

lien Google Docs :

<https://docs.google.com/document/d/12B6kNa2pw3YAq4Scpckj98agtEVqs5LBF6Bor4qh0X0/edit?usp=sharing>

général

- ☒ ExtPos sur les NUM
- ☒ quelques, différents, divers → ADJ et det

Rhapsodie

- ☐ il y avait des N comp:obl

Guide

- ☐ j'étais en talons : AUX ou VERB
- ☐ alors et donc : avoir un critère pour décider si mod ou discourse, genre commutation par ainsi ou par conséquent
- ☒ Polarity = Neg : décider pour "que" par ex. + règle de validation
- ☐ interrogatives
- ☐ rattachement des discourse : pas à un autre discourse, cc, dislocated pas toujours au mod : si tu viens, ben, ça va être compliqué sauf les "euh", qui sont toujours attachés au pied de la voûte  
<http://universal.grew.fr/?custom=63dfeed54be82>
- ☐ problème de la ponctuation avec le discours rapporté : le point final du discours rapporté n'est pas au pied de la voûte. Il faut une règle disant qu'un mot Reported=Yes doit avoir un punct !
- ☐ segment inachevé sans tête : unk@scrap, ex on s'en, on s'en fout ; nous, on, moi, j'étais
- ☐ c'est X : quand est-ce que "ce" est considéré comme impersonnel ? quand on peut ajouter "ce truc" devant ? (ce truc, c'est bizarre ; \*ce truc, c'est impossible qu'il vienne ; c'est elle vs c'est qui l'a fait)

- ☐ Sy : problème exemple

[https://surfacesyntacticud.github.io/guidelines/u/oral\\_language/conj\\_coord/](https://surfacesyntacticud.github.io/guidelines/u/oral_language/conj_coord/)

Ile saura conjuguer le changement et la cohésion sociale l' esprit d' initiative et la sécurité la modernité et le bien vivre ensemble

c'est un cas de embedded coordination (intéressant en plus)

à corriger et mettre sur

[https://surfacesyntacticud.github.io/guidelines/u/particular\\_phenomena/coord/](https://surfacesyntacticud.github.io/guidelines/u/particular_phenomena/coord/)

- ☐ mettre aussi un cas de @emb sur la page /u/oral\_language/conj\_coord/

- ☒ il y a
- ☒ lemme de je et tu : ils ont le lemme il ? mais pas on ?
- ☒ LEMMES : pas mal d'erreurs sur les lemmes des V : cherchez tous les lemmes qui ne se terminent pas par r ou re.
- ☒ LEMMES : idem pour N avec lemme en -s, ADJ avec lemmes en -x, -s, -e
- ☒ lemme=rigolet => rigolo
- ☒ est-ce que aux@pass => subj@pass ?
- ☐ j ~: rechercher les ~ seuls
- ☒ Foreign => Lang=En ou en ?
- ☐ un DET vs NUM : NUM => ExtPos=PROPN
- ☒ mamie => Mamie
- ☒ oui : ADV => INTJ et NOUN => INTJ dans GSD
- ☐ anonymisation des noms propres : NAME\_X dans PS
- ☐ Name-X, name\_X, NAME\_X : unifié et vérifier la POS
- ☒ checker tous les "ne \* que" et voir si que est ADV et bien rattaché ; a-t-on Polarity = Neg ?
- ☒ en fait : vérifier Idiom, + POS de fait
- ☒ "vu" dans "vu le problème" ou "vu que ..." : ExtPos=ADP
- ☒ "genre" avec ExtPos=INTJ, vérifier Number et Gender
- ☒ tutoie moi => tutoie moi
- ☒ La première règle concerne l'ajout des Miscellaneous Features à la locution conjonctive "parce que". En effet, il manque de manière générale les MF "ExtPos = "SCONJ"" et "Idiom = "Yes"" à l'ADV "parce" et la MF "InIdiom="Yes"" à la SCONJ "que". Afin de les identifier, nous utilisons le pattern de recherche de form "parce" et "que", "que" suivant directement "parce" ( pattern {N1 [form="parce"]; N2 [form="que"]; N1 < N2} ).
- ☒ La seconde règle concerne l'ajout de l'Universal Feature "Polarity=Neg" à l'adverbe "pas" dans les constructions négatives. En effet, cette UF manque de manière générale à l'ADV "pas" alors qu'il est présent dans les autres corpus. Afin d'y parvenir, nous identifions le pattern de "pas" selon sa form et son upos (ADV).
- ☒ La troisième règle concerne l'ajout de l'Universal Feature "Poss=Yes" aux déterminants possessifs. Comme leur lemme est "son", c'est le critère de recherche. Nous identifions le pattern d'un élément dont le lemme est "son" et dont l'upos est DET, pour lui ajouter l'Universal Feature "Poss=Yes".
- ☒ accord du sujet en personne, nombre : vérifier pour repérer les traits faux ou manquants
- ☐ dislocated ou conj:appos : pas possible de décider sans prosodie
- ☐ TRAITS MISC : Subject, Idiom, InIdiom, AttachTo
- ☒ "à peu près" 1) "à X près" quelle analyse ? ; 2) ajouter Idiom

- ☐ locution "ADV que" à discuter avec Guy : ADV <-[mod]- que
- ☐ bcp d'erreurs dans la fonction des à/de Vinf : <http://universal.grew.fr/?custom=63de94470ba4c>
- ☒ il y a : annotation trop compliquée ; il est sujet impersonnel, mais y est juste figé ; ne pas confondre @expl et figement
- ☒ trait sur dépendance vs sur nœud : @expl vs Impersonal=Yes, @fixed vs Idiom=Yes, @scrap, @lvc, @tense + Tense, @pass + Voice=Pass
- ☒ lemme des pronoms : comp:obj => le, comp:obl => lui
- ☐ (avoir) rien à Vinf : fonction de "à Vinf" ?
- ☐ prend ton courage => prends
- ☒ lemme de seconde = seconde et pas second : vérifier que le lemme d'un nom est bien la forme, la forme sans s, etc.
- ☐ "j'en ai essayé deux, trois" : "en" udep de "trois" ?
- ☐ "une à deux" => conj:coord
- ☒ problème des pronoms comme "deux" : <http://universal.grew.fr/?custom=63dea8eade7bf>
- ☐ compléments de nom : encore plein de "mod" sur PS
- ☒ "vu" ADP mais lemma=voir
- ☐ polémiques + conj:appos qui du Reported=Yes
- ☐ Reflex=Yes + lemma=lui
- ☐ NOUN-[mod]->ADP => udep, sauf LVC
- ☐ etre en PROPN => comp:obl <http://universal.grew.fr/?custom=63dfe516a59e6>
- ☐ nanana: INTJ ?
- ☒ non plus: pas homogène entre GSD et autres
- ☐ "AUX pas V" avec "pas" sur AUX : <http://universal.grew.fr/?custom=63dfeaf35d4bb>
- ☐ "pas du tout" : homogénéiser <http://universal.grew.fr/?custom=63dfeed54be82>
- ☒ "mais bon" expression figée ? <http://universal.grew.fr/?custom=63dfeed54be82>
- ☐ hum, hm => mh <http://universal.grew.fr/?custom=63dfeed54be82>
- ☐ peut-être que
- ☒ me comp:obj et lemma = lui
- ☐ je n'ai rien à faire, qu'as-tu à dire, j'ai mes vacances à organiser, j'ai un article à terminer, ??c'est un autre article que j'ai à terminer : comp:obj ou udep
- ☐ discours rapporté : expliquer les deux annotations, comp:obj et parataxis:insert
- ☐ Reported mal placés : <https://universal.grew.fr/?custom=64649e0daf4e2>
- ☐ Reported=Yes sur les "en mode" et "être là"
- ☐ utiliser orphan pour "je me je me souviens"

- ☐ annoter les moyens, notamment les SE FAIRE :  
<https://universal.grew.fr/?custom=65240f4ebd48e>
- ☐ upos=X => ExtPos

#### GSD

- ☐ VERB -[subj]-> DET : <https://universal.grew.fr/?custom=6584525c6fe9f>
- ☒ ~~cherchez les doubles PROP~~N, comme Etats Unis
- ☐ @intcl: voir SUD 2024
- ☐ analyse des relatives sans tête avec pronom ?!
- ☐ revoir les traits pour les participes "passés" et "présents"

#### FTB :

- ☒ ~~corriger la relation unk avec mwehead pour les NC+, NPP+, DET+~~
- ☐ continuer de corriger la relation unk avec mwehead pour les catégories restantes  
pattern { e: X -[unk]-> Y; X[mwehead]; X<Y }  
without { \* -[unk]-> X }  
without { Y -[unk]-> \* }  
Key: X.mwehead
- ☒ ~~concaténer deux tokens séparés par un tiret en un~~
- ☐ concaténer deux nombres séparés par une virgule en un
- ☐ vérifier les lemmes des mwehead avec tiret
- ☐ supprimer les features en trop des mwehead avec tiret
- ☐ vérifier le genre et nombre des noms communs mwehead avec tiret
- ☐ ajouter les traits ctxt
- ☐ ajouter les traits denom sur les participes
- ☐ ajouter les traits lex sur les noms
- ☐ ajouter les traits Voice sur les participes
- ☐ ajouter les traits Filler aux éléments clivés
- ☐ ajouter la fonction aux dislocated(:obj, :subj, :obl, :mod, :unk...)
- ☐ continuer de corriger les relations syntaxiques dans les expressions figées et d'ajouter les traits manquants  
corrigés : en fait, en effet, par exemple, en particulier, en général, de plus, en plus, à nouveau, à part, quelque chose, pour que, il y a (expression figée et expression impersonnelle)
- ☒ ~~corriger la relation entre "M."/"Mme" + PROP~~N
- ☐ continuer la correction des relations conj mal annotées, à l'exception des cas conj:coord déjà traités
- ☐ vérifier et corriger les relations subj@expl et comp@expl dans les expressions impersonnelles

- ☐ continuer de corriger les relations comp mal annotées selon les stuctures syntaxiques
- ☒ ~~corriger les structures comparatives~~
- ☐ corriger et vérifier les constructions clivées :  
<https://universal.grew.fr/?custom=687a64e01a933>
- ☐ corriger les relations dep mal annotées
- ☒ ~~corriger les lemmes des pronoms au nominatif et ajouter les traits Emph et PronType~~
- ☐ corriger les lemmes et les traits des autres pronoms

## Règles à appliquer sur les futurs corpus.

*Les règles ci-dessous sont générales pour donner une idée de la tâche, mais nécessitent d'être adaptées et affinées selon les cas et qui peuvent requérir une annotation (vérification) manuelle, la liste étant exhaustive.*

### **Ajout des traits ctxt :**

Faire tourner tous les scripts grs correspondants au corpus (oral ou écrit) sur le corpus en question.

<https://github.com/surfacesyntacticud/tools/tree/master/french/process>

### **Ajout des traits Case et Emph pour le nominatif, l'accusatif et le datif :**

#### **# Nominatif :**

```
rule nom {
  pattern {
    X[upos="PRON",
    form="Je"|"je"|"je"|"je"|"j"|"J"|"-j"|
    "Tu"|"tu"|"tu"|"T"|"t"|"t"|
    "Il"|"il"|"il"|"Ils"|"ils"|"ils"|
    "Elle"|"elle"|"elle"|"Elles"|"elles"|"elles"|
    "Nous"|"nous"|"nous"|
    "Vous"|"vous"|"vous"|
    "On"|"on"|"on",
    !Case];
    Y[upos];
    Y-[subj]->X;
  }

  commands {
    X.Case=Nom;
    X.Emph=No;
  }
}
```

#### **# Accusatif :**

```
rule acc {
  pattern {
    X[upos="PRON",
    form="me"|"m"|
    "te"|"t"|
    "le"|"la"|"l"|
    "les"|
  }
```

```

        "nous"|"vous"];
Y[upos];
Y-[comp:obj]->X
}
without { X.Emph=Yes }
without { X.Case=Acc }

commands {
    X.Case = Acc;
    X.Emph = No;
}
}

```

### **# Datif :**

```

rule dat {
    pattern {
        X[upos="PRON",
        form="me"|"m"|
        "te"|"t"|
        "lui"|
        "leur"|
        "nous"|"vous"];
Y[upos];
Y-[comp:obl]->X
}
without { X.Emph=Yes }
without { X.Case=Dat }

commands {
    X.Case = Dat;
    X.Emph = No;
}
}

```

### **Ajout du trait Gender[ctxt] sur les pronoms possessifs devant une voyelle :**

```

rule poss_ctxt {
    pattern {
        det [form="mon" | "ton" | "son", upos=DET, Gender];
        head [form=/^[aeiouy].*/];
    }
}

```

```

        head > det;
    }

    commands {
        det.Gender__ctxt=det.Gender;
        del_feat det.Gender
    }
}

```

### **Ajout du trait Gender[ctxt] sur le déterminant "l" devant une voyelle :**

#### **# Devant les noms communs**

```

rule det_ctxt_nc {
    pattern {
        D [form="l", upos="DET", !Gender__ctxt];
        N [upos="NOUN", Gender__lex];
        N-[det]-> D
    }

    commands {
        D.Gender__ctxt = N.Gender__lex;
        del_feat D.Gender
    }
}

```

#### **# Devant les noms propres**

```

rule det_ctxt_propn {
    pattern {
        D [form="l", upos="DET", !Gender__ctxt];
        P [upos="PROPN", Gender];
        P-[det]-> D
    }

    commands {
        D.Gender__ctxt = P.Gender;
        del_feat D.Gender
    }
}

```

#### **# Devant les autres POS**



```

rule det_ctxt_special {
    pattern {
        D [form="l'", upos=DET, !Gender__ctxt, !Gender];
        S[Gender];
        S-[det]->D;
    }

    commands {
        D.Gender__ctxt = S.Gender
    }
}

```

### **Ajout du trait Tense[denom] sur les participes :**

```

rule part_denom {
    pattern {
        verb [upos=VERB, VerbForm="Part", Tense, !Tense__denom];
    }

    commands {
        verb.Tense__denom = verb.Tense;
        del_feat verb.Tense;
    }
}

```

### **Ajout du trait Gender[lex] sur les noms :**

#### **# Noms communs**

```

rule noun_lex{
    pattern { noun [upos="NOUN", Gender, !Gender__lex] }

    commands {
        noun.Gender__lex = noun.Gender;
        del_feat noun.Gender
    }
}

```

#### **# Noms propres :**

```

rule gender_unk {
    pattern { propn [upos="PROPN", !Gender, !Gender__lex] }
}

```

```

        commands {
            propn.Gender__lex = "Unknown"
        }
    }
}

```

### **Ajout du trait Number[lex] sur les numéraux :**

#### **# ExtPos = PROPN → Number[lex] = Sing**

```

rule num_lex_propn {
    pattern { X [upos="NUM", ExtPos="PROPN", Number, !Number__lex ] }

    commands {
        X.Number__lex = "Sing";
        del_feat X.Number
    }
}

```

#### **# ExtPos = PRON → Number[lex] = Plur**

```

rule num_lex_pron {
    pattern { X [upos="NUM", ExtPos=PRON, Number, !Number__lex] }

    commands {
        X.Number__lex = "Plur";
        del_feat X.Number;
    }
}

```

#### **# ExtPos = DET → Number[lex] = Plur**

```

rule num_lex_det {
    pattern { X [upos="NUM", ExtPos=DET, Number, !Number__lex] }

    commands {
        X.Number__lex = "Plur";
        del_feat X.Number;
    }
}

```

### **Ajout du trait Polite=Yes/No sur les “vous”**

ajout du trait Polite="Unknown" aux “vous” qui sont sujet de l’auxiliaire avoir :

```

rule polite_unk {

```

```

        pattern { S[form="vous", !Polite]; V->S; V[upos=AUX, lemma="avoir"];
V-[2=aux]->A }
        without { O-[vocative]->X }
        without { V-[vocative]->X }

        commands {
        S.Polite="Unknown";
}
}

```

ajout du trait Polite="Yes" aux "vous" qui sont sujet de l'auxiliaire être et dont le complément est au singulier :

```

rule polite_yes {
    pattern { S[form="vous", !Polite]; V->S; V[upos=AUX, lemma="être"];
V-[1=comp]->A; A[Number=Sing] }

    commands {
    S.Polite="Yes";
}
}

```

ajout du trait Polite="No" aux "vous" qui sont sujet de l'auxiliaire être et dont le complément est au pluriel :

```

rule polite_no {
    pattern { S[form="vous", !Polite]; V->S; V[upos=AUX, lemma="être"];
V-[1=comp]->A; A[Number=Plur] }

    commands {
    S.Polite="No";
}
}

```

ajout du trait Polite=Yes aux "vous" dont le complément de l'auxiliaire dont il est sujet est singulier :

```

rule polite_yes {
    pattern { S[form="vous", !Polite]; V->S; V[upos=AUX]; V-[1=comp]->A;
A[Number=Sing] }

    commands {
    S.Polite="Yes";
}
}

```

```
}  
}
```

ajout du trait Polite=No aux "vous" dont le complément de l'auxiliaire dont il est sujet est pluriel :

```
rule r {  
    pattern { S[form="vous", !Polite]; V->S; V[upos=AUX]; V-[1=comp]->A;  
A[Number=Plur|Number__ctxt=Plur] }  
  
    commands {  
        S.Polite="No";  
    }  
}
```

ajout du trait Polite="Unknown" aux "vous" dont le complément de l'auxiliaire dont il est sujet n'a pas de nombre :

```
rule r {  
    pattern { S[form="vous", !Polite]; V->S; V[upos=AUX]; V-[1=comp]->A }  
    without { A[Number=Sing] }  
    without { A[Number=Plur] }  
    without { A[Number__ctxtr=Sing] }  
    without { A[Number__ctxt=Plur] }  
  
    commands {  
        S.Polite="Unknown"  
    }  
}
```

ajout du trait Polite="Unknown" aux "vous" dont le verbe dont il est sujet n'est pas un auxiliaire et du moment qu'il n'y a pas de vocative dans la phrase :

```
rule r {  
    pattern { S[form="vous", !Polite]; V->S }  
    without { V.upos=AUX }  
    without { O-[vocative]->X }  
    without { V-[vocative]->X }  
  
    commands {  
        S.Polite=Unknown  
    }  
}
```

ajout du trait Polite="Yes" aux "vous" dont le verbe ou auxiliaire dont il est sujet est lié à un nom singulier par la relation vocative :

```
rule r {  
    pattern { X[form="vous", !Polite]; V->X; V-[vocative]->S; S[Number=Sing] }  
  
    commands {  
        X.Polite=Yes  
    }  
}
```

### **Ajout du trait Voice sur les participes :**

ajouter Voice="Act" à tous les participes passés dont l'auxiliaire est "avoir" :

```
rule r {  
    pattern{ X[upos=VERB, VerbForm=Part, !Voice]; A[upos=AUX, lemma="avoir"];  
e: A -[2=aux]-> X }  
    commands {  
        X.Voice="Act"  
    }  
}
```

ajouter Voice="Pass" à tous les participes passés qui ont une relation comp:aux@pass :

```
rule r {  
    pattern{ X[upos=VERB, VerbForm=Part, !Voice]; A[upos=AUX]; e: A -[2=aux,  
deep=pass]-> X }  
    commands {  
        X.Voice="Pass"  
    }  
}
```

ajouter Voice="Act" à tous les participes passés restants :

```
rule r {  
    pattern{ X[upos=VERB, VerbForm=Part]; A[upos=AUX]; e: A -[2=aux]-> X }  
    without { X.Voice="Pass" }  
    without { X.Voice="Act" }  
    commands {  
        X.Voice="Act"  
    }  
}
```

### **Ajout du trait Filler aux éléments clivés :**

ajouter Filler=subj aux éléments clivés dont le pronom est sujet :

```
rule r {
  pattern { X-[comp:cleft]->Z; Y[upos=PRON, PronType=Rel]; e: Z->Y; Z[!Filler] }
  commands {
    Z.Filler=subj;
  }
}
```

règle où l'élément clivé est mod :

```
rule r {
  pattern { X-[comp:cleft]->Y; X->A; A[upos=ADP, form="pour"]; Y[upos=SCONJ]; e: Y->Z;
  Z[!Filler] }
  commands {
    Z.Filler=mod
  }
}
```

```
rule r {
  pattern { X-[comp:cleft]->Y; Y[upos=SCONJ]; e: Y->Z; Z[!Filler] }

  commands {
    Z.Filler=mod
  }
}
```

```
rule r {
  pattern { X-[comp:cleft]->Z; Y[upos=PRON]; e: Z->Y; Z[!Filler] }

  commands {
    Z.Filler="subj"
  }
}
```

### **Concaténation des nombres à virgule en un seul token :**

```
rule r {
  pattern { X1[NumType=Card, mwehead]; X2[form=",", pred=y];
  X3[NumType=Card, pred=y]; X1<X2; X2<X3 }
  commands {
```

```
    add_node N:> X1;  
    shift X1 ==> N;  
    append_feats X3 ==> N;
```

```
    N.mwehead = X1.mwehead;  
    N.upos = NUM;  
    N.lemma = X1.lemma + X2.lemma + X3.lemma;  
    N.form = X1.form + X2.form + X3.form;  
    N.textform = X1.form + X2.form + X3.form;
```

```
    del_feat N.pred;  
    del_node X1;  
    del_node X2;  
    del_node X3;  
}  
}
```