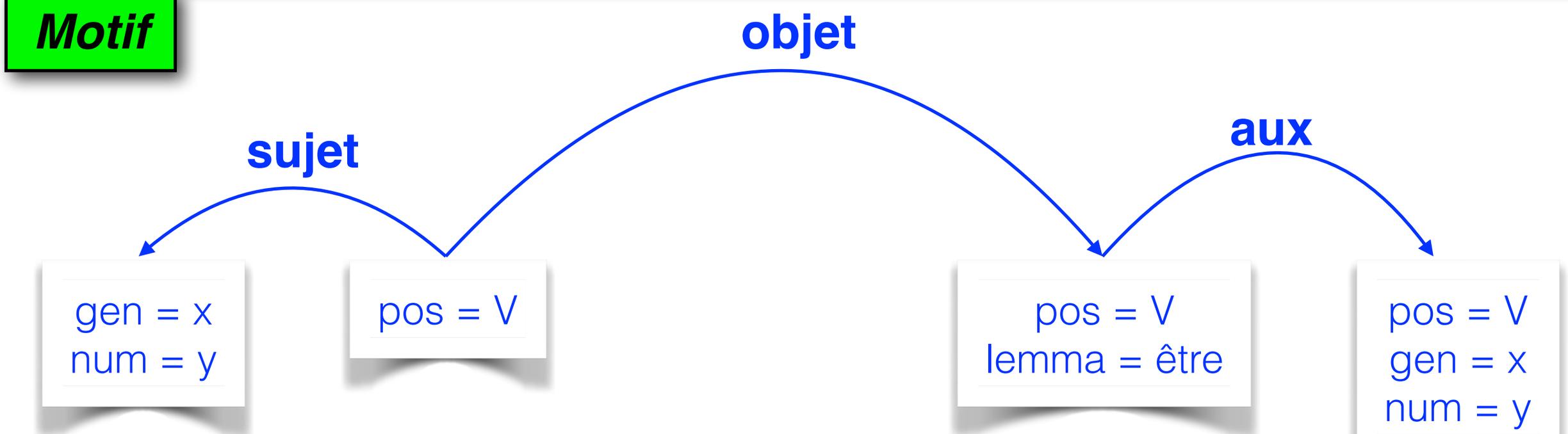


Annotation

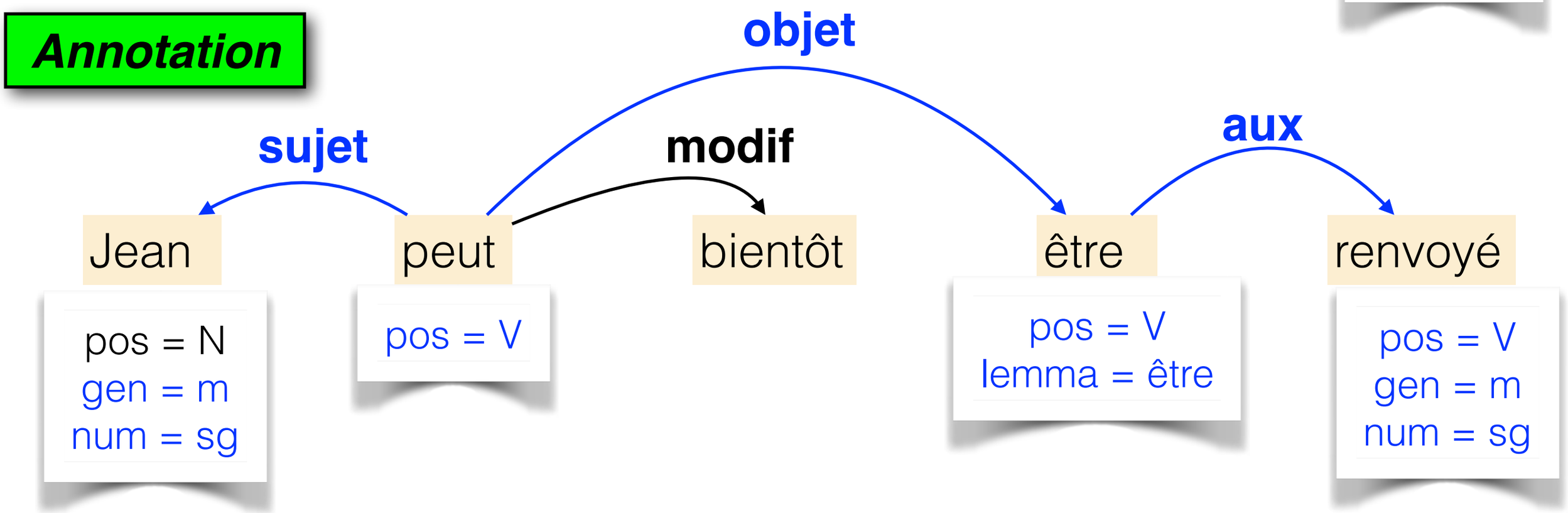


Exemple de « matching » d'un motif avec une annotation

Motif



Annotation



L'implémentation du « matching » de graphes dans GREW-MATCH

- GREW-MATCH est le module de GREW qui effectue le « **matching** » **de graphes**.

- GREW-MATCH est utilisable en ligne (http://match.grew.fr/) sur un ensemble de corpus de 97 langues issus du projet Universal Dependencies annotés principalement selon deux formats syntaxiques : **UD** et **SUD**.

Exemples d'utilisation de Grew-match

- On peut rechercher un motif dans un corpus annoté :
  - pour détecter des incohérences et des erreurs d'annotation dans un corpus, des erreurs d'accord par exemple,
  - pour étudier un phénomène linguistique particulier, l'ordre verbe sujet selon différentes langues par exemple
- On peut partitionner les résultats d'une requête en fonction de différents critères.
- On peut utiliser un tableau récapitulatif de toutes les relations présentes dans un corpus donné.

La réécriture de graphes

- La réécriture de graphes est un modèle de calcul qui utilise des **systèmes de règles**. Une règle de réécriture de graphes est formée de deux parties :
  - une **partie gauche** qui représente un motif à remplacer dans un graphe,
  - une **partie droite** qui représente un motif qui va venir se substituer au motif à remplacer.

La réécriture de graphes

- Un système de règles de réécriture ne **termine** pas nécessairement et n'est pas toujours **confluent**.

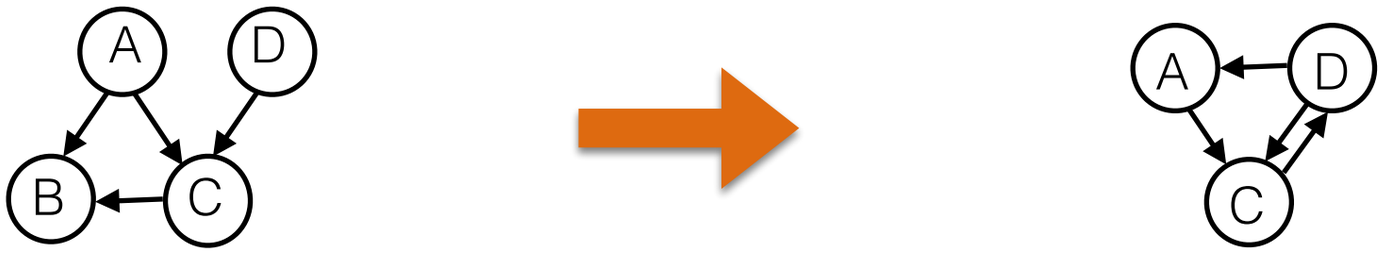
- Ce modèle est bien adapté au traitement des langues, les règles de réécriture, étant **locales**, permettent de représenter simplement des règles linguistiques tout aussi locales.

- La non confluence des calculs permet de représenter l'**ambiguïté** inhérente aux langues.

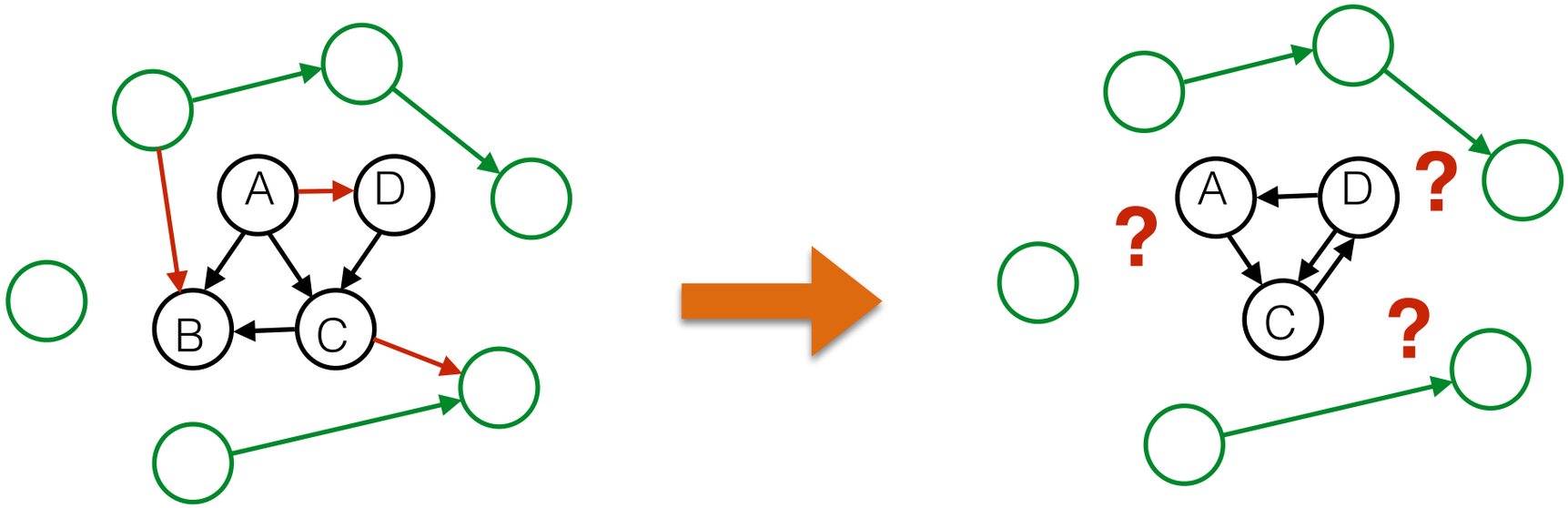
La réécriture de graphes

Pas de définition mathématique canonique

règle



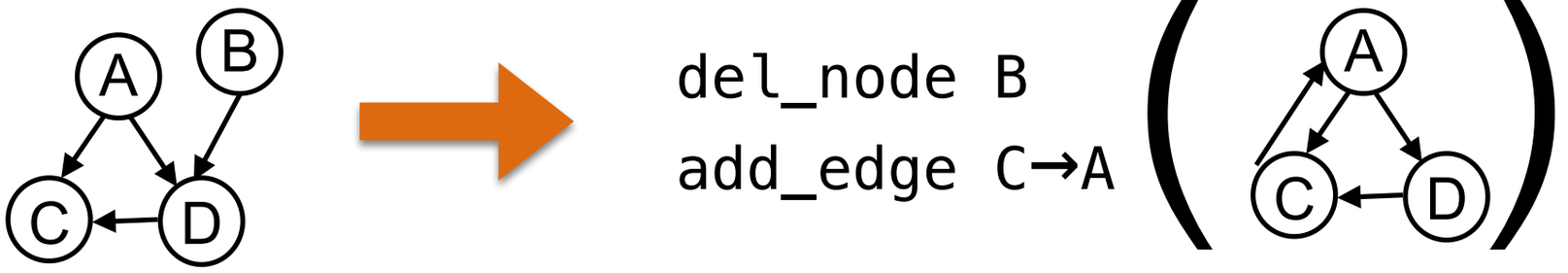
application
d'une règle



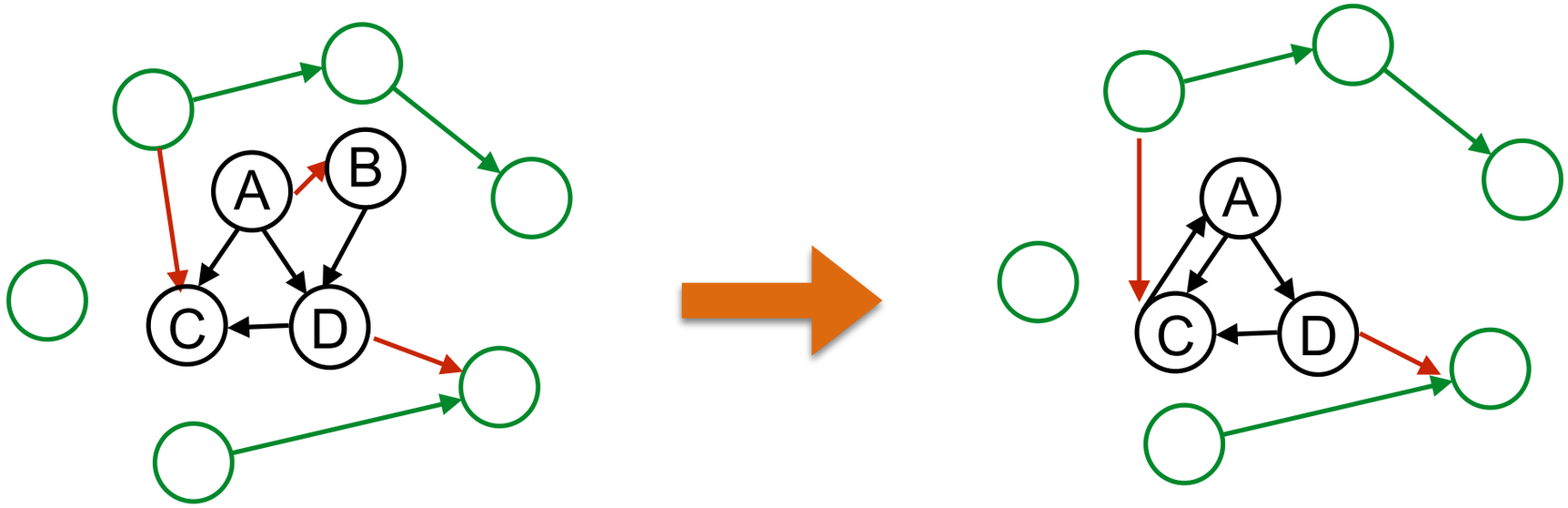
La réécriture de graphes

On propose une définition opérationnelle

règle



application d'une règle



L'implémentation de la réécriture de graphe dans **Grew**

Grew est une implémentation de la réécriture de graphes :

- où les transformations sont décrites par des **commandes**,
- dédiée aux applications **TAL** :
  - gestion des **structures de traits** sur les nœuds,
  - les arcs codent des **relations** (avec des étiquettes, pas de doublons),
  - les règles peuvent être contrôlées par des **lexiques.**

http://grew.fr

L'application de **Grew** au traitement de corpus

Grew, en tant que **transformateur de graphes**, est utilisé pour :

- corriger des erreurs régulières d'annotation d'un corpus,
- convertir l'annotation d'un corpus d'un format dans un autre (de SUD dans UD par exemple),
- passer d'une annotation d'un niveau linguistique à un autre (du niveau syntaxique au niveau sémantique par exemple).

Le format d'annotation syntaxique UD

- UD est un format d'annotation au centre du projet **Universal Dependencies** qui se veut universel, c'est-à-dire utilisable pour n'importe quelle langue.

- L'annotation est proche de la sémantique : les têtes des dépendances sont des **mots lexicaux** et les **mots grammaticaux** sont des marqueurs des mots lexicaux.

- Les noms de relations prennent en compte non seulement la **fonction syntaxique** qu'elles représentent mais aussi les **POS** des mots qu'elles relient (*obj, ccomp, xcomp* pour la fonction objet direct par exemple).

Le format d'annotation syntaxique SUD

- SUD est un schéma d'annotation syntaxique proche de la forme de surface et fondée sur une **approche distributionnelle**.

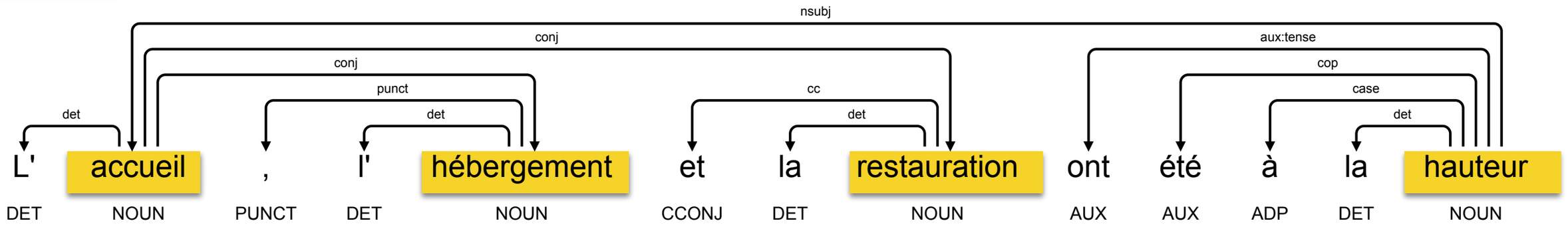
- Les têtes des dépendances contrôlent la distribution des unités linguistiques et les noms des dépendances prennent en compte uniquement les **fonctions syntaxiques** (*subj, comp, mod*).

- Les relations sont organisées en une **hiérarchie** à l'aide notamment d'extensions des noms (*comp:aux, comp:obj, comp:obl, comp:pred, comp:cleft*).

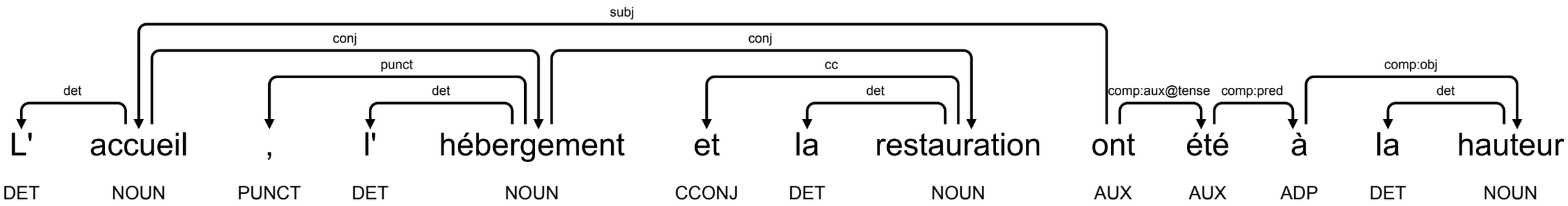
- Une extension supplémentaire commençant par @ permet d'ajouter des **informations sémantiques**.

Exemple d'annotation UD et SUD

UD

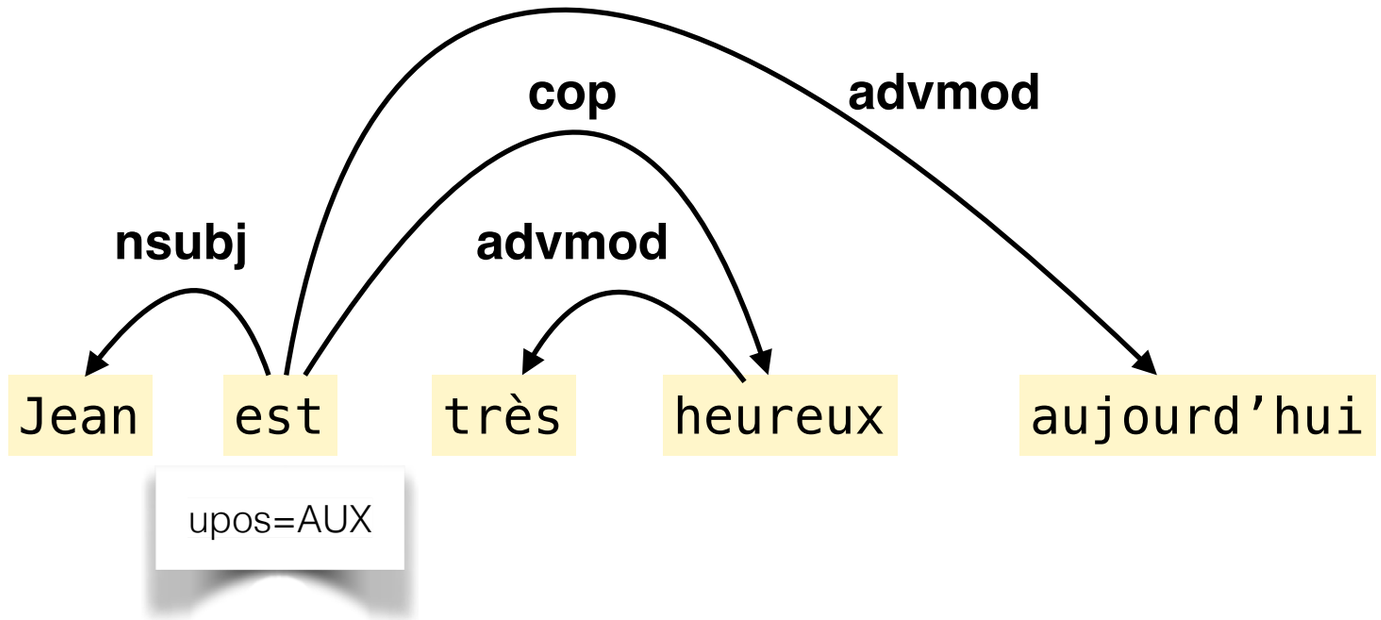


SUD



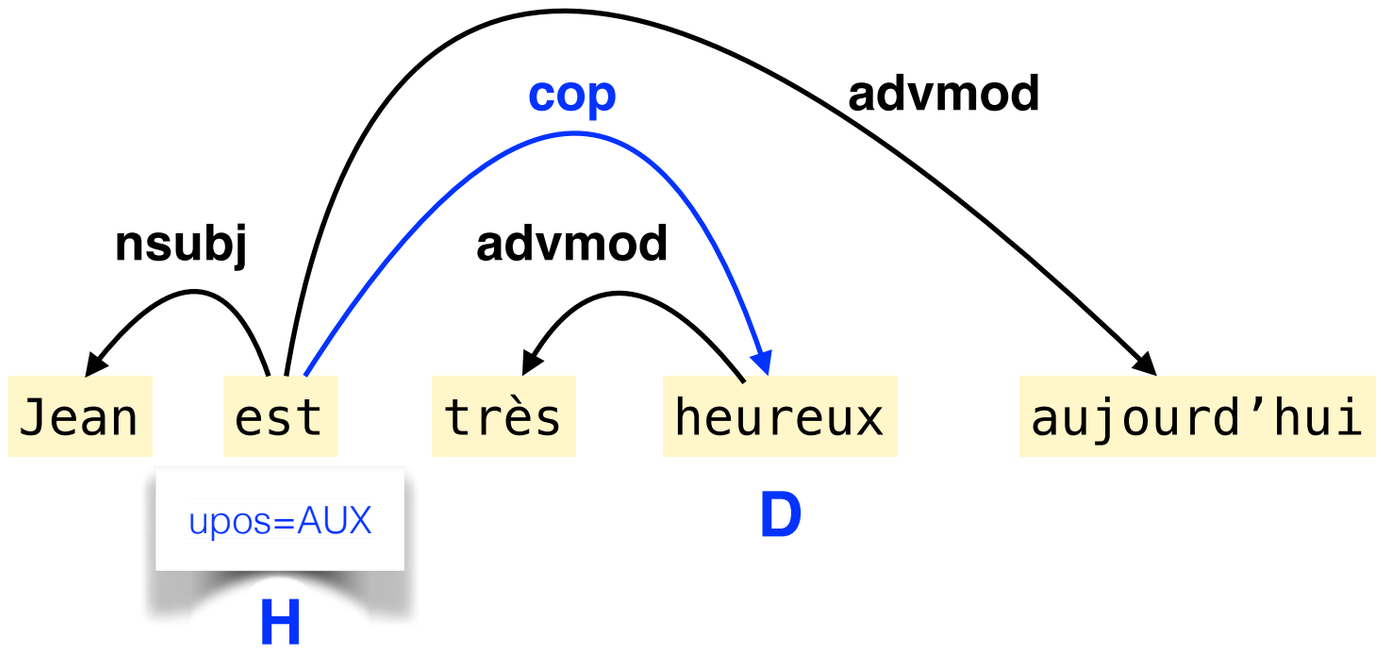
Exemple de règle de conversion de SUD en UD

```
rule rev_head {
  pattern {
    H[upos=AUX];
    e: H -[1=aux|cop]-> D;
  }
  without {
    D[upos=AUX];
    D -[1=aux|cop]-> D1
  }
  commands {
   add_edge f:D -> H;
    f.label = e.label;
    del_edge e;
    shift H ==> D;
 }
}
```



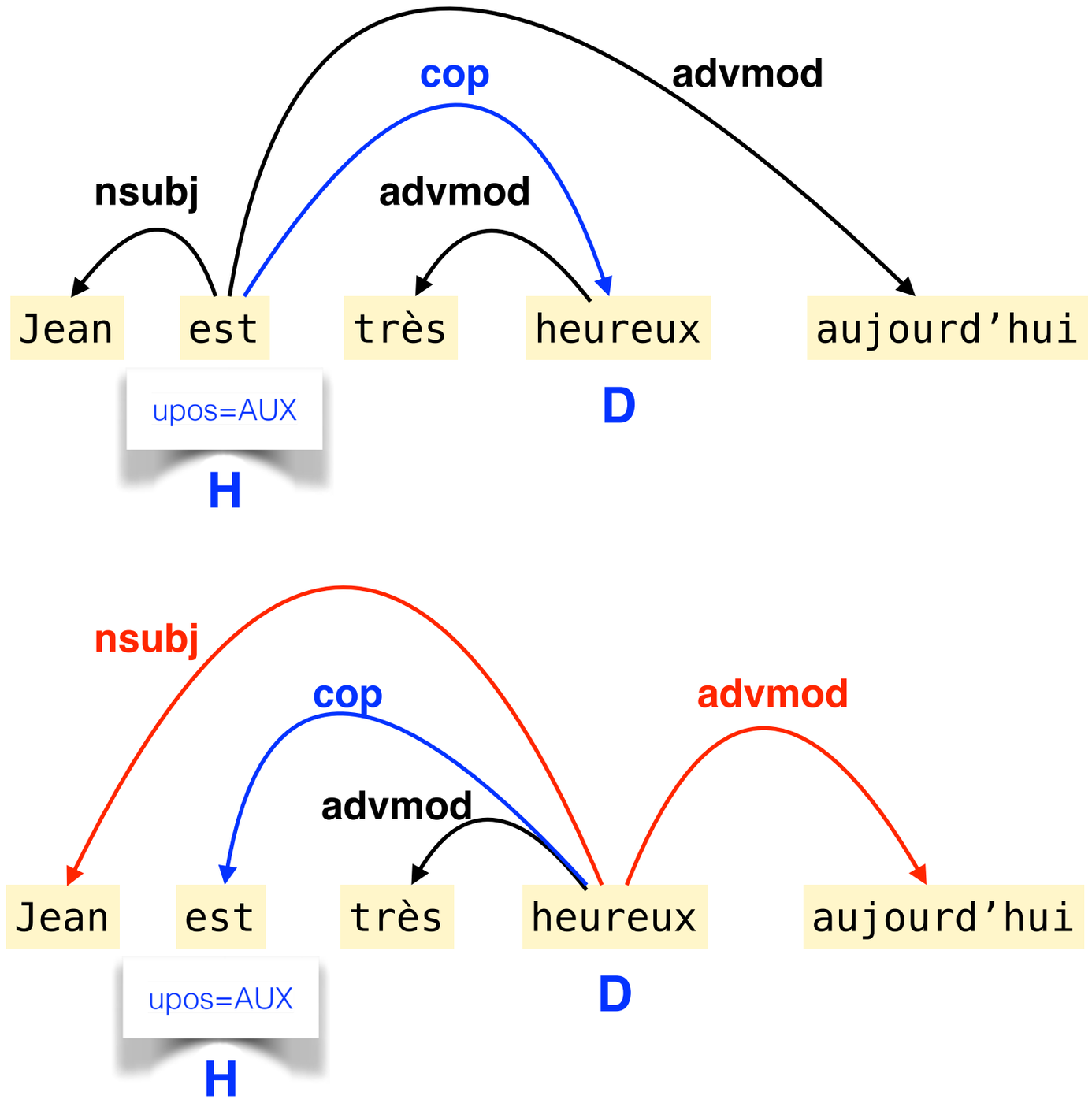
Exemple de règle de conversion de SUD en UD

```
rule rev_head {
  pattern {
    H[upos=AUX];
    e: H -[1=aux|cop]-> D;
  }
  without {
    D[upos=AUX];
    D -[1=aux|cop]-> D1
  }
  commands {
    add_edge f:D -> H;
    f.label = e.label;
    del_edge e;
    shift H ==> D;
 }
}
```

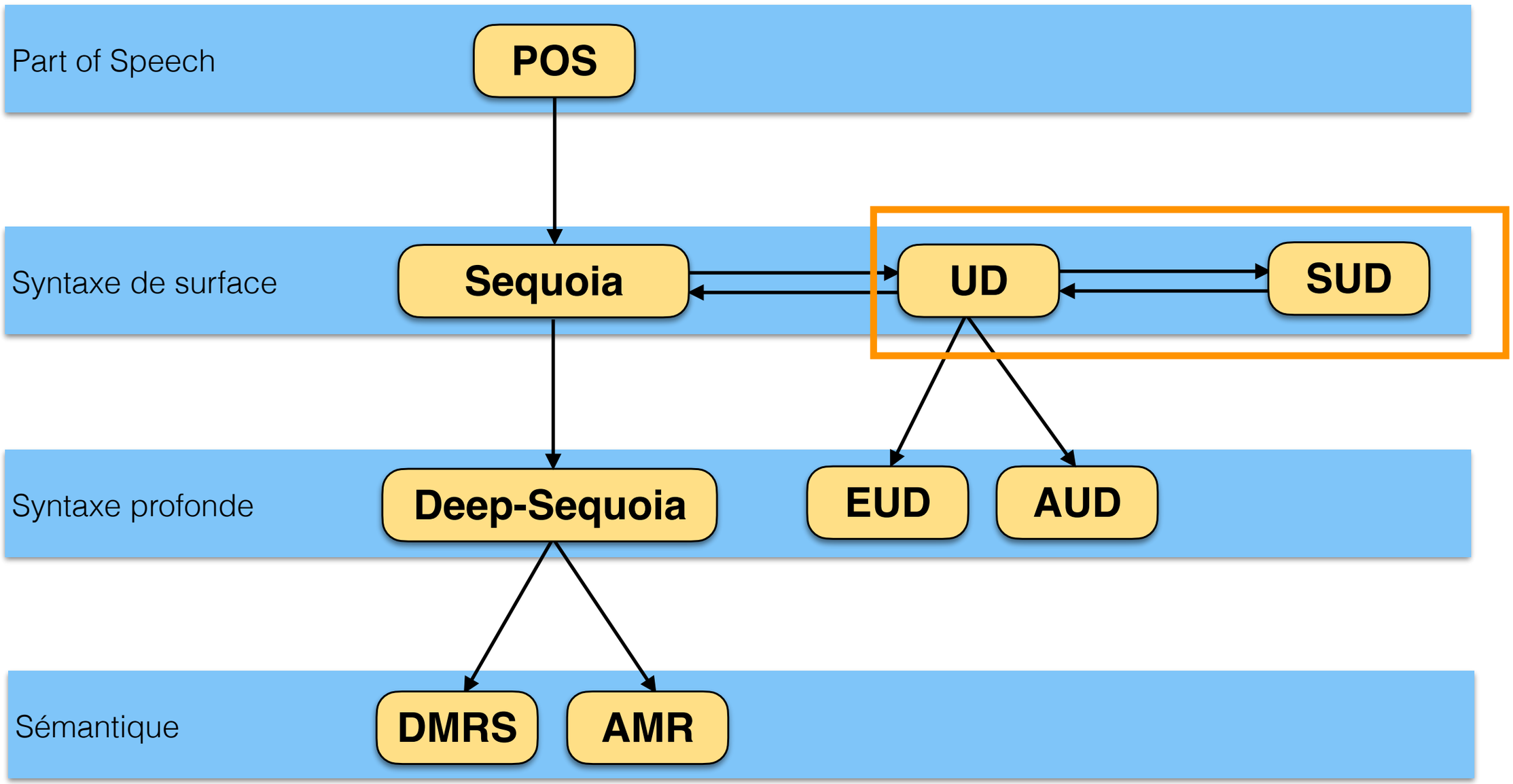


Exemple de règle de conversion de SUD en UD

```
rule rev_head {
  pattern {
    H[upos=AUX];
    e: H -[1=aux|cop]-> D;
  }
  without {
    D[upos=AUX];
    D -[1=aux|cop]-> D1
  }
  commands {
   add_edge f:D -> H;
    f.label = e.label;
    del_edge e;
    shift H ==> D;
 }
}
```



Systèmes de réécriture de graphes existants



La conversion entre UD et SUD

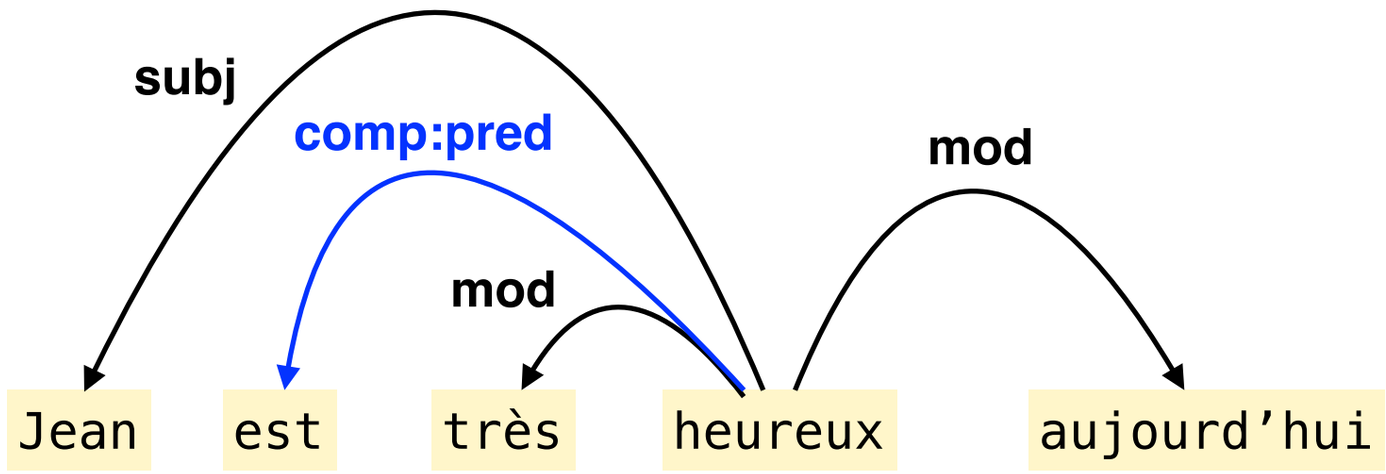
- Le format SUD est plus riche que le format UD, donc la conversion de SUD vers UD ne pose pas de problème, alors que la conversion inverse comporte nécessairement des erreurs.

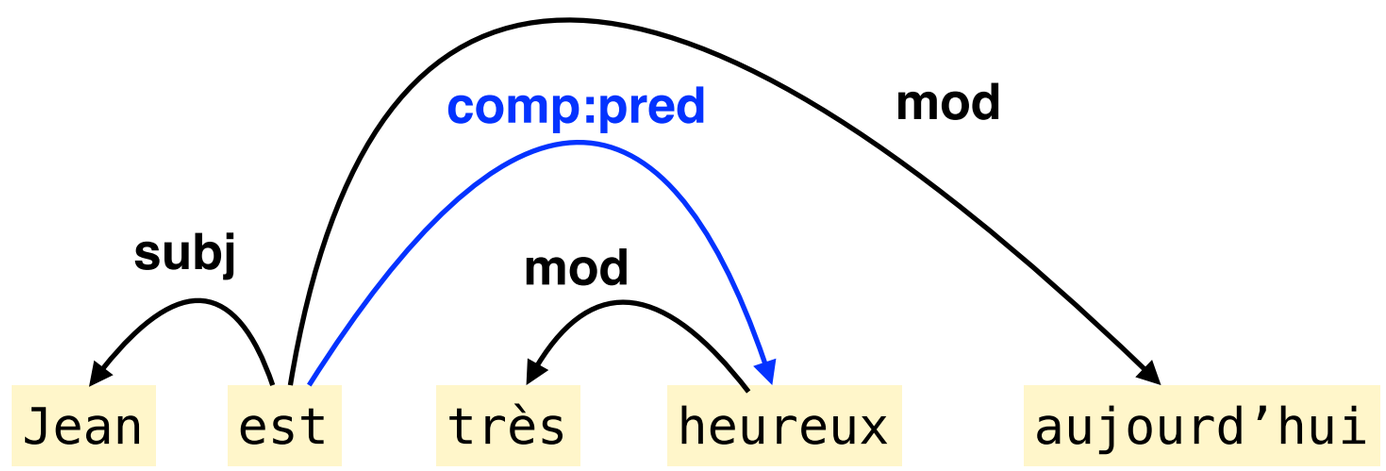
- Il y a deux façons de limiter les erreurs dans la conversion de UD vers SUD :

  - en ajoutant de l'information lexicale, les règles de Grew pouvant utiliser des lexiques,

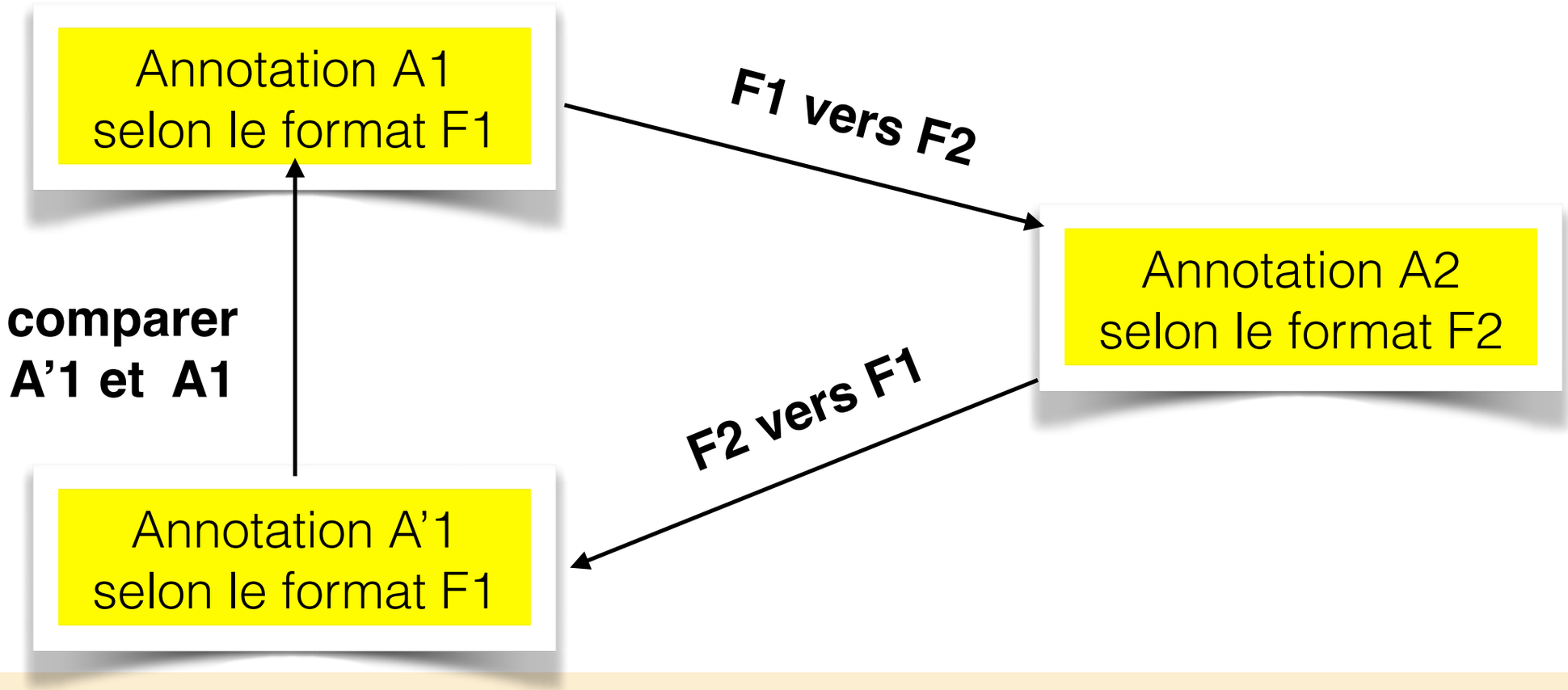
  - en enrichissant l'annotation UD par l'ajout d'extensions aux noms de relations.

Exemple de problème de conversion de UD en SUD





Correction de corpus par double conversion



- La comparaison de l'annotation A'1 avec A1 permet de détecter des erreurs dans A1 ou dans les 2 systèmes de règles de conversion.

- Cette méthode a été appliquée à la vérification de l'annotation du corpus French-GSD selon les formats UD et SUD.

Évaluation du corpus GSD

- Il est difficile d'avoir une mesure sur un corpus déjà constitué.
- 2 évaluations ont été menées sur le corpus GSD.
- Première évaluation : 108 phrases sélectionnées aléatoirement ont été annotées manuellement et comparées à l'annotation à évaluer.

	A1/A2	A1/A3	A2/A3	Moyenne
Résultats bruts	89,11	85,97	93,42	89,50
Après normalisation des dates	90,33	88,08	93,42	90,61

Tableau 4. *Accords entre annotateurs pour le corpus de référence (108 phrases)*

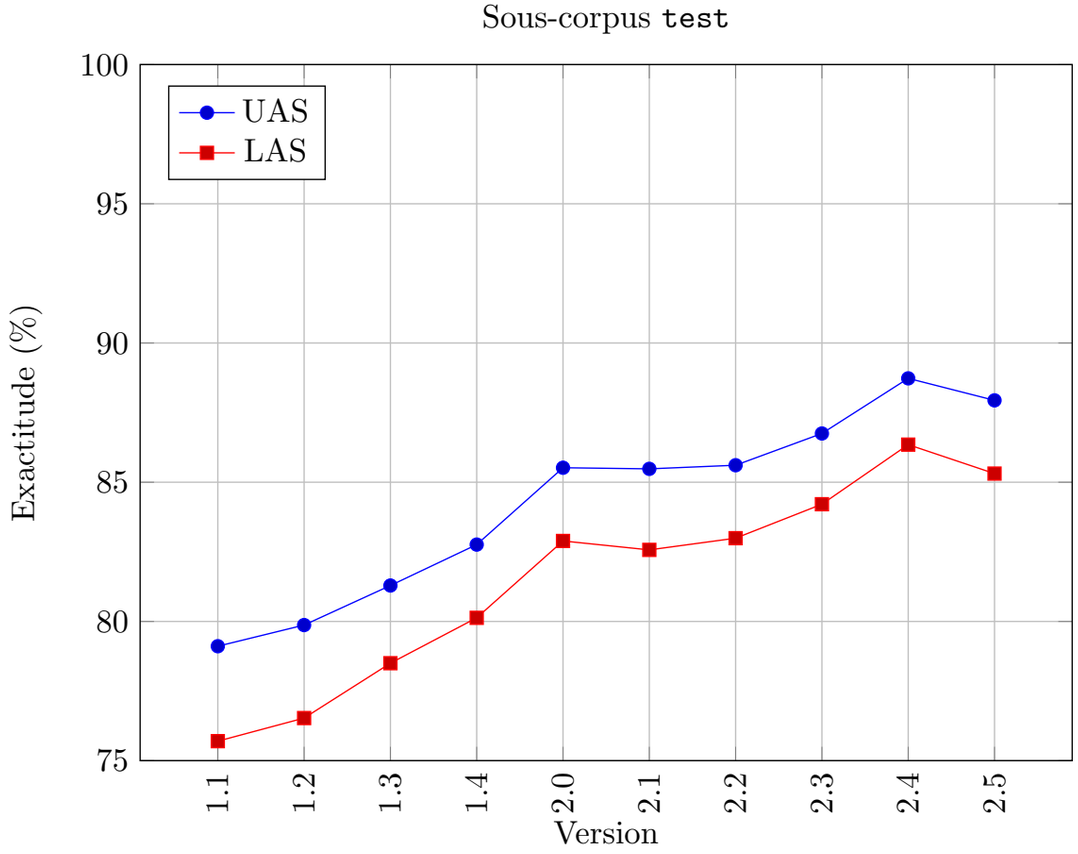
Version	2.0	2.1	2.2	2.3	2.4
Exactitude (%)	85,13	87,86	88,35	91,12	92.00

Tableau 5. *Exactitude pour les différentes versions du corpus* UD_FRENCH-GSD

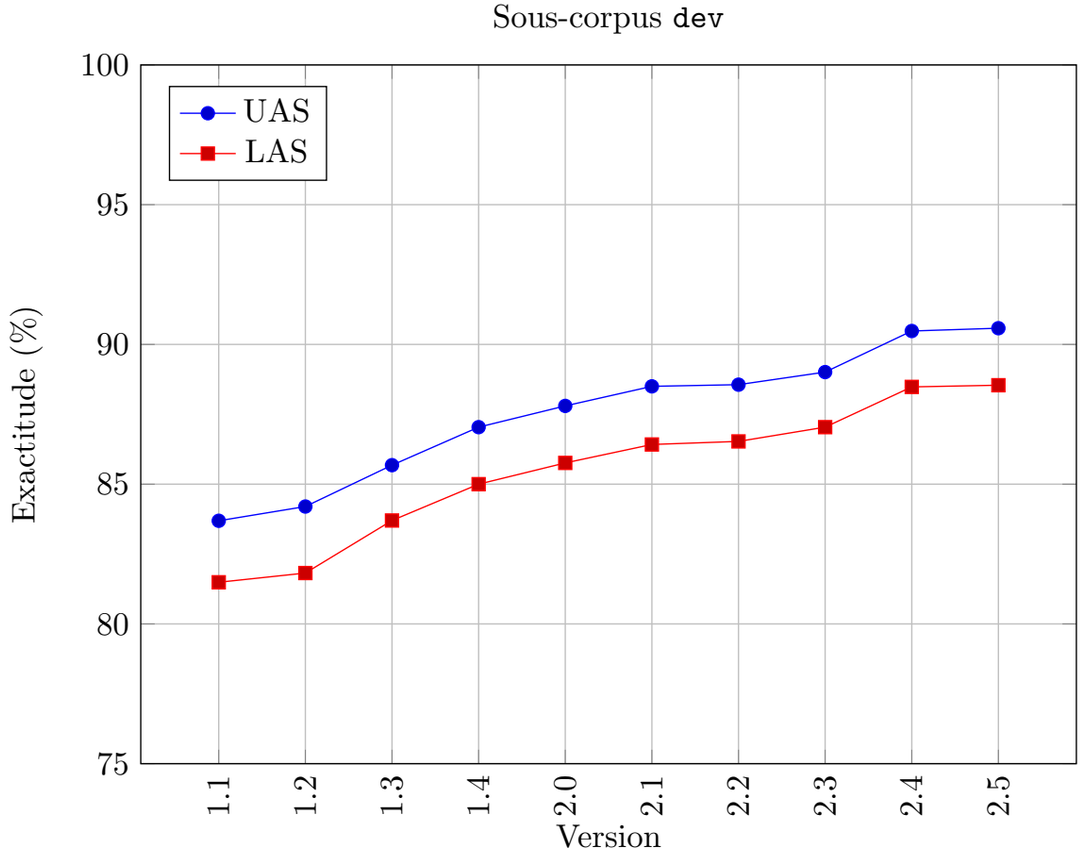
Évaluation du corpus GSD

- La seconde évaluation s'est faite par analyse syntaxique de l'ensemble du corpus avec UDPIPE et comparaison de l'annotation obtenue avec celle à évaluer.

Sous-corpus `test`



Sous-corpus `dev`



Conclusion

- **Grew-match** (http://match.grew.fr) a été et continue à être utilisé par des linguistes et des informaticiens pour étiqueter des corpus très divers (Naija, Wolof, French Spoken).
- Il est couplé avec **Arborator** (https://arborator.ilpga.fr/) qui est un outil d'annotation manuel de corpus.
- De nouvelles fonctionnalités vont être installées, notamment la possibilité pour l'utilisateur de télécharger ses propres corpus.

- **Grew** (https://grew.fr/) a été utilisé pour convertir les 184 corpus du projet Universal Dependencies du format UD vers le format SUD.
- Il va être utilisé pour transformer des annotations syntaxiques en **annotations sémantiques** (formats AMR et DMRS de représentation sémantique sous forme de graphes)

Bibliographie