# Enrichissements ISTEX



## REFBIBS

## Balisage de références bibligraphiques avec grobid pour la base ISTEX

romain.loth@iscpif.fr

(avec l'aide de P. Lopez - INRIA)

# Plan de la présentation

1) **Grobid**

2) **Formats des refbibs dans ISTEX**

3) **Prototypes refbibs-stack pour clarifier les interactions entre grobid et l'API**

4) **Les 3 modèles obtenus après entraînement sur nos données**

5) **L'entraînement concrètement**

   - Manuellement

   - Avec bako

6) **Cycles d'améliorations qualité**

7) **Remarques sur l'utilisation et le bon entraînement des CRF**

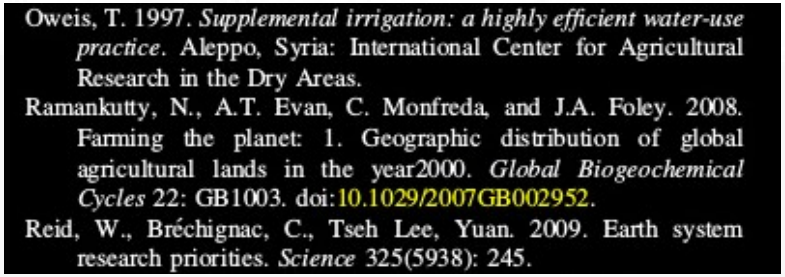
# 1) Présentation de grobid

- **Un outil java**
  - `mvn -Dmaven.test.skip=true clean install`
  - `Qui utilise une librairie CRF : wapiti`
- **Ré-entraîné chez nous**
  - Dépôt git d'origine de Patrice Lopez :
    - https://github.com/kermitt2/grobid
    - Y ajouter nos modèles sous grobid-home/models
  - Dépôt avec nos modèles
    - https://github.com/rloth/grobid
    - N'inclut plus les derniers commits de P. L.

# 1) Installation de grobid

- **Documentation officielle**
  - http://grobid.readthedocs.org/en/latest/
  - Pour clarifier les 3 lancements possibles
    - balisage via serveur REST
      **vp-istex-grobid.intra.inist.fr**
    - balisage via ligne de commande
    - apprentissage
- **Mon installateur**
  **https://git.istex.fr/loth/refbibs-stack/blob/master/bib-install-vp/install_grobid.sh**
  - Permet de configurer maven
  - Met la variable GROBID_HOME dans .bashrc

# 2) Enrichir les articles non-structurés en s'entraînant sur le structuré



Oweis, T. 1997. *Supplemental irrigation: a highly efficient water-use practice*. Aleppo, Syria: International Center for Agricultural Research in the Dry Areas.

Ramankutty, N., A.T. Evan, C. Monfreda, and J.A. Foley. 2008. Farming the planet: 1. Geographic distribution of global agricultural lands in the year2000. *Global Biogeochemical Cycles* 22: GB1003. doi:10.1029/2007GB002952.

Reid, W., Bréchignac, C., Tseh Lee, Yuan. 2009. Earth system research priorities. *Science* 325(5938): 245.

- **LES REFBIBS DANS ISTEX c'est :**
- **Structurer les citations biblio. dans les textes aux notices pauvres**
  - données non-structurées PDF ==> analyse ==> balisage ==> données XML
    - **Avec leurs erreurs OCR, leur variantes stylistiques et historiques**
    - **Cf exemplier (http://wiki.istex.fr/_media/enrichissement/pdf/exemples_pdfs.zip)**
    - **Cf aussi le wiki http://wiki.istex.fr/enrichissement/pdf**
  - objectifs à terme : rendre les références de fin d'article cliquables

    \+ critère ajouté à la recherche documentaire

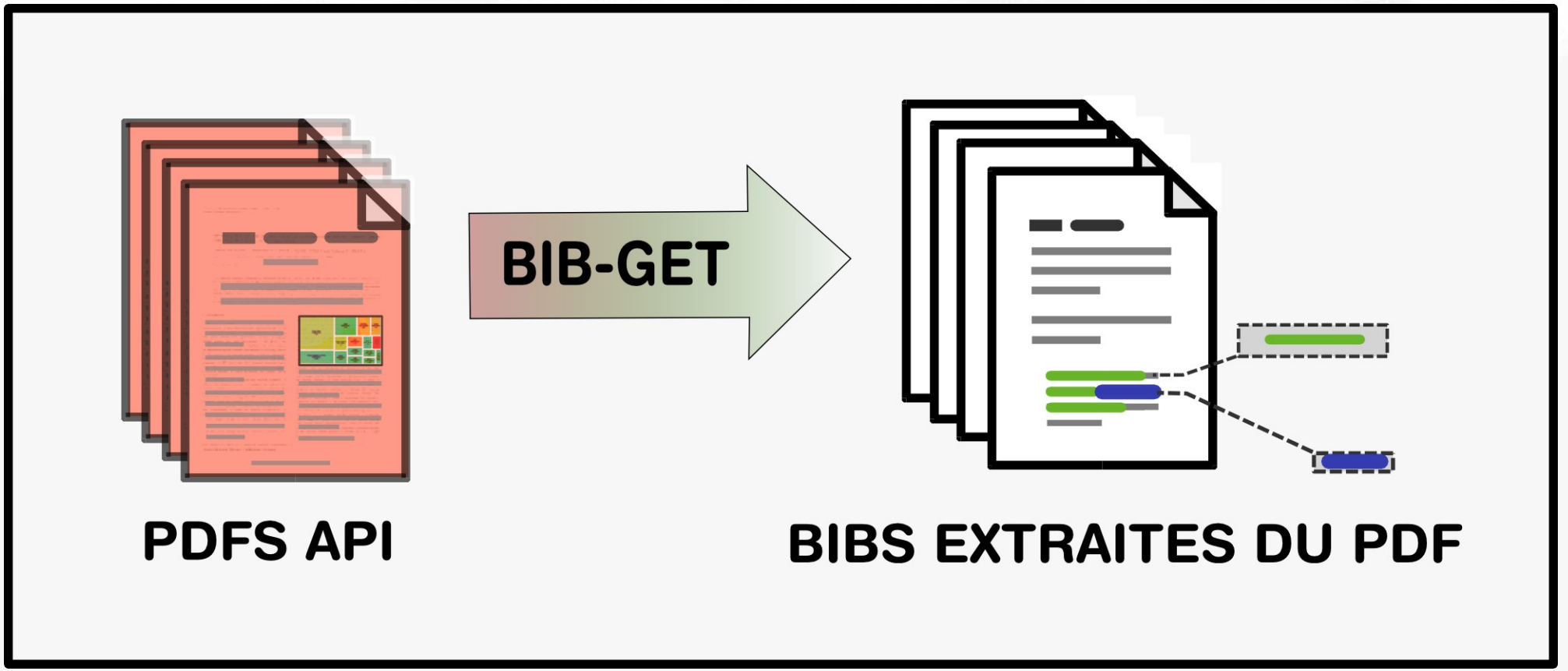
    \+ liens au sein de notre base et en dehors
  - index = « archéologie » des réseaux scientifiques
- **En utilisant la robustesse de l'apprentissage sur données (machine learning)**
  - améliorer les baliseurs automatiques
    - 5 millions de documents déjà +/- bien annotés ISTEX
  - un A/R constant entre tests d'ensembles et gestion cas particuliers
- **Permet de palier plusieurs types de problèmes**

# 2) S'y retrouver dans les formats

- **On veut annoter les citations dans tout type de texte « brut »**
  - les intégrer aux index, à la navigation et à la recherche textuelle
  - les fournir aux observatoires (bibliométrie, veille thématique)
  - les fournir aux analystes (classif. documentaire, termino., th. des graphes)
- **formats de stockage**
  - verbatim visuel = .pdf
  - texte brut = .raw|.txt
  - xmls natifs (nombreuses dtd)
- **formats projet**
  - format TEI principal (évaluation et sorties)
  - formats pseudo-TEI plats (*entraînements*)
- **formats de mise en ligne**
  - métadonnées (liens, tags)
  - données (texte organisé/segmenté)

```
<biblStruct>
    <analytic>
        <title level="a" type="main">DDT and PCB residues in
        airborne fallout and animals in Iceland</title>
        <author>
            <persName>
                <forename type="first">S</forename>
                <forename type="middle">A</forename>
                <surname>Bengtson</surname>
            </persName>
        </author>
        <author>
            <persName>
                <forename type="first">A</forename>
                <surname>Sodergren</surname>
            </persName>
        </author>
    </analytic>
    <monogr>
        <title level="j">Ambio</title>
        <imprint>
            <biblScope unit="volume">3</biblScope>
            <biblScope unit="page" from="84" to="86" />
            <date type="published" when="1974" />
        </imprint>
    </monogr>
</biblStruct>
```

# 2) Le coeur de la tâche refbibs

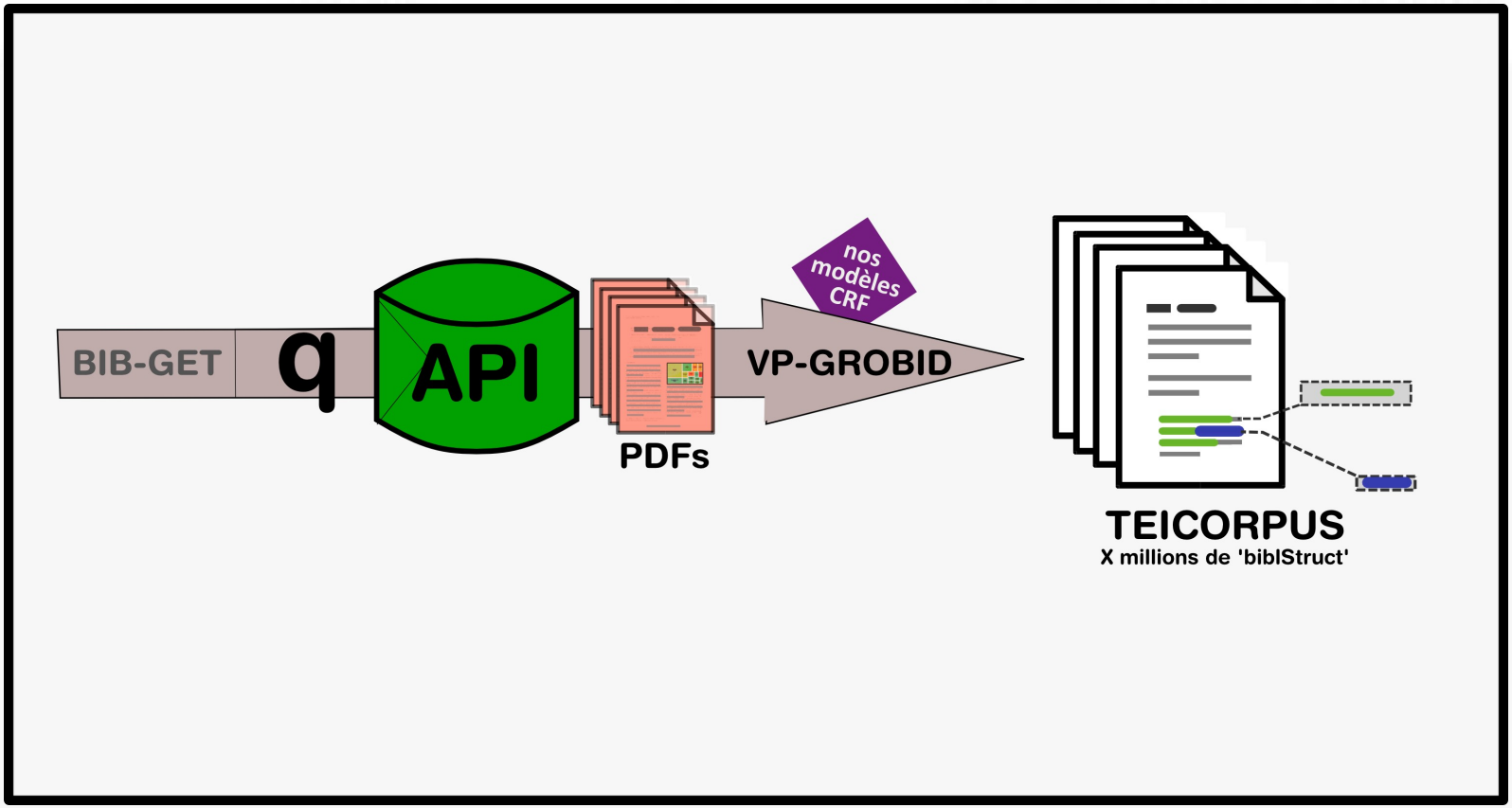



- **Utilise grobide en mode service REST**
  - Client minimal : `curl -v -include --form input=@./thefile.pdf vp-istex-grobid.intra.inist.fr:8080/processReferences`
- **Un client grobid pour vos tests : bib-get.py**
  - **Par exemple bib-get.py -q 'corpusName:springer'**
  - **Sortie XML-TEI**

# 2) Autour de la tâche refbibs

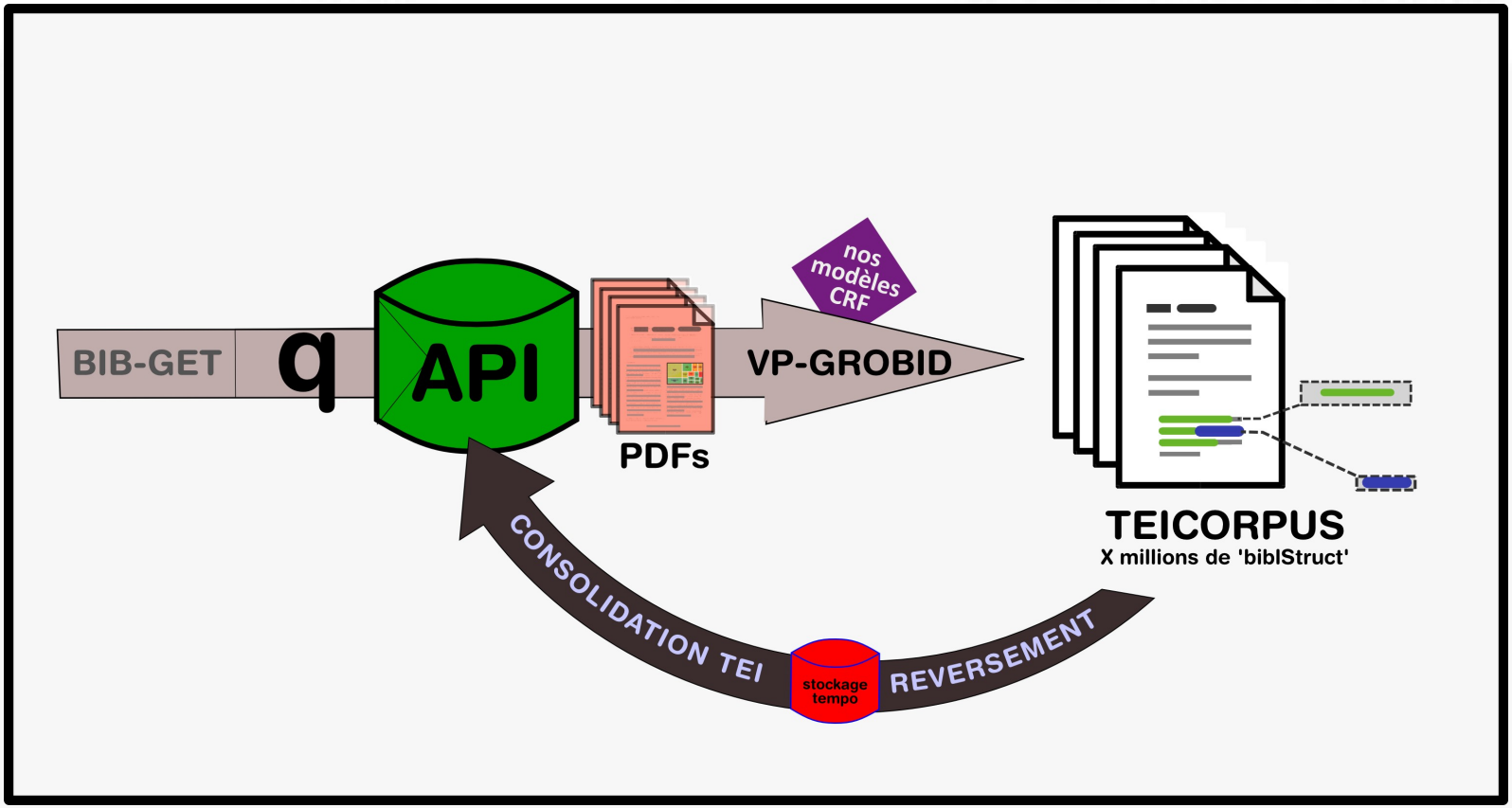
- **Banc d'essai début 2014**
  - choix parmi l'état de l'art
    - Expressions régulières | **CRF**
    - Bilbo | Cermine | **Grobid**
- **Qualité hiver 2014 - 2015 : amélioration des sorties**
  - préparateur de données d'entraînement à partir des bases
  - cycles : corpus supplémentaire => test => modèle
  - améliorations sur la segmentation : **+ 11 %** de rappel
  - améliorations sur la refseg : **+ 4 %** de rappel
- **Quantité : « essai transformé » 2015**
  - Janvier 2015 : première montée en charge sur **2,4 M docs**
  - mise en place du serveur grobid en REST
  - ajout d'une interface pour aider à l'entraînement
  - intégration par l'équipe API à la chaîne loadIstex

# 3) Pour infos: interactions bib-get <=> API




- interaction INPUT avec ISTEX-API (query => PDF)
- client vis-à-vis de la vp-grobid-service
- qui intègre elle-même nos modèles CRF préparés l'hiver dernier

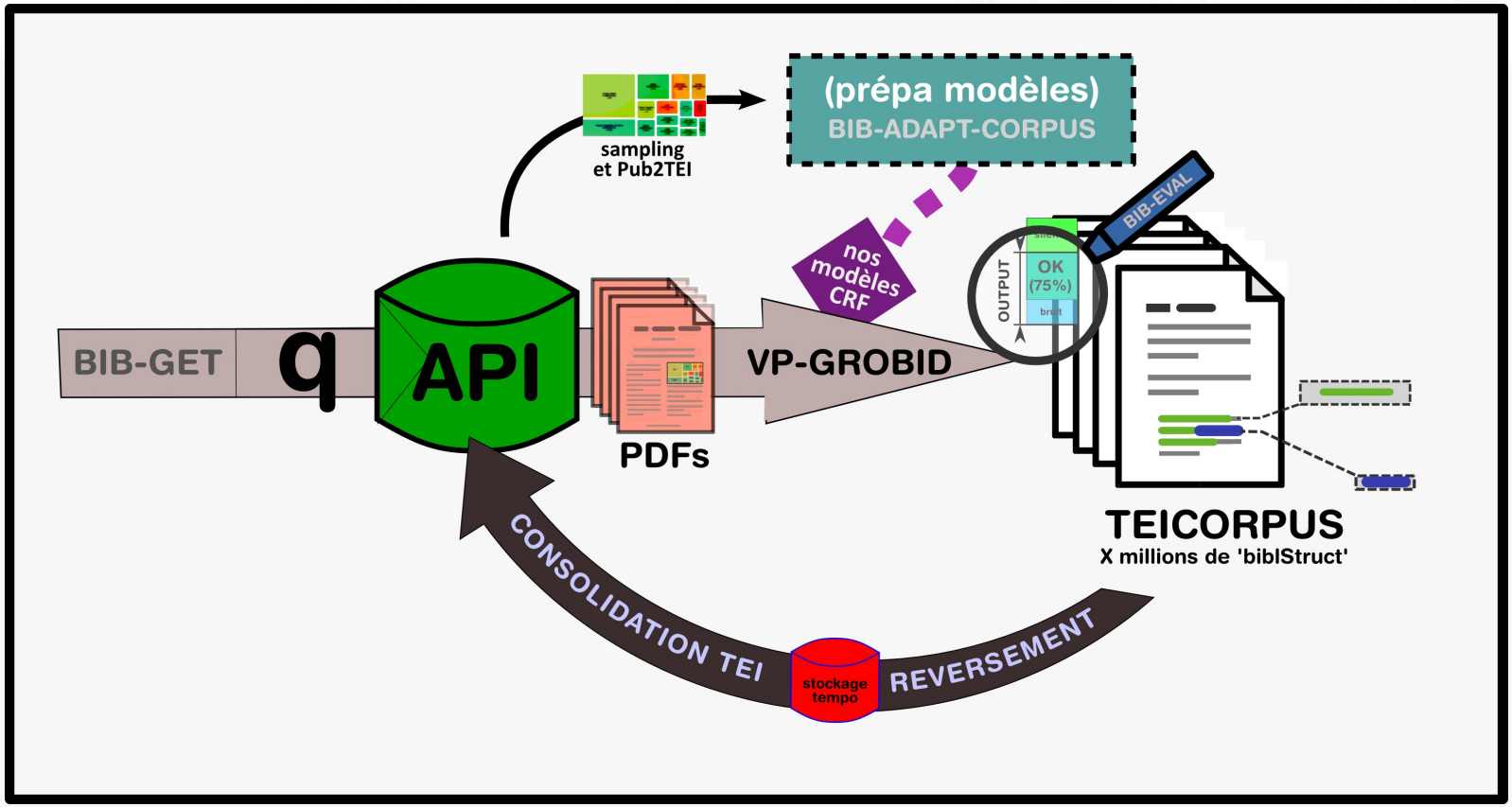
# 3) Pour infos: interactions bib-get <=> API




- SPRINT #28 de l'équipe ISTEX-API
- reversement depuis le format teiCorpus
- client vis-à-vis de la vp-grobid-service

MERCI L'API !!!

# 3) fin packaging bib-adapt-corpus




- **Pérennisation du processus d'entraînement**
  - refactoring des outils de prépa utilisés en R&D vers un module BIB-ADAPT-CORPUS (aka bako)
  - scripts pour un mécanisme d'apprentissage assisté

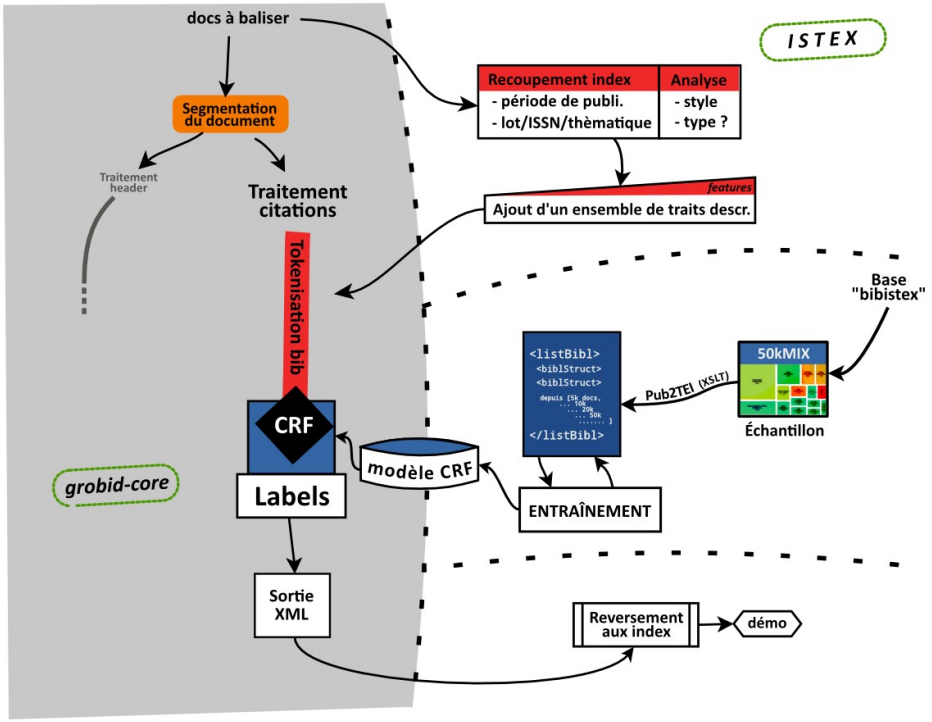
# 3) Récap: refbibs-stack se veut comme une suite d'applis intégrées

- **Vous pouvez utiliser grobid directement**
- **Mais il y a aussi mes scripts qui devraient guider le travail**
  - Pour la chaîne de production
    - *bib-install-vp*
      - pour installer grobid-service  sur un serveur de production
      - fork: une version adaptée à nos besoins.
    - *bib-get*
      - pour obtenir les bibs structurées depuis les docs API
    - *bib-findout-api*
      - pour les ré-identifier dans des bases
  - Pour des nouveaux "profils de documents"
    - *bib-adapt-corpus* pour entraîner grobid
- **J'ai regroupé les autre modules upstream en 1 package**
  - un seul dépôt à télécharger

    https://git.istex.fr/loth/refbibs-stack

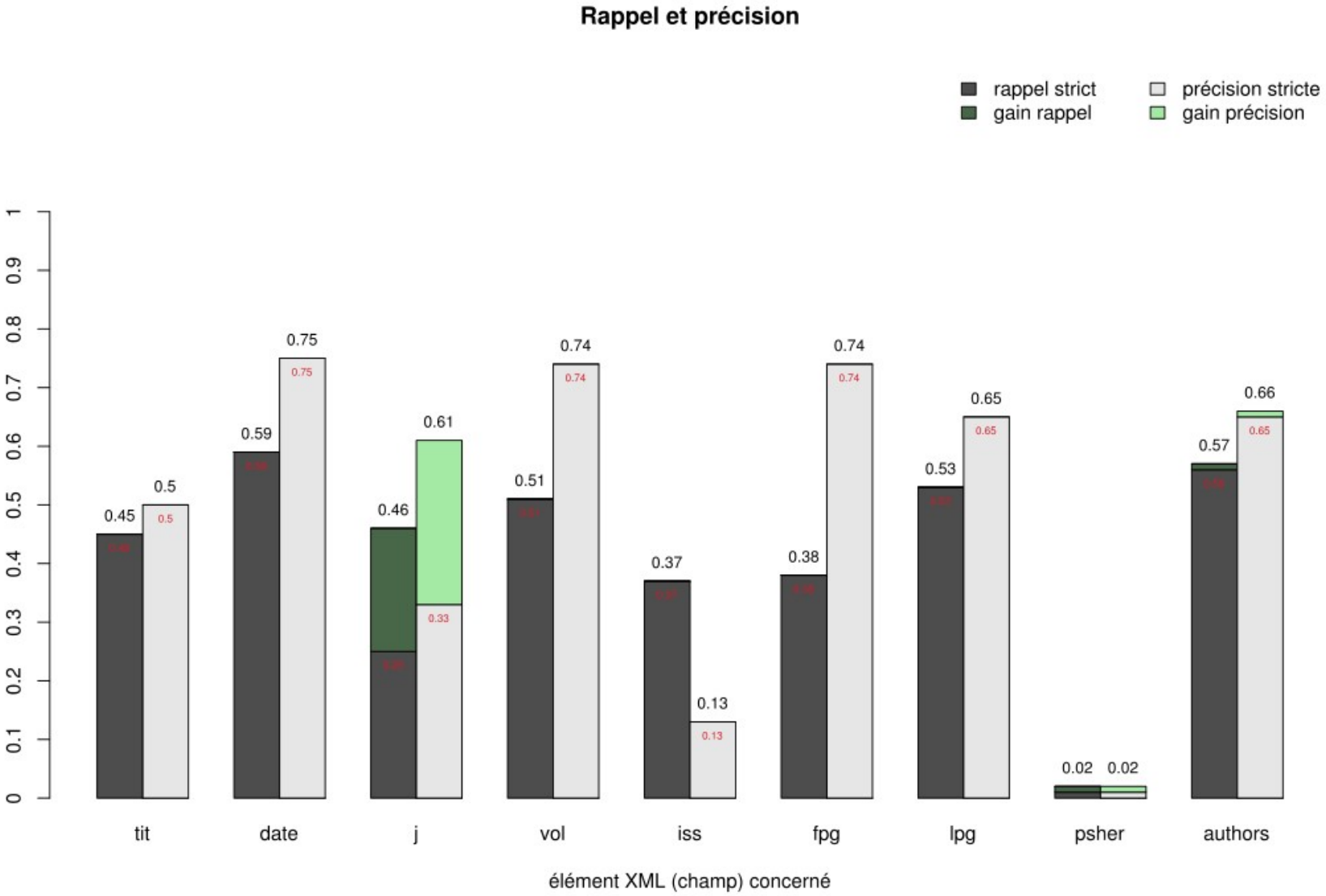
# 4) Nos modèles : la piste suivie

- **«réntraînement sur nos données (« piste bleue »).**
  - bénéficie du fait que la moitié des corpus contiennent d natifs avec les refbibs déjà structurées.
  - on peut s'entraîner dessus pour mieux traiter les autres.
  - effectué partiellement hiver 2014
    - obtention de 3 modèles améliorés (leur nom selon éc source)
      - segmentation : **seg-grosto-478**
        - https://github.com/rloth/grobid/blob/master/grobid-home/models/segmentation/model.crf?raw=true
      - reference-segmentation : **refseg-lastcor-100**
        - https://github.com/rloth/grobid/blob/master/grobid-home/models/reference-segmenter/model.crf?raw=true
      - citations : **cit-111-92**
        - https://github.com/rloth/grobid/blob/master/grobid-home/models/citation/model.crf?raw=true
  - **CES 3 LIENS SONT IMPORTANTS**
  - CE SONT CES 3 MODELES QUI PERMETTENT D'AVOIR UN BONNE QUALITE DE RESULTATS DANS ISTEX
  - Le nom des modèles est lié à l'échantillon source utilisé
  - Tableau de suivi
    https://git.istex.fr/loth/refbibs-stack/blob/master/bib-adapt-corpus/rapport_training_initial.ods?raw=true

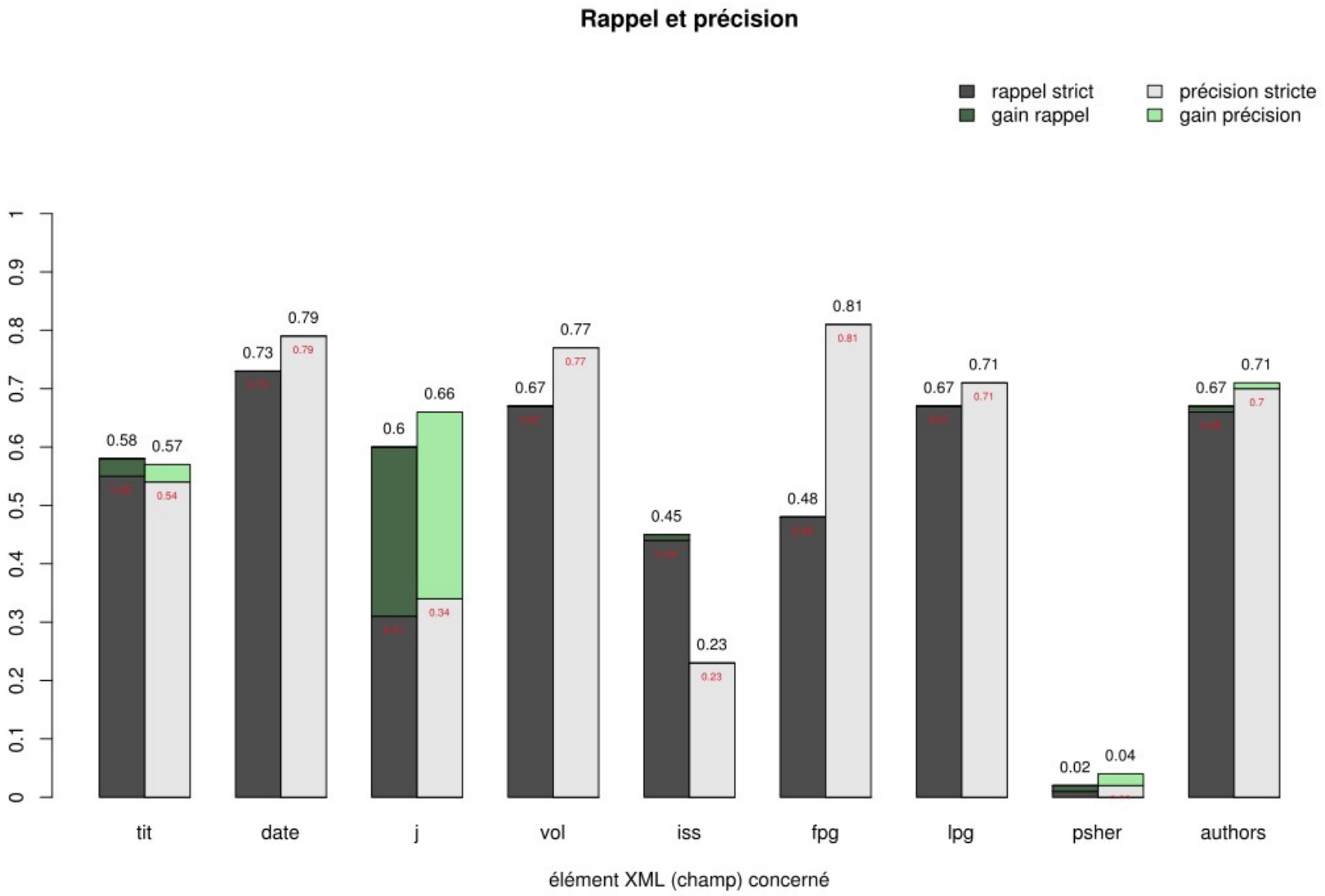



- Une autre piste d'améliorations était envisagée : améliorations sur les features CRF (ajout de traits descriptifs utiles à la prédiction) (« piste rouge ») .
  - piste encore non travaillée

# 4) Résultats par champs sans nos modèles



# 4) Nos modèles : Résultats par champs



# 5) Utilisation des CRF dans Grobid

- **11 modèles CRF « en cascade »**
  - On en utilise 5 pour les refbibs

**1) Déterminer la zone des refbibs**

**modèle « segmentation »**

2) déterminer pour chaque ligne si c'est une nouvelle refbib

modèle « **reference-segmenter** »

3) déterminer dans chaque refbib les champs majeurs

modèle « **citation** »

4) déterminer dans le champ auteurs les prénoms et noms

modèle « **name** »

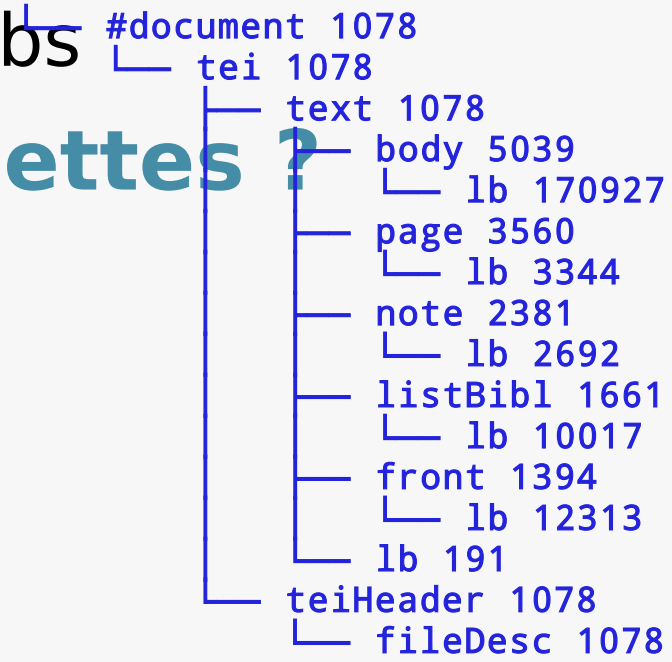
- **Chaque modèle a besoin de 3 éléments**
  - une séquence d'étiquettes ad hoc
    - pseudo-TEI
  - un flux textuel observable aligné sur cette séquence
  - une template pour décrire le flux en terme de features

# 5) L'entraînement grobid sans bib-adapt

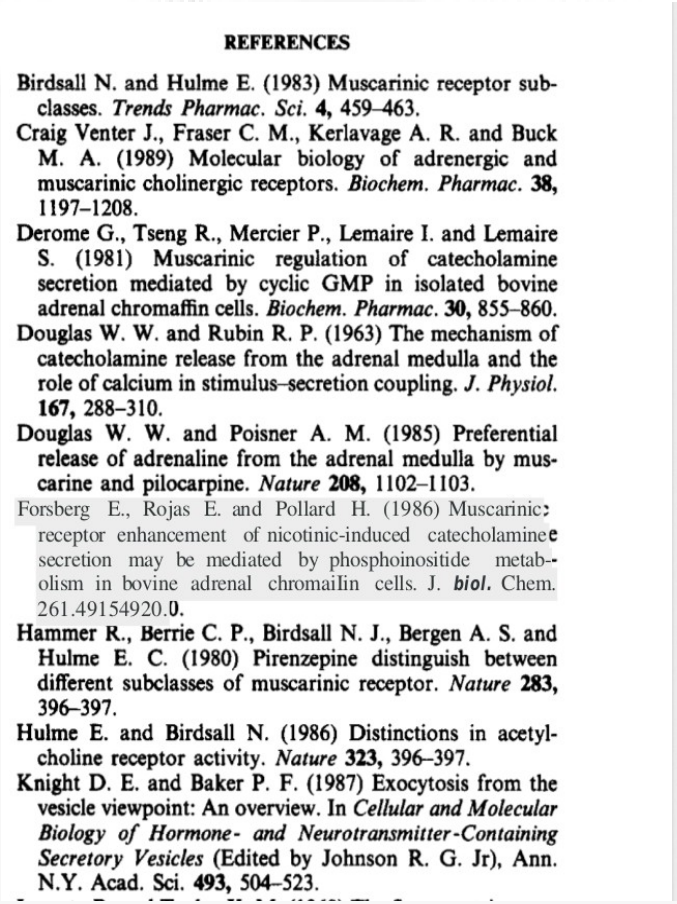
- **grobid-core createTraining**
  - (preparation des formats pour entraîner une étape n)
    - Étape segmentation : les instances à baliser sont des lignes
    - Etapes suivantes : les instances à baliser sont les mots
  - Prend les PDF, passe les n-1 premières étapes du process habituel
    - raw : pour avoir les flux textes qui seraient vus par l'étape n (juste avant l'étape)
      - Pas juste sous forme .txt mais plutôt avec une instance et les colonnes
    - tei : pour créer des teis « locaux » avec juste les résultats de l'étape n (juste après)
      - Dans l'idéal ces tei doivent être corrigés à la main pour qu'ils servent de gold
      - Ou automatiquement corrigés si possible (par exemple ré-intégration de balises issues des XML éditeurs annotés)
  - Par exemple passe la segmentation pour entraîner la ref-segmentation
  - **Exemple :**
  - `java -Xmx1024m -jar grobid-core/target/grobid-core-0.3.3-SNAPSHOT.one-jar.jar -gH grobid-home -gP grobid-home/config/grobid.properties -dIn ~/refbib/corpus/mon_corpus_train_cit/les_pdfs_d_origine/ -dOut ~/refbib/corpus/mon_corpus_train_cit/createTrainingFulltext_out/ -exe createTrainingFulltext`
- **grobid-trainer generate-resources**
  - **(entrainement)**
  - **Le coeur de l'algorithme (WAPITI CRF) permettant l'entraînement :**
    - **Prend en entrée un dossier grobid-trainer/resources/dataset/<NOM_DU_MODELE>**
    - **Tourne de 5 à 30 h (« boîte noire »: crée des milliers de fonctions (features => balise) et ajuste leur importance)**
    - **Sort un fichier .crf enregistré directement dans grobid-home/models/<NOM_DU_MODELE>**
  - **Exemple :**
  - `mvn generate-resources -P train_reference-segmentation`
  - `Cf. aussi scripts bash utilisés avant le développement de bako : https://git.istex.fr/loth/refbibs-stack/tree/7e2501a/train )`
- **grobid-core processReferences**
  - On lance une annotation simple avec le nouveau modèle => pour évaluer

# Exemple : le modèle segmentation

- **Objectif du modèle :**
  - Déterminer la zone des refbibs
- **Quelle séquence d'étiquettes ?**
  - pseudo-TEI
    - teiHeader
    - front
    - body
    - listBibl
    - notes
  - + les sauts de ligne <lb/>
  - le flux textuel est celui issu du « ragréage »



# Préparation de corpus d'entraînement



REFERENCES

Birdsall N. and Hulme E. (1983) Muscarinic receptor subclasses. *Trends Pharmac. Sci.* **4,** 459–463.

Craig Venter J., Fraser C. M., Kerlavage A. R. and Buck M. A. (1989) Molecular biology of adrenergic and muscarinic cholinergic receptors. *Biochem. Pharmac.* **38,** 1197–1208.

Derome G., Tseng R., Mercier P., Lemaire I. and Lemaire S. (1981) Muscarinic regulation of catecholamine secretion mediated by cyclic GMP in isolated bovine adrenal chromaffin cells. *Biochem. Pharmac.* **30,** 855–860.

Douglas W. W. and Rubin R. P. (1963) The mechanism of catecholamine release from the adrenal medulla and the role of calcium in stimulus–secretion coupling. *J. Physiol.* **167,** 288–310.

Douglas W. W. and Poisner A. M. (1985) Preferential release of adrenaline from the adrenal medulla by muscarine and pilocarpine. *Nature* **208,** 1102–1103.

Forsberg E., Rojas E. and Pollard H. (1986) Muscarinic receptor enhancement of nicotinic-induced catecholamine secretion may be mediated by phosphoinositide metabolism in bovine adrenal chromaiIin cells. J. *biol.* Chem. 261.49154920.0.

Hammer R., Berrie C. P., Birdsall N. J., Bergen A. S. and Hulme E. C. (1980) Pirenzepine distinguish between different subclasses of muscarinic receptor. *Nature* **283,** 396–397.

Hulme E. and Birdsall N. (1986) Distinctions in acetylcholine receptor activity. *Nature* **323,** 396–397.

Knight D. E. and Baker P. F. (1987) Exocytosis from the vesicle viewpoint: An overview. In *Cellular and Molecular Biology of Hormone- and Neurotransmitter-Containing Secretory Vesicles* (Edited by Johnson R. G. Jr), Ann. N.Y. Acad. Sci. **493,** 504–523.

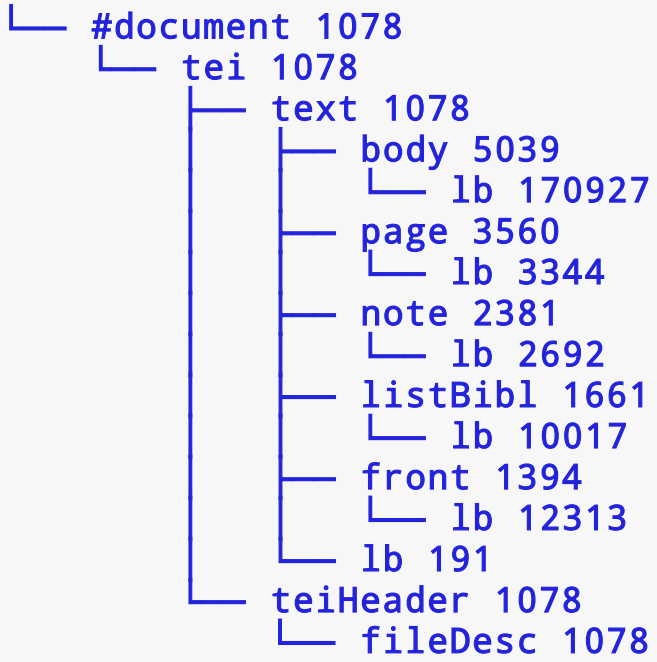
```
<tei>
<teiHeader>
    <fileDesc xml:id="RAG-03064492_v100i3_074284139190029S-check"/>
</teiHeader>
<text xml:lang="en">
    <listBibl>
REFERENCES <lb/>
<lb/>
<lb/>
<bibl>classes. TrenL Pharmac. Sci. 4, 459463. <lb/>
<lb/>
Birdsall N. and Hulme E. (1983) Muscarinic receptor sub-<lb/></bibl>
<bibl>Craig Venter J., Fraser C. M., Kerlavage A. R. and Buck <lb/>M. A.
(1989) Molecular biology of adrenergic and <lb/>muscarinic choline@
receptors. Biochem. Pharmac. 38, <lb/><lb/>1197-1208. <lb/></bibl>
<lb/>
<bibl>Derome G., Tseng R., Mercier P., Lemaire I. and Lemaire <lb/>S. (198
1) Muscarinic regulation of catecholamine <lb/>secretion mediated by cyclic
GMP in isolated bovine <lb/>adrenal chromaffin cells. biochem. Pharmac.
30,855-860. <lb/></bibl>
<bibl>Do&amp;as W. W. and Rubin R. P. (1963) The mechanism of <lb/>
carecholamine release from the-adrenal medulla and the <lb/>role of calcium
in stimulus-secretion coupling. J. Physiol. <lb/><lb/>167, 288-310. <lb/>
</bibl>
<lb/>
<bibl>Douglas W. W. and Poisner A. M. (1985) Preferential <lb/>release of
adrenaline from the adrenal medulla by mus-<lb/>carine and pilocarpine.
Nature 208, 1102-I 103. <lb/></bibl>
<bibl>Forsberg E., Rojas E. and Pollard H. (1986) Muscarinic <lb/>receptor
enhancement of nicotinic-induced catecholamine <lb/>secretion may be
mediated by phosphoinositide metab-<lb/>olism in bovine adrenal chromaiIin
cells. J. biol. Chem. <lb/>261.49154920. <lb/></bibl>
<bibl>Hammer R., Berrie C. P., Birdsall N. J., Bergen A. S. and <lb/>Hulme
E. C. (1980) Pirenzepine distinguish between <lb/>different subclasses of
muscarinic receptor. Nature 283, <lb/>396-397. <lb/></bibl>
<bibl>Hulme E. and Birdsall N. (1986) Distinctions in acetyl-<lb/>choline
receptor activity. katur; 323, 396-397. <lb/></bibl>
<lb/>
<lb/>
<lb/>
<bibl>Kniaht D. E. and Baker P. F. (1987) Exocytosis from the <lb/>vesicle
viewpoint: An overview. In Cellular and Molecular <lb/><lb/><lb/>Biology of
Hormone-and Neurotransmitter-Containing <lb/>Secretory Vesicles (Edited by
Johnson R. G. Jr), Ann. <lb/>N.Y. Acad. Sci. 493, 504-523. <lb/></bibl>
```

- **Méthode manuelle => utile pour le modèle segmentation**

  (peu de corpus nécessaire pour généraliser)

# Les features du modèle segmentation

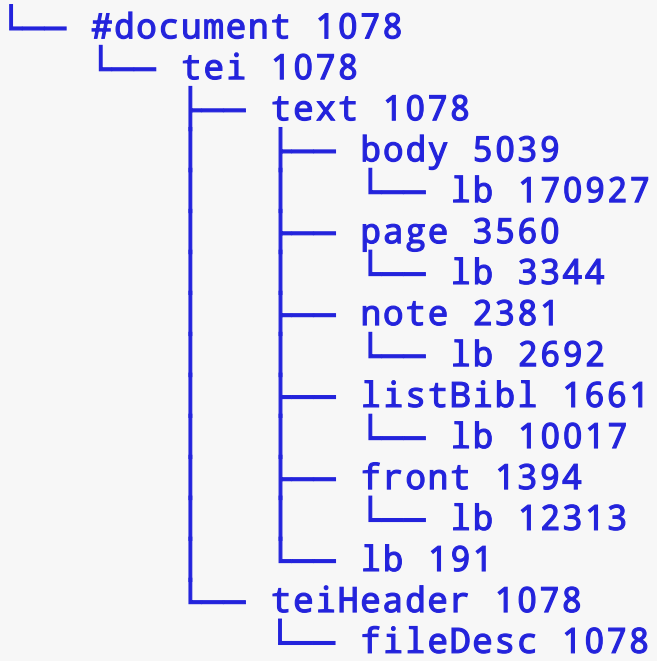
- **Objectif du modèle :**
  - Déterminer la zone des refbibs
- **Templates de features utilisées**
  - # Lowercase token 1
  - # Prefix 1-4 characters (2-5)
  - # Suffix 1-4 characters (6-9)
  - # Block info (10)
  - # Line info (11)
  - # page info (12)
  - # Font info (13-14)
  - # Bold info (15)
  - # Italic info (16)
  - # Capitalization (17)
  - # Digits (18)
  - # Char (19)
  - # Dict info (20-26)
  - # Punctuation (27)
  - # relative document position (28)
  - # relative page position (29)
  - # Output




- **Ces features apparaissent dans les fichiers corpus/raw de grobid-trainer/resources/dataset**
- **Pour ce qu'est une feature cf. en fin de présentation les remarques théoriques sur les CRF**

# Les features du modèle segmentation

- **Objectif du modèle :**
  - Déterminer la zone des refbibs
- **Templates de features utilisées**
  - # Lowercase token 1
  - # Prefix 1-4 characters (2-5)
  - # Suffix 1-4 characters (6-9)
  - # Block info (10)
  - # Line info (11)
  - # page info (12)
  - # Font info (13-14)
  - # Bold info (15)
  - # Italic info (16)
  - # Capitalization (17)
  - # Digits (18)
  - # Char (19)
  - # Dict info (20-26)
  - # Punctuation (27)
  - # relative document position (28)
  - # relative page position (29)
  - # Output




- **Ces features apparaissent dans les fichiers corpus/raw de grobid-trainer/resources/dataset**
- **Pour ce qu'est une feature cf. en fin de présentation les remarques théoriques sur les CRF**

# 5) Packaging bib-adapt-corpus : commandes

- **bako new_workshop**
  - récupère les modèles courants
  - crée un corpus d'évaluation
- **bako make_set**
  - prépare des corpus (import ou sampling direct API)
- **bako make_trainers**
  - prépare les fichiers d'entraînement pour chaque doc
    - A) Obtention du flux brut tel qu'il est vu par grobid
    - B) Récupération des balises des XML natifs converties en TEI via Pub2TEI
    - Création des fichiers TEI d'entraînement en copiant ces balises sur le flux (A)
- **bako make_training**
  - lance la génération du modèle CRF

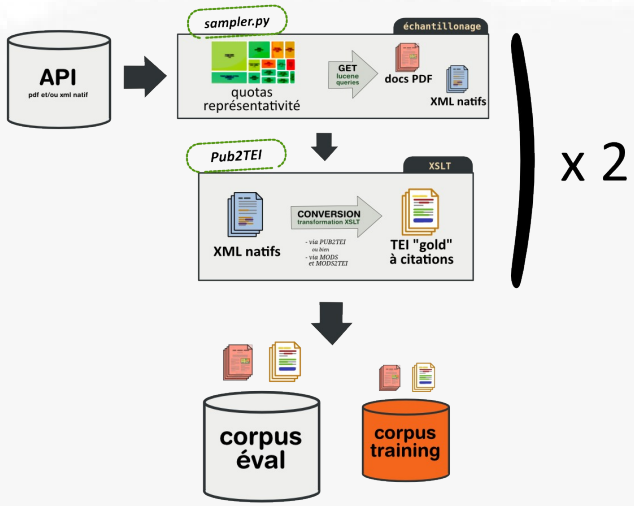
    /!\ entraîner un CRF c'est souvent plusieurs heures
- **bako make_eval**
- **bako modelstore**
  - Les modèles sont rangés dans un seul dossier

# 5) Packaging bib-adapt-corpus : détails

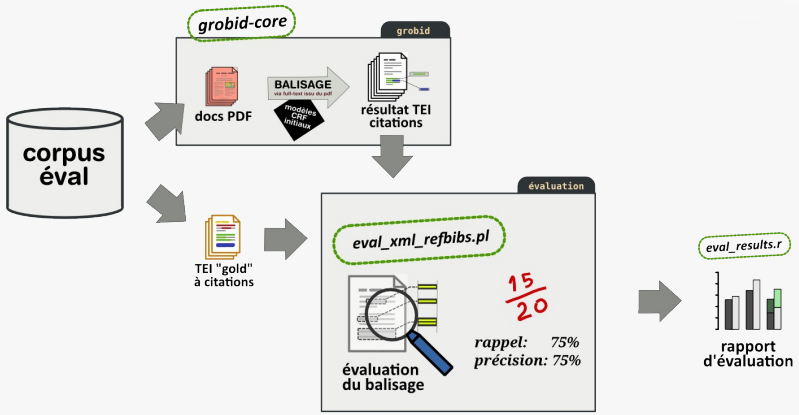
- **Entrée: 7 types de formats XML natifs supportés**
  - **elsevier, wiley, nature, rsc, iop, oup, springer**
  - ajout DTD, transformation natif => TEI gold => reports flux => grobid-training
  - ON CREE DU CORPUS D'ENTRAINEMENT A PARTIR DU STRUCTURE EXISTANT
- **2 grandes librairies internes : libconsulte et libtrainers**
  - libconsulte: api.py /sampler.py / corpusdirs.py
    - Pour aider à faire des échantillons représentatifs, récupérer et stocker les corpus
  - libtrainers: grobid_corpusdirs.py / grobid_models.py
    - un model-store interfacé avec le coeur du traitement
      - grobid-core createTraining (preparation)
      - **grobid-trainer generate-resources (entrainement)** **<-----** **(WAPITI CRF)**
      - grobid-core processReferences (annotation)
- **Utilisation en théorie**
  - nouveaux corpus d'entraînement + anciens => le balisage devient meilleur
  - sortie des modèles entrainés + évaluation de gain par modèle
  - obtention modèle mieux adapté : intégrable dans bib-get en balisage
- **Pour plus d'infos il y a un README sur le git bib-adapt-corpus**

# 5) Packaging bib-adapt-corpus : schéma

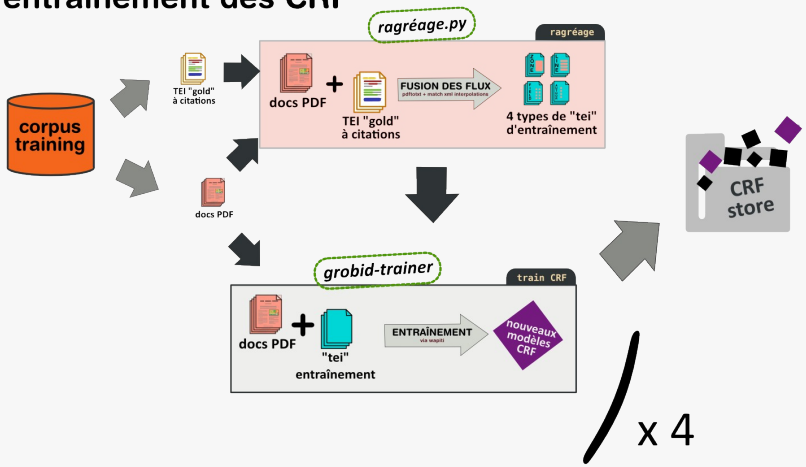
étape 1 :
génération de 2 corpus



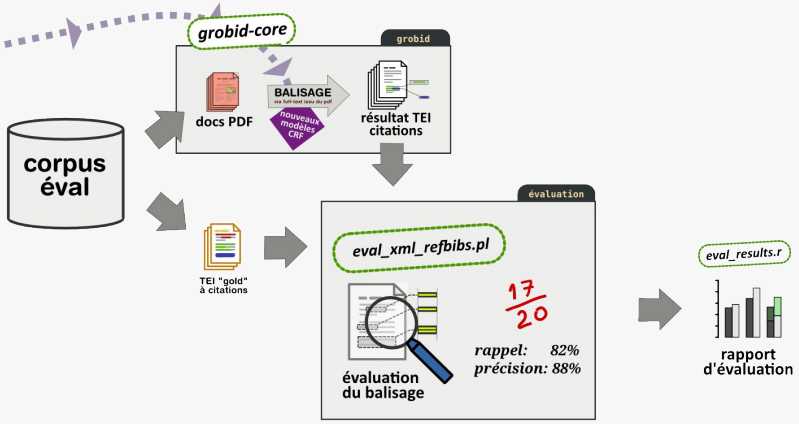

étape 2 :
évaluation initiale (baseline)




étape 3 :
entraînement des CRF

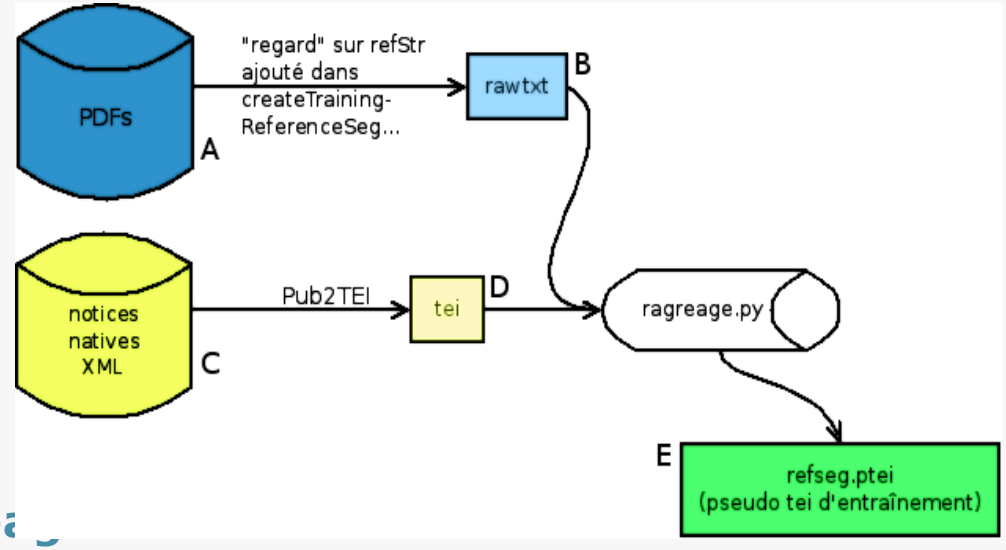



étape 4 :
évaluation des nouveaux modèles CRF



# Préparation de corpus d'entraînement

- **Procédure pour préparer du corpus automatiquement**
  - On utilise nos données déjà annotées (notices riches)
- **Différence entre les formats**
  - Mais les infos des notices ne préservent pas tout :
    - les virgules
    - ni les tirets, les parenthèses,
    - ni les mots d'escorte dédiés
      Vol:
      In:
    - ça paraît pas grand chose...
      - détails typiques fondamentaux
      - pour reconnaître les champs




- **Développement d'un script de « ragréa**
  - Fusion des informations typées avec le texte
  - Le texte tel que Grobid le verrait, avec les annotations telles que les éditeurs nous les ont livrées dans les XML les plus structurés
  - https://git.istex.fr/loth/refbibs-stack/blob/master/bib-adapt-corpus/libtrainers/ragreage.py

# Outils supplémentaires développés

- **Des développements complémentaires**
- **Outils de traitement du document**
  - Echantilloneur
    - https://git.istex.fr/loth/libconsulte (intégré dans bako)
  - « Re-formateuses »
    - bibl (markups plats) <=> biblStruct (markups arborés)
    - de 7 formats « natifs » vers la TEI
      - Formats déjà gérés par Pub2TEI (Laurent Romary et Patrice Lopez)
        - Elsevier, OUP (NLM-medline), RSC, springer
      - Formats ajoutés pour bib-adapt à Pub2TEI
        - Wiley, Nature, IOP
    - https://git.istex.fr/loth/libconsulte/tree/master/etc/Pub2TEI
- **Divers**
  - Afficheuse arbre XML
    - https://git.istex.fr/loth/utilitaires/blob/master/xml_tagstats.pl

# 5) Récap : les points-clefs

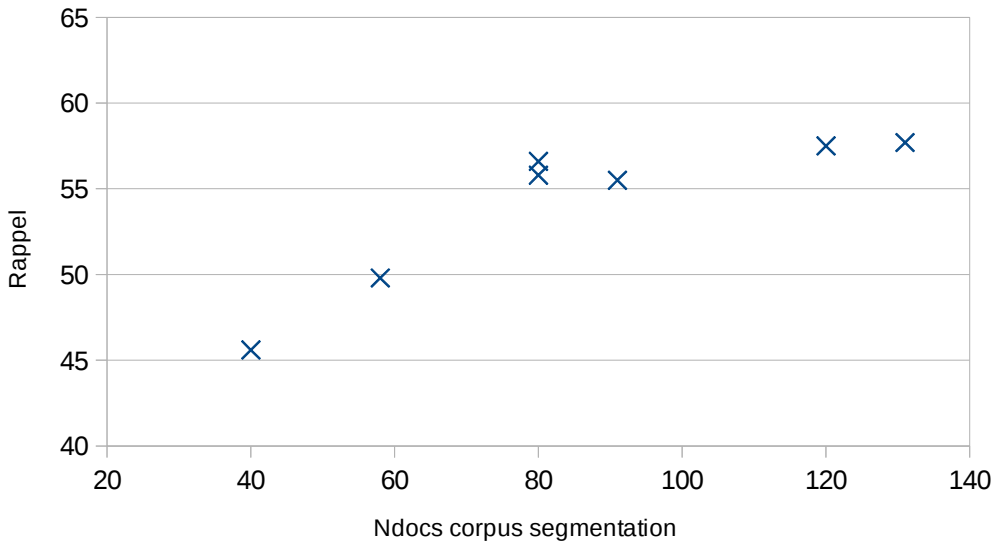
- **Taux de réussite réel et taux théorique**
  - choix d'échantilloner le corpus avec ses défauts
    - selon source
    - selon version PDF
    - selon type de document
    - selon style typographique de la citation
  - fruit d'une analyse approfondie des métadonnées en présence
  - ainsi les taux de succès reflètent l'horizon d'attente face au « tout-venant »
- **Un modèle probabiliste => robustesse à long terme**
  - des modules d'analyse « CRF » en cascade
  - entraînées sur des données réelles connues (« gold »)
- **Résultats en sortie**
  - XML-TEI
    - niveaux composites
    - stockage du texte et de différentes annotations
  - Une « couche » d'enrichissement
    - Pour l'utilisation en production

# Suivi du modèle « segmentation »

| executable (gris = baseline) | Ndocs corpus seg | Rappel |
|---|---|---|
| grobid 0.3.3 | 40 | 45,6 |
| grobid 0.3.3b (= avec seg-b-18PL) | 58 | 49,8 |
| grobid 0.3.3 + seg-a-40 | 80 | 56,6 |
| grobid 0.3.3c (= avec seg-c-40PL) | 80 | 55,8 |
| grobid 0.3.3d (=avec seg-d-51PL) | 91 | 55,5 |
| grobid 0.3.3c (mais avec seg-a+c) | 120 | 57,5 |
| grobid 0.3.3d (=avec seg-a+d) | 131 | 57,7 |

- **« segmentation »**
  **= passage obligé**
- **difficulté typique de la base**
- **explique une partie du silence**

- **modèle retravaillé chez P. Lopez**
  - baisse résultats
  - mais baisse erreurs algo
- **ajout de nouveau corpus manuellement**
- **remontée logarithmique**

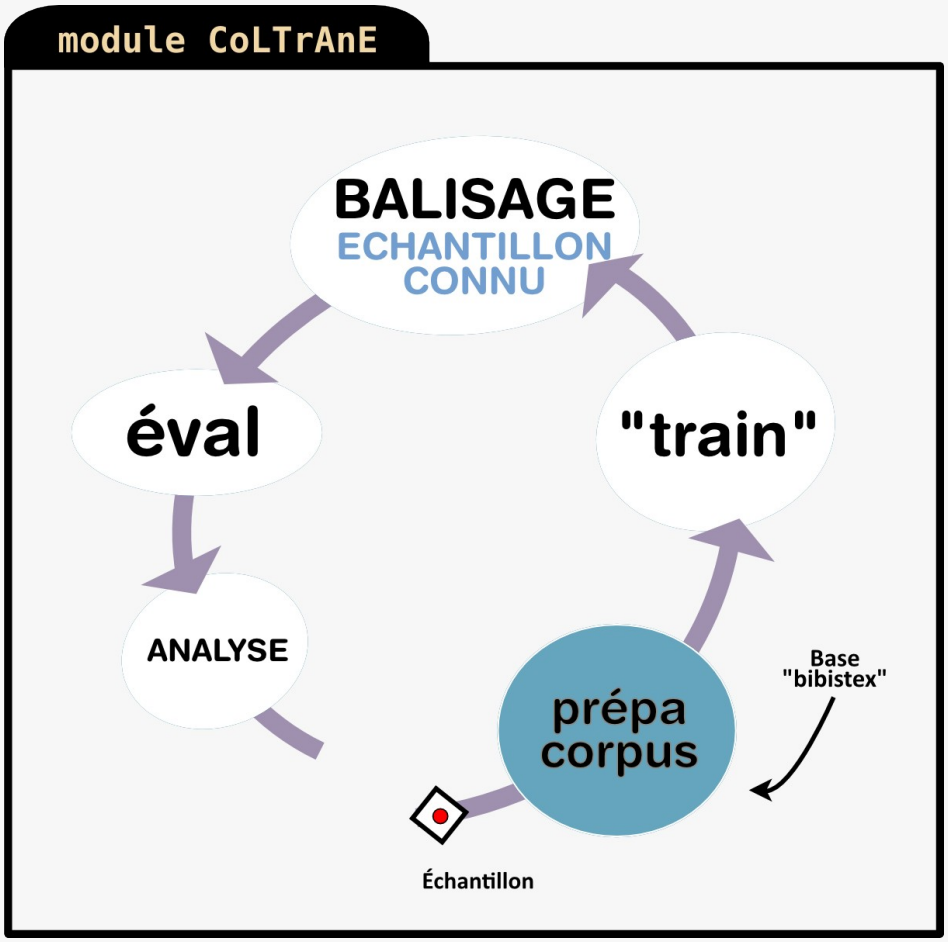


# 6) Méthode des « Cycles » incrémentaux

| | Ndocs corpus seg | Nbibl corpus refseg | eps | Rappel | Précision | taux docs xerr | taux docs vides |
|---|---|---|---|---|---|---|---|
| grobid 0.2.1 branche trunk | 0 | 0 | | 46,2 | 59,2 | 0,41% | 8,89% |
| grobid 0.3.0 branche segm | 36 | 416 | 10-5 | 60,6 | 64 | 4,93% | 3,44% |
| grobid 0.3.0 branche segm | 36 | 416 | 10-5 | 63,3 | 67 | 4,93% | 3,44% |
| grobid 0.3.0 branche segm + seg-a-40 | 76 | 416 | 10-5 | 61 | 67,4 | 3,35% | 2,46% |
| grobid 0.3.3 | 40 | 416 | 10-5 | 45,6 | 68,6 | 4,60% | 21,97% |
| grobid 0.3.3 + seg-a-40 | 80 | 416 | 10-5 | 56,6 | 68,9 | 6,12% | 9,51% |
| grobid 0.3.3b (= avec seg-b-18PL) | 58 | 416 | 10-5 | 49,8 | 69,7 | 4,20% | 21,21% |
| grobid 0.3.3c (= avec seg-c-40PL) | 80 | 416 | 10-5 | 55,8 | 69,4 | 6,03% | 11,00% |
| grobid 0.3.3c (avec seg-a+c) | 120 | 416 | 10-5 | 57,5 | 69,2 | 5,61% | 10,62% |
| grobid 0.3.3c avec seg-a+c et refseg-test-2500 | 120 | 2916 | 10-5 | 32,6 | 54,4 | 5,18% | 10,67% |
| grobid 0.3.3d (=avec seg-a+c) | 120 | 416 | 10-5 | 57,4 | 69,1 | 5,60% | 10,62% |
| grobid 0.3.3d (=avec seg-d-51PL) | 91 | 416 | 10-5 | 55,5 | 69,6 | 5,94% | 13,94% |
| grobid 0.3.3d (=avec seg-a+d) | 131 | 416 | 10-5 | 57,7 | 69,2 | 5,92% | 10,47% |
| grobid 0.3.3d (=avec seg-a+c et refseg-bcor-25) | 120 | 999 | 10-5 | 54,6 | 60,9 | 1,55% | 11,21% |
| grobid 0.3.3d (=avec seg-a+c et refseg-bcor-25 et eps 10 | 120 | 999 | 10-4 | 61,8 | 62,2 | 2,23% | 10,91% |
| grobid 0.3.3d (=avec seg-a+c et refseg-bcor-25 et eps 10 | 120 | 999 | 10-3 | 60,3 | 66,9 | 2,18% | 11,14% |

# 6) Méthode des « Cycles » incrémentaux

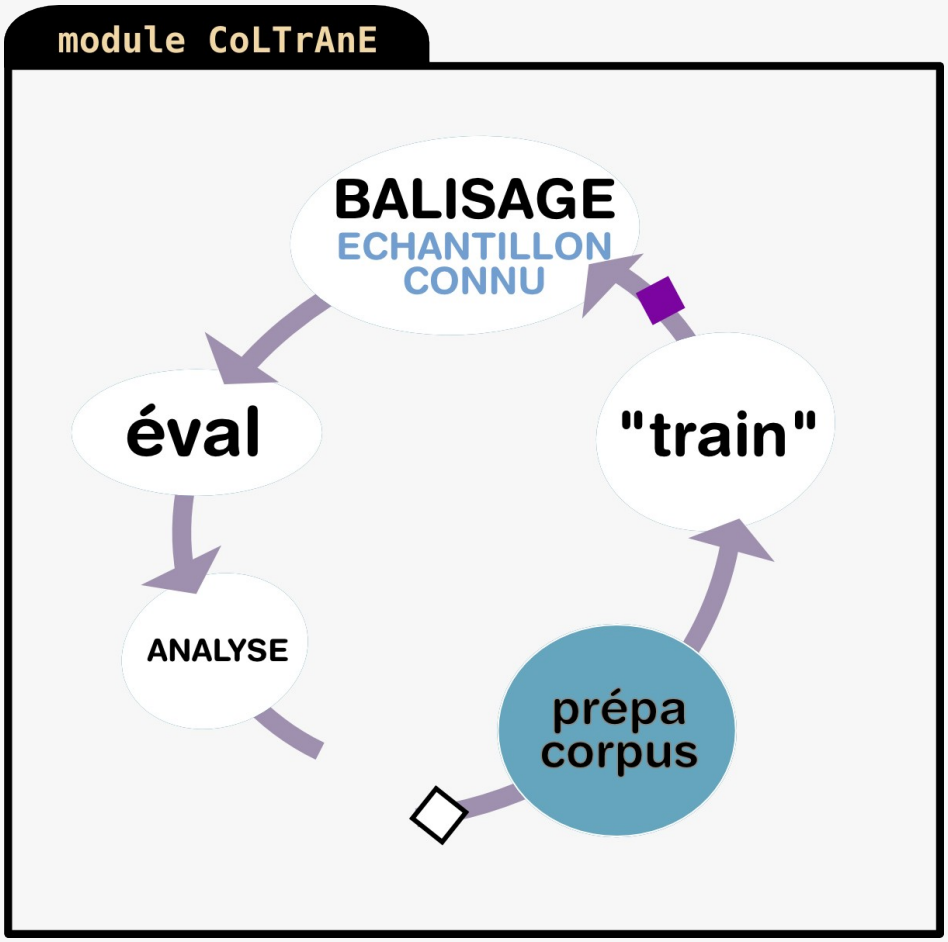
## Cycles entraînement/éval




| | Ndocs corpus seg | Nbibl corpus refseg | eps | Rappel | Précision | taux docs xerr | taux docs vides |
|---|---|---|---|---|---|---|---|
| grobid 0.2.1 branche trunk | 0 | 0 | | 46,2 | 59,2 | 0,41% | 8,89% |
| grobid 0.3.0 branche segm | 36 | 416 | 10-5 | 60,6 | 64 | 4,93% | 3,44% |
| grobid 0.3.0 branche segm | 36 | 416 | 10-5 | 63,3 | 67 | 4,93% | 3,44% |
| grobid 0.3.0 branche segm + seg-a-40 | 76 | 416 | 10-5 | 61 | 67,4 | 3,35% | 2,46% |
| grobid 0.3.3 | 40 | 416 | 10-5 | 45,6 | 68,6 | 4,60% | 21,97% |
| grobid 0.3.3 + seg-a-40 | 80 | 416 | 10-5 | 56,6 | 68,9 | 6,12% | 9,51% |
| grobid 0.3.3b (= avec seg-b-18PL) | 58 | 416 | 10-5 | 49,8 | 69,7 | 4,20% | 21,21% |
| grobid 0.3.3c (= avec seg-c-40PL) | 80 | 416 | 10-5 | 55,8 | 69,4 | 6,03% | 11,00% |
| grobid 0.3.3c (avec seg-a+c) | 120 | 416 | 10-5 | 57,5 | 69,2 | 5,61% | 10,62% |
| grobid 0.3.3c avec seg-a+c et refseg-test-2500 | 120 | 2916 | 10-5 | 32,6 | 54,4 | 5,18% | 10,67% |
| grobid 0.3.3d (=avec seg-a+c) | 120 | 416 | 10-5 | 57,4 | 69,1 | 5,60% | 10,62% |
| grobid 0.3.3d (=avec seg-d-51PL) | 91 | 416 | 10-5 | 55,5 | 69,6 | 5,94% | 13,94% |
| grobid 0.3.3d (=avec seg-a+d) | 131 | 416 | 10-5 | 57,7 | 69,2 | 5,92% | 10,47% |
| grobid 0.3.3d (=avec seg-a+c et refseg-bcor-25) | 120 | 999 | 10-5 | 54,6 | 60,9 | 1,55% | 11,21% |
| grobid 0.3.3d (=avec seg-a+c et refseg-bcor-25 et eps 10- | 120 | 999 | 10-4 | 61,8 | 62,2 | 2,23% | 10,91% |
| grobid 0.3.3d (=avec seg-a+c et refseg-bcor-25 et eps 10- | 120 | 999 | 10-3 | 60,3 | 66,9 | 2,18% | 11,14% |

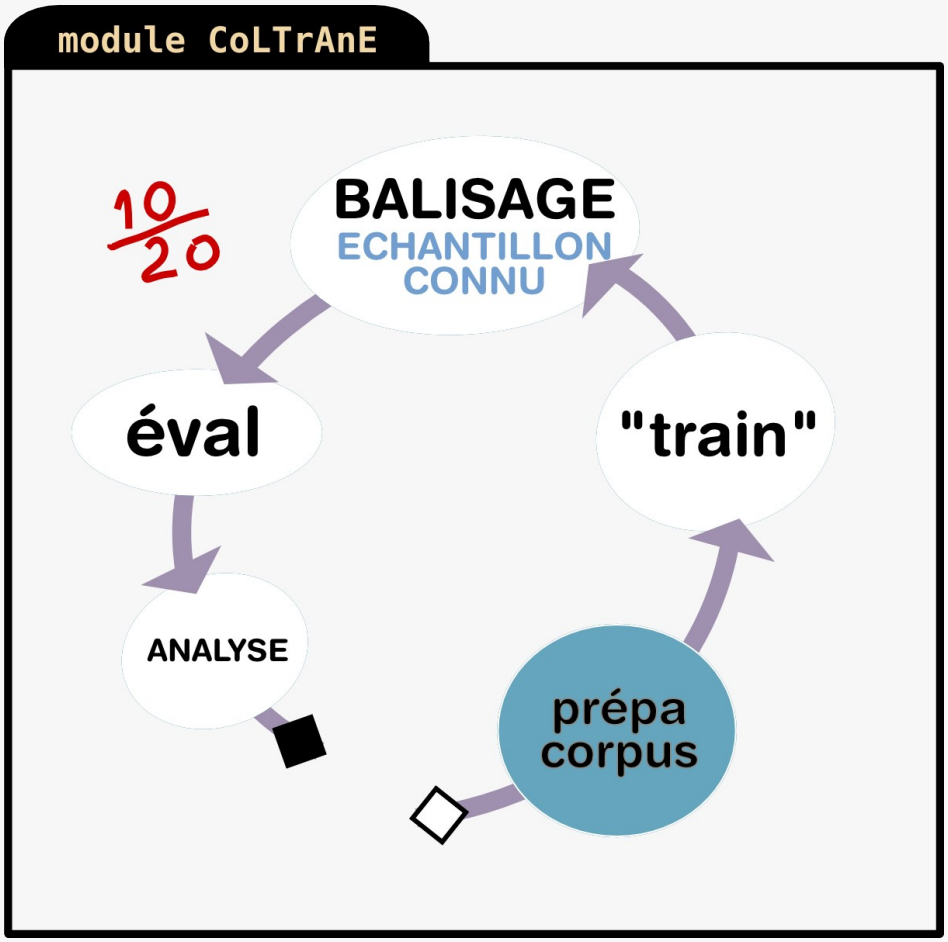
- **Un échantillon de la base**
  - déjà annoté
- **Produit un modèle**
  - recréera annotations sur toute entrée
- **On évalue ce modèle**
  - sur un autre échantillon connu
  - Rappel/Précision

# 6) Méthode des « Cycles » incrémentaux



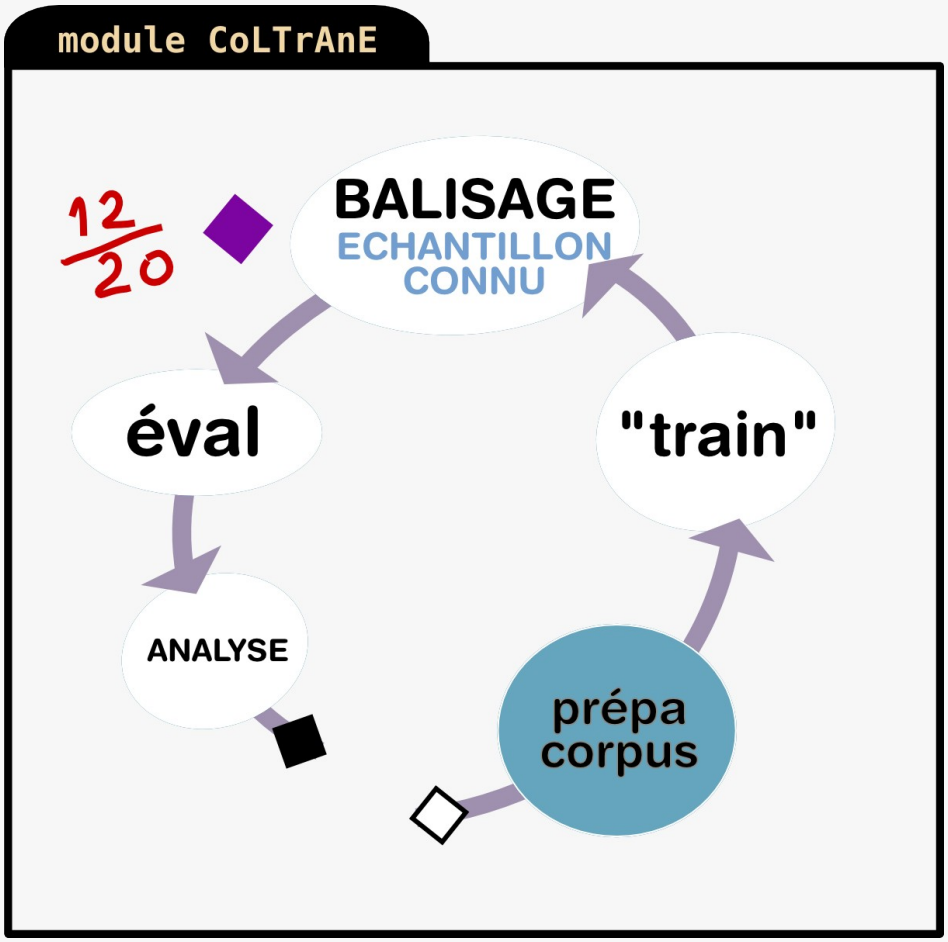

| | Ndocs corpus seg | Nbibl corpus refseg | eps | Rappel | Précision | taux docs xerr | taux docs vides |
|---|---|---|---|---|---|---|---|
| grobid 0.2.1 branche trunk | 0 | 0 | | 46,2 | 59,2 | 0,41% | 8,89% |
| grobid 0.3.0 branche segm | 36 | 416 | 10-5 | 60,6 | 64 | 4,93% | 3,44% |
| grobid 0.3.0 branche segm | 36 | 416 | 10-5 | 63,3 | 67 | 4,93% | 3,44% |
| grobid 0.3.0 branche segm + seg-a-40 | 76 | 416 | 10-5 | 61 | 67,4 | 3,35% | 2,46% |
| grobid 0.3.3 | 40 | 416 | 10-5 | 45,6 | 68,6 | 4,60% | 21,97% |
| grobid 0.3.3 + seg-a-40 | 80 | 416 | 10-5 | 56,6 | 68,9 | 6,12% | 9,51% |
| grobid 0.3.3b (= avec seg-b-18PL) | 58 | 416 | 10-5 | 49,8 | 69,7 | 4,20% | 21,21% |
| grobid 0.3.3c (= avec seg-c-40PL) | 80 | 416 | 10-5 | 55,8 | 69,4 | 6,03% | 11,00% |
| grobid 0.3.3c (avec seg-a+c) | 120 | 416 | 10-5 | 57,5 | 69,2 | 5,61% | 10,62% |
| grobid 0.3.3c avec seg-a+c et refseg-test-2500 | 120 | 2916 | 10-5 | 32,6 | 54,4 | 5,18% | 10,67% |
| grobid 0.3.3d (=avec seg-a+c) | 120 | 416 | 10-5 | 57,4 | 69,1 | 5,60% | 10,62% |
| grobid 0.3.3d (=avec seg-d-51PL) | 91 | 416 | 10-5 | 55,5 | 69,6 | 5,94% | 13,94% |
| grobid 0.3.3d (=avec seg-a+d) | 131 | 416 | 10-5 | 57,7 | 69,2 | 5,92% | 10,47% |
| grobid 0.3.3d (=avec seg-a+c et refseg-bcor-25) | 120 | 999 | 10-5 | 54,6 | 60,9 | 1,55% | 11,21% |
| grobid 0.3.3d (=avec seg-a+c et refseg-bcor-25 et eps 10- | 120 | 999 | 10-4 | 61,8 | 62,2 | 2,23% | 10,91% |
| grobid 0.3.3d (=avec seg-a+c et refseg-bcor-25 et eps 10- | 120 | 999 | 10-3 | 60,3 | 66,9 | 2,18% | 11,14% |

# 6) Méthode des « Cycles » incrémentaux



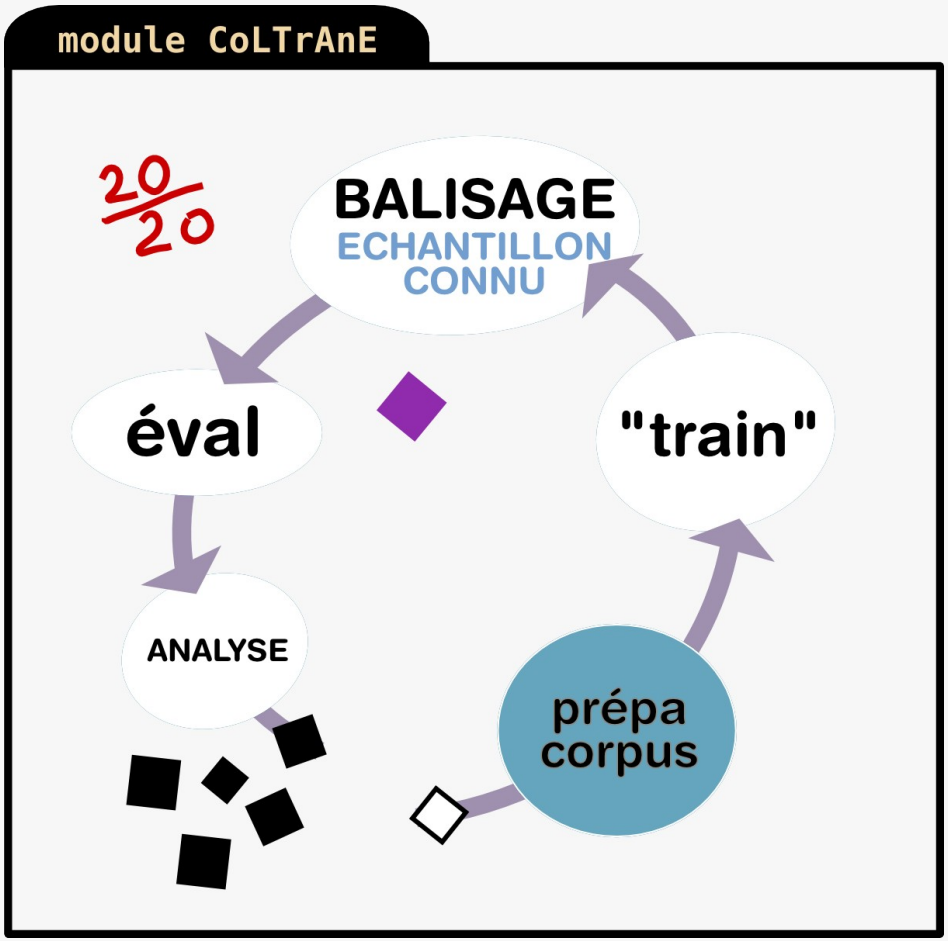

| | Ndocs corpus seg | Nbibl corpus refseg | eps | Rappel | Précision | taux docs xerr | taux docs vides |
|---|---|---|---|---|---|---|---|
| grobid 0.2.1 branche trunk | 0 | 0 | | 46,2 | 59,2 | 0,41% | 8,89% |
| grobid 0.3.0 branche segm | 36 | 416 | 10-5 | 60,6 | 64 | 4,93% | 3,44% |
| grobid 0.3.0 branche segm | 36 | 416 | 10-5 | 63,3 | 67 | 4,93% | 3,44% |
| grobid 0.3.0 branche segm + seg-a-40 | 76 | 416 | 10-5 | 61 | 67,4 | 3,35% | 2,46% |
| grobid 0.3.3 | 40 | 416 | 10-5 | 45,6 | 68,6 | 4,60% | 21,97% |
| grobid 0.3.3 + seg-a-40 | 80 | 416 | 10-5 | 56,6 | 68,9 | 6,12% | 9,51% |
| grobid 0.3.3b (= avec seg-b-18PL) | 58 | 416 | 10-5 | 49,8 | 69,7 | 4,20% | 21,21% |
| grobid 0.3.3c (= avec seg-c-40PL) | 80 | 416 | 10-5 | 55,8 | 69,4 | 6,03% | 11,00% |
| grobid 0.3.3c (avec seg-a+c) | 120 | 416 | 10-5 | 57,5 | 69,2 | 5,61% | 10,62% |
| grobid 0.3.3c avec seg-a+c et refseg-test-2500 | 120 | 2916 | 10-5 | 32,6 | 54,4 | 5,18% | 10,67% |
| grobid 0.3.3d (=avec seg-a+c) | 120 | 416 | 10-5 | 57,4 | 69,1 | 5,60% | 10,62% |
| grobid 0.3.3d (=avec seg-d-51PL) | 91 | 416 | 10-5 | 55,5 | 69,6 | 5,94% | 13,94% |
| grobid 0.3.3d (=avec seg-a+d) | 131 | 416 | 10-5 | 57,7 | 69,2 | 5,92% | 10,47% |
| grobid 0.3.3d (=avec seg-a+c et refseg-bcor-25) | 120 | 999 | 10-5 | 54,6 | 60,9 | 1,55% | 11,21% |
| grobid 0.3.3d (=avec seg-a+c et refseg-bcor-25 et eps 10- | 120 | 999 | 10-4 | 61,8 | 62,2 | 2,23% | 10,91% |
| grobid 0.3.3d (=avec seg-a+c et refseg-bcor-25 et eps 10- | 120 | 999 | 10-3 | 60,3 | 66,9 | 2,18% | 11,14% |

# 6) Méthode des « Cycles » incrémentaux




| | Ndocs corpus seg | Nbibl corpus refseg | eps | Rappel | Précision | taux docs xerr | taux docs vides |
|---|---|---|---|---|---|---|---|
| grobid 0.2.1 branche trunk | 0 | 0 | | 46,2 | 59,2 | 0,41% | 8,89% |
| grobid 0.3.0 branche segm | 36 | 416 | 10-5 | 60,6 | 64 | 4,93% | 3,44% |
| grobid 0.3.0 branche segm | 36 | 416 | 10-5 | 63,3 | 67 | 4,93% | 3,44% |
| grobid 0.3.0 branche segm + seg-a-40 | 76 | 416 | 10-5 | 61 | 67,4 | 3,35% | 2,46% |
| grobid 0.3.3 | 40 | 416 | 10-5 | 45,6 | 68,6 | 4,60% | 21,97% |
| grobid 0.3.3 + seg-a-40 | 80 | 416 | 10-5 | 56,6 | 68,9 | 6,12% | 9,51% |
| grobid 0.3.3b (= avec seg-b-18PL) | 58 | 416 | 10-5 | 49,8 | 69,7 | 4,20% | 21,21% |
| grobid 0.3.3c (= avec seg-c-40PL) | 80 | 416 | 10-5 | 55,8 | 69,4 | 6,03% | 11,00% |
| grobid 0.3.3c (avec seg-a+c) | 120 | 416 | 10-5 | 57,5 | 69,2 | 5,61% | 10,62% |
| grobid 0.3.3c avec seg-a+c et refseg-test-2500 | 120 | 2916 | 10-5 | 32,6 | 54,4 | 5,18% | 10,67% |
| grobid 0.3.3d (=avec seg-a+c) | 120 | 416 | 10-5 | 57,4 | 69,1 | 5,60% | 10,62% |
| grobid 0.3.3d (=avec seg-d-51PL) | 91 | 416 | 10-5 | 55,5 | 69,6 | 5,94% | 13,94% |
| grobid 0.3.3d (=avec seg-a+d) | 131 | 416 | 10-5 | 57,7 | 69,2 | 5,92% | 10,47% |
| grobid 0.3.3d (=avec seg-a+c et refseg-bcor-25) | 120 | 999 | 10-5 | 54,6 | 60,9 | 1,55% | 11,21% |
| grobid 0.3.3d (=avec seg-a+c et refseg-bcor-25 et eps 10- | 120 | 999 | 10-4 | 61,8 | 62,2 | 2,23% | 10,91% |
| grobid 0.3.3d (=avec seg-a+c et refseg-bcor-25 et eps 10- | 120 | 999 | 10-3 | 60,3 | 66,9 | 2,18% | 11,14% |

# 6) Méthode des « Cycles » incrémentaux

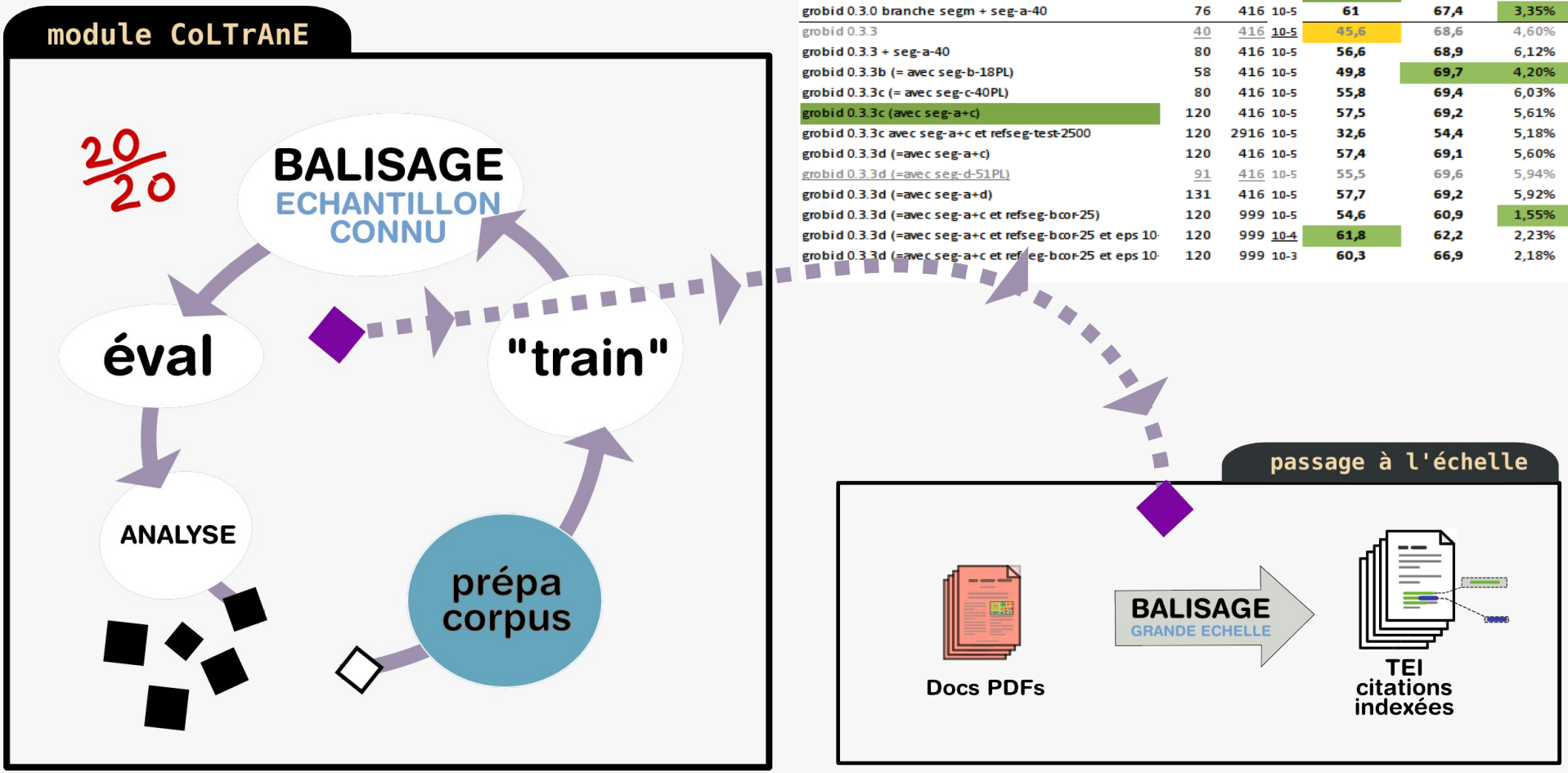



| | Ndocs corpus seg | Nbibl corpus refseg | eps | Rappel | Précision | taux docs xerr | taux docs vides |
|---|---|---|---|---|---|---|---|
| grobid 0.2.1 branche trunk | 0 | 0 | | 46,2 | 59,2 | 0,41% | 8,89% |
| grobid 0.3.0 branche segm | 36 | 416 | 10-5 | 60,6 | 64 | 4,93% | 3,44% |
| grobid 0.3.0 branche segm | 36 | 416 | 10-5 | 63,3 | 67 | 4,93% | 3,44% |
| grobid 0.3.0 branche segm + seg-a-40 | 76 | 416 | 10-5 | 61 | 67,4 | 3,35% | 2,46% |
| grobid 0.3.3 | 40 | 416 | 10-5 | 45,6 | 68,6 | 4,60% | 21,97% |
| grobid 0.3.3 + seg-a-40 | 80 | 416 | 10-5 | 56,6 | 68,9 | 6,12% | 9,51% |
| grobid 0.3.3b (= avec seg-b-18PL) | 58 | 416 | 10-5 | 49,8 | 69,7 | 4,20% | 21,21% |
| grobid 0.3.3c (= avec seg-c-40PL) | 80 | 416 | 10-5 | 55,8 | 69,4 | 6,03% | 11,00% |
| grobid 0.3.3c (avec seg-a+c) | 120 | 416 | 10-5 | 57,5 | 69,2 | 5,61% | 10,62% |
| grobid 0.3.3c avec seg-a+c et refseg-test-2500 | 120 | 2916 | 10-5 | 32,6 | 54,4 | 5,18% | 10,67% |
| grobid 0.3.3d (=avec seg-a+c) | 120 | 416 | 10-5 | 57,4 | 69,1 | 5,60% | 10,62% |
| grobid 0.3.3d (=avec seg-d-51PL) | 91 | 416 | 10-5 | 55,5 | 69,6 | 5,94% | 13,94% |
| grobid 0.3.3d (=avec seg-a+d) | 131 | 416 | 10-5 | 57,7 | 69,2 | 5,92% | 10,47% |
| grobid 0.3.3d (=avec seg-a+c et refseg-bcor-25) | 120 | 999 | 10-5 | 54,6 | 60,9 | 1,55% | 11,21% |
| grobid 0.3.3d (=avec seg-a+c et refseg-bcor-25 et eps 10- | 120 | 999 | 10-4 | 61,8 | 62,2 | 2,23% | 10,91% |
| grobid 0.3.3d (=avec seg-a+c et refseg-bcor-25 et eps 10- | 120 | 999 | 10-3 | 60,3 | 66,9 | 2,18% | 11,14% |

# 6) Méthode des « Cycles » incrémentaux

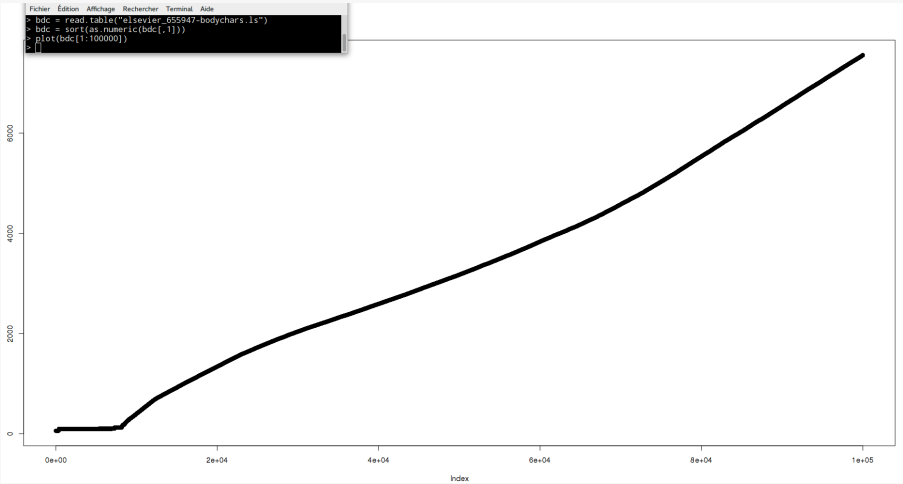
## Un modèle jugé valable peut être inséré dans le moteur de balisage

| | Ndocs corpus seg | Nbibl corpus refseg | eps | Rappel | Précision | taux docs xerr | taux docs vides |
|---|---|---|---|---|---|---|---|
| grobid 0.2.1 branche trunk | 0 | 0 | | 46,2 | 59,2 | 0,41% | 8,89% |
| grobid 0.3.0 branche segm | 36 | 416 | 10-5 | 60,6 | 64 | 4,93% | 3,44% |
| grobid 0.3.0 branche segm | 36 | 416 | 10-5 | 63,3 | 67 | 4,93% | 3,44% |
| grobid 0.3.0 branche segm + seg-a-40 | 76 | 416 | 10-5 | 61 | 67,4 | 3,35% | 2,46% |
| grobid 0.3.3 | 40 | 416 | 10-5 | 45,6 | 68,6 | 4,60% | 21,97% |
| grobid 0.3.3 + seg-a-40 | 80 | 416 | 10-5 | 56,6 | 68,9 | 6,12% | 9,51% |
| grobid 0.3.3b (= avec seg-b-18PL) | 58 | 416 | 10-5 | 49,8 | 69,7 | 4,20% | 21,21% |
| grobid 0.3.3c (= avec seg-c-40PL) | 80 | 416 | 10-5 | 55,8 | 69,4 | 6,03% | 11,00% |
| grobid 0.3.3c (avec seg-a+c) | 120 | 416 | 10-5 | 57,5 | 69,2 | 5,61% | 10,62% |
| grobid 0.3.3c avec seg-a+c et refseg-test-2500 | 120 | 2916 | 10-5 | 32,6 | 54,4 | 5,18% | 10,67% |
| grobid 0.3.3d (=avec seg-a+c) | 120 | 416 | 10-5 | 57,4 | 69,1 | 5,60% | 10,62% |
| grobid 0.3.3d (=avec seg-d-51PL) | 91 | 416 | 10-5 | 55,5 | 69,6 | 5,94% | 13,94% |
| grobid 0.3.3d (=avec seg-a+d) | 131 | 416 | 10-5 | 57,7 | 69,2 | 5,92% | 10,47% |
| grobid 0.3.3d (=avec seg-a+c et refseg-bcor-25) | 120 | 999 | 10-5 | 54,6 | 60,9 | 1,55% | 11,21% |
| grobid 0.3.3d (=avec seg-a+c et refseg-bcor-25 et eps 10- | 120 | 999 | 10-4 | 61,8 | 62,2 | 2,23% | 10,91% |
| grobid 0.3.3d (=avec seg-a+c et refseg-bcor-25 et eps 10- | 120 | 999 | 10-3 | 60,3 | 66,9 | 2,18% | 11,14% |



# 6) Au passage une remarque : une bonne prépa corpus c'est connaître ses données et en mettre un peu de chaque

- **Dépouiller les types de documents annoncés**
  - articles de revues
  - éditoriaux, courriers
  - news, résultats expérimentaux
  - indexs => dont compilations refbibs



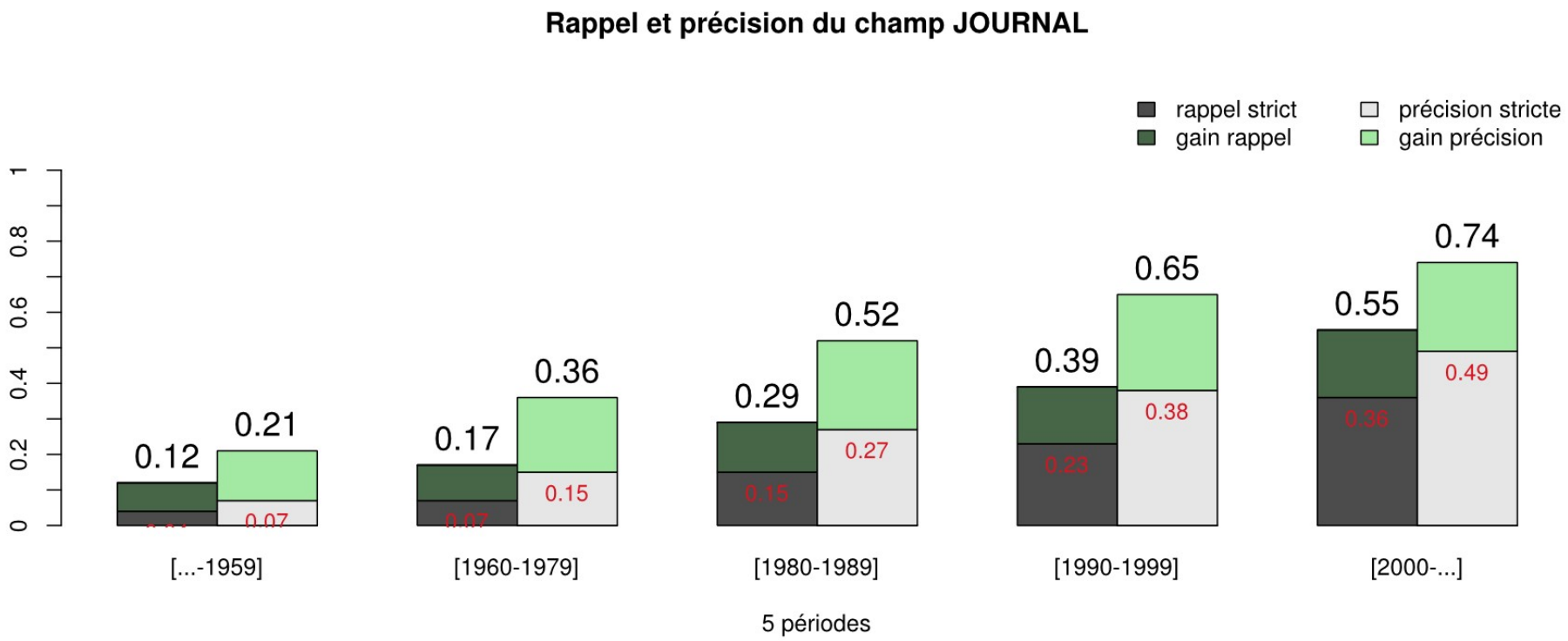

**+ se faire sa typologie empirique**

- Ex stats
  - selon son domaine
  - selon sa source
  - selon la longueur du texte
    - (courbe ci-contre)
    - Echantillon > boucle for | wc -c > tableau > graphique

- **Stats sur toutes les infos qui permettent d'avoir une vue d'ensemble sur les grands corpus**

- **Concrètement**
  - Faire des requêtes facettes API
  - Utiliser le sampler de https://git.istex.fr/loth/libconsulte
    - Inpecter les décomptes de pool_cache de ce sampler
    - Il est intégré dans bako make_set
  - Discuter avec ISTEX-DATA

# 6) … et une bonne évaluation par catégories




**Par exemple on voit ici que + la publication est récente, + le balisage est facile**

(rôle des formats digitaux mais aussi des standards/conventions de citations qui ont changé dans le temps)

- Une bonne évaluation relie nos catégories de documents au rappel et précision
- Concrètement (métadonnées par document concernent input) + (évaluation des bibs pour ces mêmes documents) => résultats croisés
- cf. script R : https://git.istex.fr/loth/refbibs-stack/blob/master/bib-eval/eval_results.r

# 7) Documentation en vrac autour des CRF

- **Pourquoi les CRF plutôt que les transducteurs**
- **Logique du balisage CRF**
  - Préalable sur les chaînes de Markov caché
  - Automate de Moore pour voir le côté séquentiel
  - Des décomptes d'exemples aux règles de prédiction
  - Présentation du modèle discriminant pour chaque prédiction (pour les matheux)
- **Exemple de MacCallum sur l'extraction de tableaux de rapports txt**

# 7) Pourquoi pas un traitements transducteurs ?

- **La citation biblio. est une séquence linguistique « formulaire »**
- **« Grammaire de transformations » : de l'info à la chaîne de signes**
  - BIB <= DATE + ART + JOURN
  - ART <= (k x AUT) + [TIT] + [pp]
  - JOURN <= $NomJ_{ital}$ + vol
    - ...
- **Habituellement on traite ça par transducteurs**
  - Exemple courant : les chaînes Unitex
  - partir des formes connues
  - observer les règles de capture utilisées dans la vie
  - modéliser les règles en graphes génératifs
- **Problème**
  - mode passée ?
  - implémentations plus lentes que les CRF
  - pas facile de l'adapter sur de nouvelles données

# Logique des modèles de balisage CRF

- **En balisage (annotation) on prendra le texte en entrée comme un long tableau**
  - une ligne = 1 observable "str" + infos associées ou immédiatement reconnues par match
- **A chaque mot (ou groupe) à annoter va correspondre une ligne du tableau avec les infos qu'on a déjà sur le mot**
  - on appelle souvent ces infos les "**features**" ou "attributs"
  - à chaque observable (mot ou groupe à baliser) sa série d'infos => ce sont autant d'indices pour décider

    => on parle d'un vecteur de features **x** associé au mot

    => par exemple : "ce mot est en majuscules" "il est dans ma liste des CODEN" ou "il est composé de 4 chiffres" sont des features
- Pour chaque mot (ou groupe) il y aura une **case vide dans le tableau**

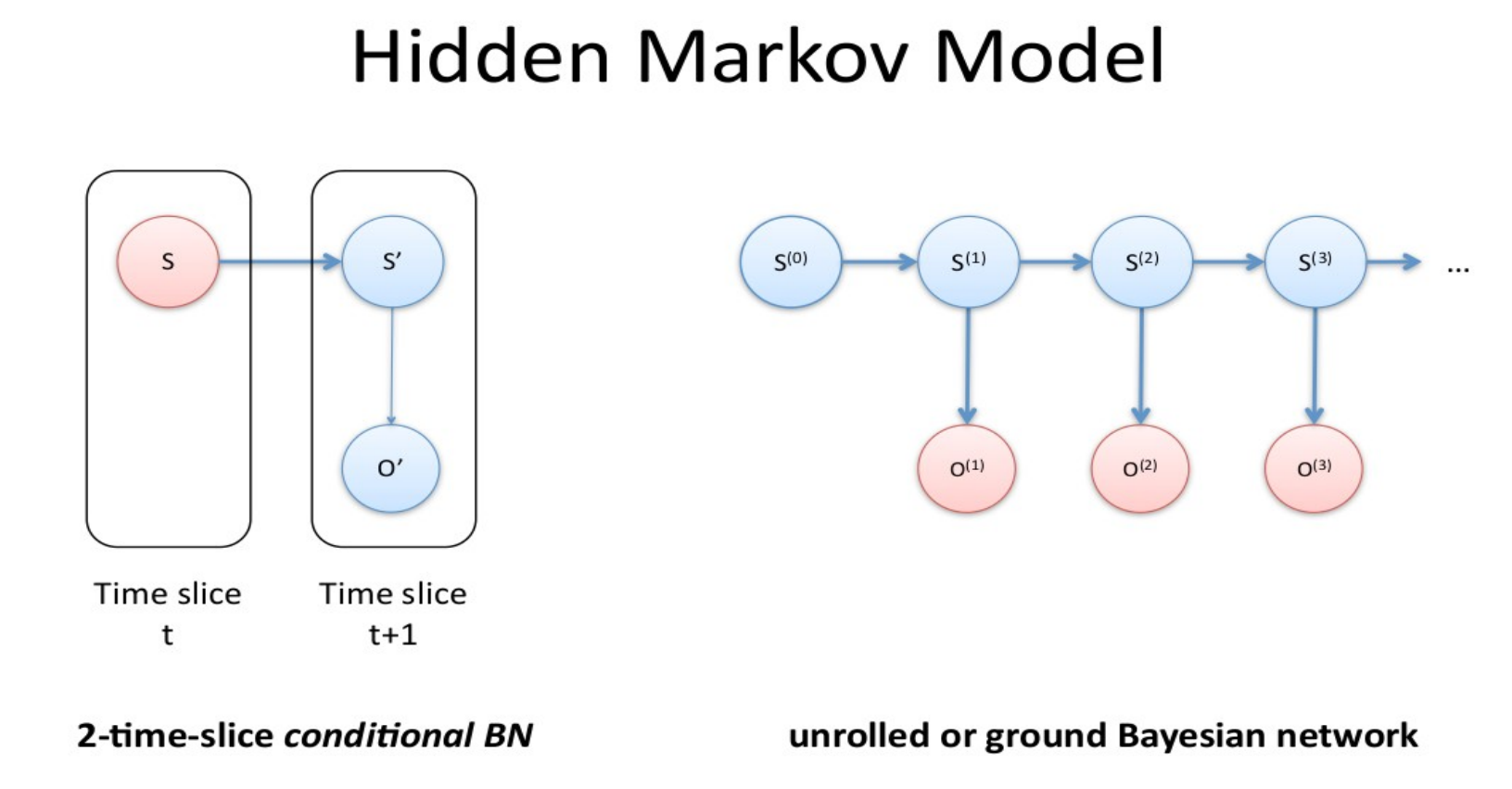
  ==> c'est l'étiquette à prédire !!!

  ==> par exemple "DATE" ou "AUTEUR" dans un balisage refbib
  - ou encore "DET" ou "ADJ" ou "NOM" ou "VERBE" ... dans un balisage POS
  - etc (balisage de groupes pour entités nommées, balisage de lignes pour découpe de document, nombreuses autres applications)
- Le fameux "MODELE" c'est un ensemble de règles pour passer des "features" à une prédiction de l'étiquette
  - concrètement des règles de scoring
    - par ex "il est tout en majuscules" donne des points pour prédire l'étiquette "SIGLE"

# Préalable : modèles « Markov caché »

Pour comprendre l'utilisation de la séquentialité, on regarde d'abord un modèle plus simple

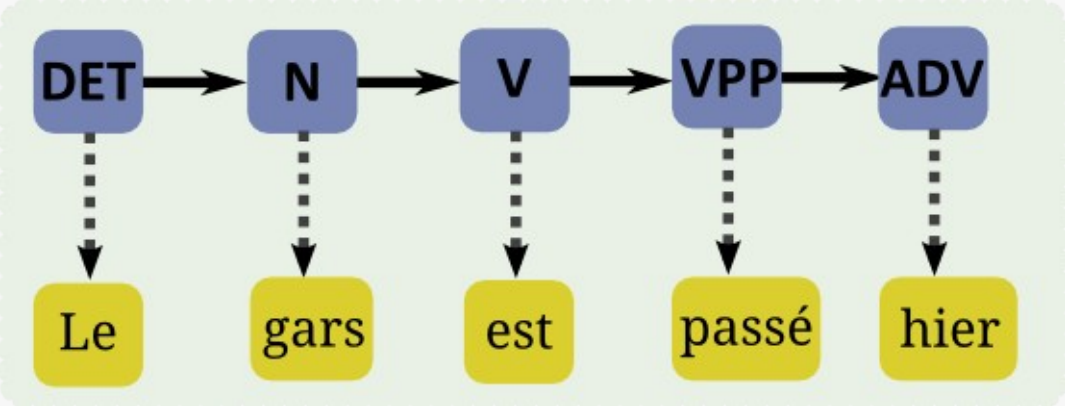
- Si on cherche HMM, voilà le genre de schéma qu'on trouve dans l'état de l'art :




source McCallum http://people.cs.umass.edu/~mccallum/courses/gm2011/04-undirected.pdf

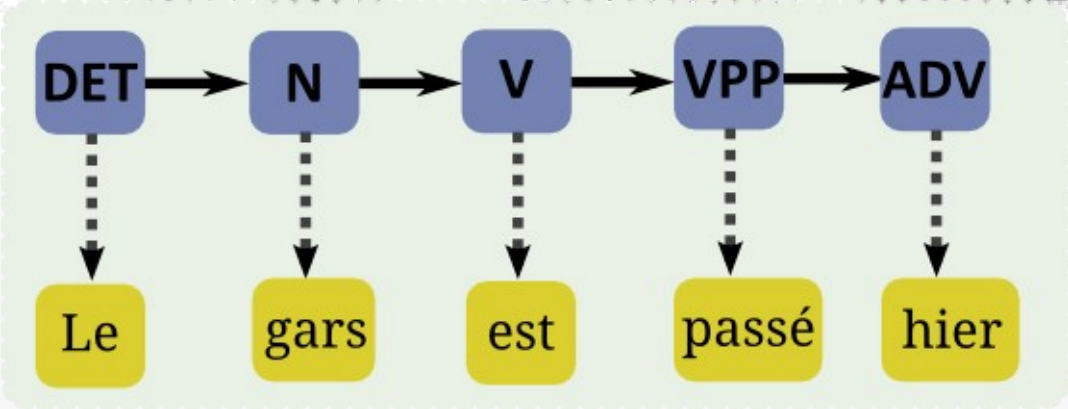
# Automate de Moore

- **Le schéma page précédente est un automate de Moore**
  - Représente la séquence des données
    - Des observables (input du baliseur)
    - Des états cachés à deviner (output)
- **Exemple en étiquetage POS**




- **Attention : ça ne représente pas le modèle profond**
  - Cet automate de Moore montre la route cachée (bleu) et ses effets visibles (jaune)
  - Le modèle à entraîner serait plutôt le code de la route et les stats de mortalité routière
    - Transitions possibles/impossibles dans l'absolu
    - Indices qui permettent de retrouver le bleu quand on a le jaune
      - aka modèle stationnaire ou automate probabiliste à états finis

# Clarification : quels décomptes utiles ?




← états cachés

← observables

- **Le modèle aura des ambigüités à résoudre**
  - « est » peut être un V ou un N
  - « passé » peut être un N, un ADJ ou un VPP
- **Utilisation des données d'entraînement**
  - on arrive à avoir les fréquences relatives des relations (étiquette, mot)
    - « passé, ADJ » vu 14 fois
    - « passé, VPP » vu 7 fois
    - « passé, N » vu 3 fois
  - **et on relève aussi les fréquences des séquences d'étiquettes** (dits bigrammes d'étiquettes)
    - « V puis VPP »
    - « DET puis N »

# Chaque élément d'un CRF est un modèles discriminant

- **Oublions le séquentiel pour le moment**
  - Les règles sont diverses et changent imperceptiblement
  - Les indices symboliques font feu de tout bois
    - style graphique
    - marques typographiques
    - unités lexicales +/- fonctionnalisées (« journal », « vol »)
  - Ces indices peuvent être comptés et analysés statistiquement
- **Un classifieur discriminant c'est quoi ?**
  - fonction réelle
    - qui ajoute les features
    - pour prendre une décision binaire sur un label
  - $f(x_1,x_2,\ldots,x_n) = y_i$ ← valeur du label recherché 0 ou 1
- **Exemple courant : la régression logistique**

$$\operatorname{logit}(p_i) = \ln\left(\frac{p_i}{1-p_i}\right) = \beta_1 x_{1,i} + \cdots + \beta_k x_{k,i}.$$

  - une manière habile de partir des fréquences observées
  - de passer par les probas
  - et d'arriver à une courbe sur R
    - pleine de logarithmes :-( => permet à la fonction d'être définie sur [-∞ ; +∞]
    - mais ça ne change rien :-) => pour revenir aux probas il suffit de prendre l'exp

# Logique des modèles de balisage CRF (2)

- **Le modèle CRF utilise donc des "features"...**
  - au lieu de regarder juste la chaîne de caractères et sa fréquence
  - les features sont souvent ce qu'on aurait mis dans une règle regexp ou un graphe
  - **par exemple "il commence par une majuscule" ou  "il contient www" ou "il est tout en chiffres" ou "il est dans ma liste de noms"**
  - concrètement ***des règles de match*** permettent de passer de l'observable (jaune) à une série d'indices oui/non
  - On peut appeler ça des modèles discriminants (et ils sont enchaînés en séquence)
- **Le fameux "MODELE CRF" c'est un ensemble de règles pour passer des features observables à une prédiction de l'étiquette ("label')**
  - *concrètement* ***des règles de scoring*** *permettent de passer de la série l'indice à l'annotation*
  - par ex j'ai un input "2015" et je dois lui donner un label "DATE"
    - parmi mes attributs possibles "2015" remplit la feature "il est tout en chiffres":
      - ça donne des points pour prédire les étiquettes "ISSN" et "DATE"
      - ça enlève des points pour prédire  les étiquettes "AUTEUR" ou "TITRE"
    - il a aussi la feature "sa longueur est courte (pas beaucoup de caractères)"
      - ça donne des points pour "DATE" et "AUTEUR"
      - ça enlève des points pour "ISSN" et "TITRE"
    - après toutes mes features examinées, le score les plus haut est "DATE" !

# Logique des modèles de balisage CRF (3)

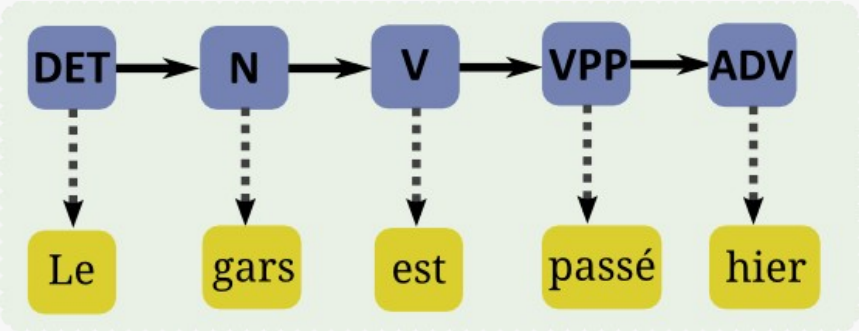
donc... après toutes mes features examinées, le score les plus haut est "DATE" ! ==> prédiction ==> annotation ==> sortie balisée

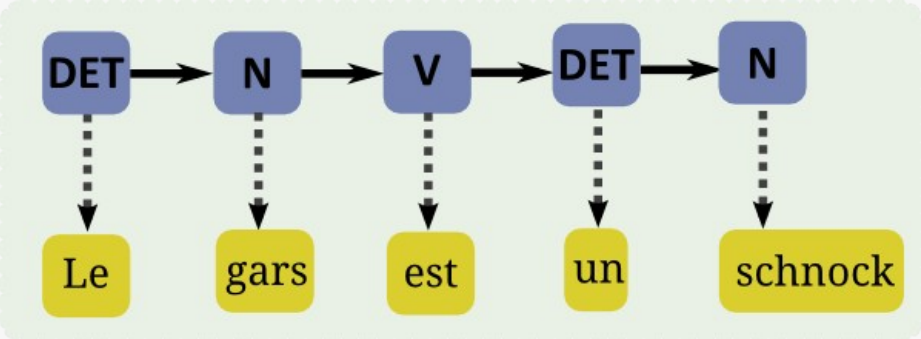
- (pour chaque score, en maths ils parlent de "fonctions locales")
  - nombre de fois label bleu / score total feature Z verte => proba sur [0;1]
  - proba => log de la (proba/(1-proba)) => fonction locale phi de décision (sur R) pour un label (par exemple AUTEUR)
  - si la fonction phi est négative => pas AUTEUR
  - si elle est positive => AUTEUR
- quand plusieurs étiquettes on prend la séquence d'étiquette qui maximise toutes les fonctions à la fois

- **fixer les scores de chaque features pour chaque fonction s'appelle "lancer un entraînement"**
  - **on parle de "poids" des features dans la décision**
  - **se fait sur un corpus où on a déjà les bonnes étiquettes**
  - **on parle de corpus "gold"**

# passage au CRF : stockage des décomptes

- On garde l'idée d'un corpus et de décomptes




- on va faire les mêmes décomptes de séquences horizontales dans le corpus
- ajouter au niveau vertical des éléments intermédaires entre les labels et les input => "features" (en vert)
- moyennant une équation + longue et une hypothèse simplificatrice

# Exemple concret en segmentation

## Table Extraction from Government Reports

Cash receipts from marketings of milk during 1995 at $19.9 billion dollars, was slightly below 1994. Producer returns averaged $12.93 per hundredweight, $0.19 per hundredweight below 1994. Marketings totaled 154 billion pounds, 1 percent above 1994. Marketings include whole milk sold to plants and dealers as well as milk sold directly to consumers.

An estimated 1.56 billion pounds of milk were used on farms where produced, 8 percent less than 1994. Calves were fed 78 percent of this milk with the remainder consumed in producer households.

Milk Cows and Production of Milk and Milkfat:
United States, 1993-95

| Year | Number of Milk Cows 1/ | Production of Milk and Milkfat 2/ | | | | |
|---|---|---|---|---|---|---|
| | | Per Milk Cow | | Percentage of Fat in All Milk Produced | Total | |
| | | Milk | Milkfat | | Milk | Milkfat |
| | 1,000 Head | --- Pounds --- | | Percent | Million Pounds | |
| 1993 | 9,589 | 15,704 | 575 | 3.66 | 150,582 | 5,514.4 |
| 1994 | 9,500 | 16,175 | 592 | 3.66 | 153,664 | 5,623.7 |
| 1995 | 9,461 | 16,451 | 602 | 3.66 | 155,644 | 5,694.3 |

1/ Average number during year, excluding heifers not yet fresh.
2/ Excludes milk sucked by calves.

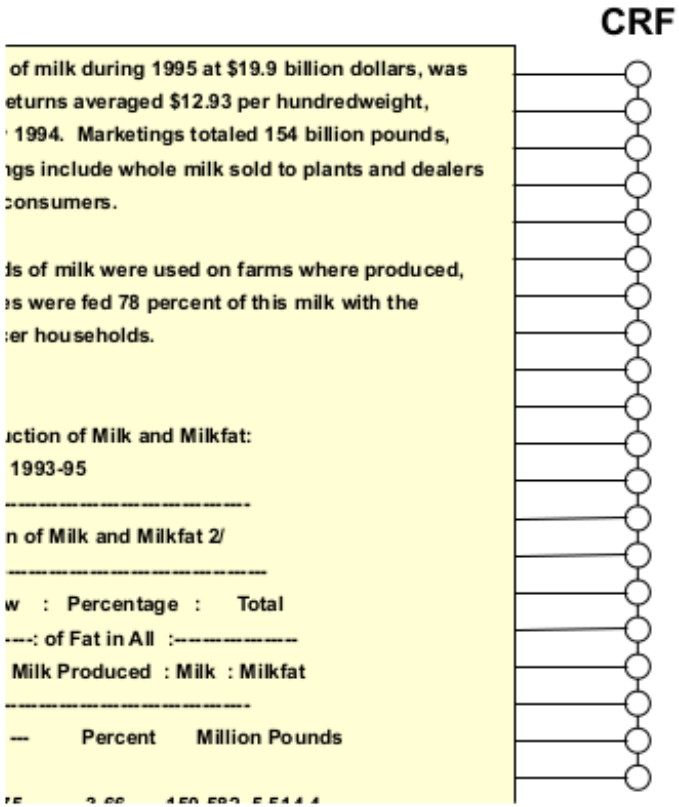
# Features utiles pour la segmentation

- features multiples
- tout ce qui paraît pertinent
- ≠ niveaux linguistiques de représentation des données du texte

## Table Extraction from Government Reports

*[Pinto, McCallum, Wei, Croft, 2003 SIGIR]*

**100+ documents from www.fedstats.gov**




**Labels:**

- Non-Table
- Table Title
- Table Header
- Table Data Row
- Table Section Data Row
- Table Footnote
- *... (12 in all)*

**Features:**

- Percentage of digit chars
- Percentage of alpha chars
- Indented
- Contains 5+ consecutive spaces
- Whitespace in this line aligns with prev.
- ...
- Conjunctions of all previous features, time offset: {0,0}, {-1,0}, {0,1}, {1,2}.

# Annexe: références


- Besagni & Belaïd (2004). Citation recognition for scientific publications in digital libraries.
- **Giles, Bollacker & Lawrence (1998). CITESEER: An automatic citation indexing system.**
- **Kim, Bellot, Faath & Dacos (2011). BILBO: Automatic annotation of bibliographical references in digital humanities books, articles and blogs.**
- **Lopez (2009). GROBID: Combining automatic bibliographic data recognition and term extraction for scholarship publications.**
- Peng & McCallum (2006). Information extraction from research papers using conditional random fields.
- Pinto, McCallum, Wei, Bruce Croft(2003). Table Extraction using CRF

  (https://people.cs.umass.edu/~mccallum/papers/crftable-sigir2003.pdf)
- Seymore, McCallum & Rosenfeld (1999). CORA: Learning hidden Markov model structure for information extraction.
- **Shotton, Dutton & O'Steen (2014). JISC: Open Citations Database.**
- Tkaczyk & Bolikowski (2011). Workflow of metadata extraction from retro-born-digital documents.