Journée ATALA: quels analyseurs syntaxiques pour le français ?
Samedi 10 octobre 2009

TagParser: combiner un corpus annoté avec un corpus brut

Gil Francopoulo / Tagmatica
www.tagmatica.com

Introduction

- La présentation se situe dans le contexte d'une recherche appliquée du secteur privé dans le domaine de la presse + blogs.
- Les diapositives traitent à la fois de la démarche méthodologique et de la description d'un système existant.

Ligne directrice

- Après avoir constaté que la combinaison d'un bon lexique avec un bon analyseur ne suffit pas à constituer un analyseur industriel, nous avons cherché des solutions efficaces pour aller un peu plus loin.
- Nous sommes arrivés à la conclusion qu'il faut confronter l'analyseur à des corpus réels mais aussi il faut gérer le cycle de vie de l'analyseur dans un atelier logiciel qui permette une évolution incrémentale et stable sur plusieurs années.

Problématique

- La difficulté ne réside pas tant dans l'élaboration d'un analyseur qui donnerait un bon résultat dans la majeure partie des circonstances: atteindre une FM de 55% aux tests Easy-Passage est en fait assez facile.
- Le défi est plutôt d'améliorer le score de manière incrémentale quand des insuffisances ou problèmes sont rencontrés. Et passer de 55% à 96% est bien plus complexe.

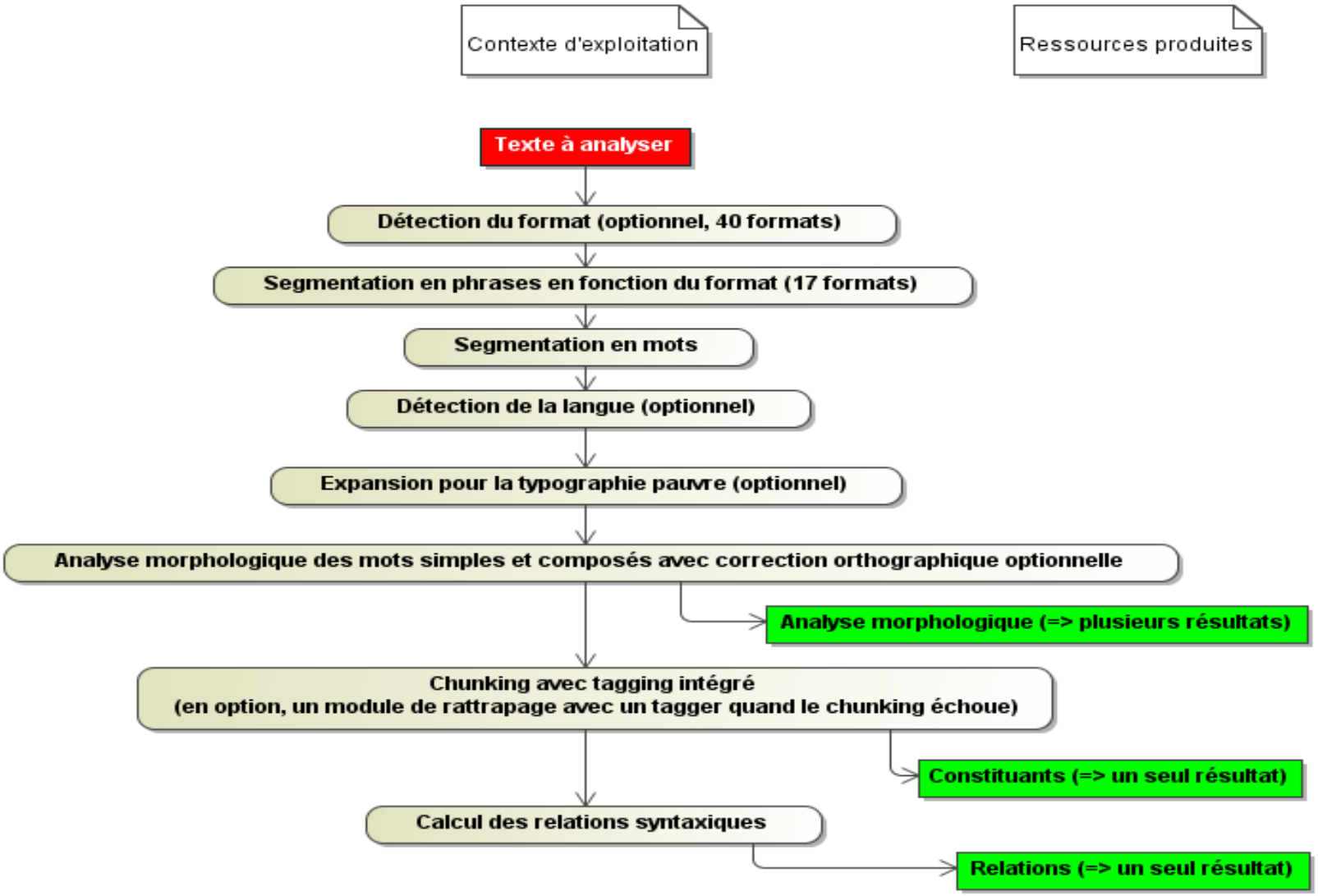
Quelles ressources ?

- Les règles. Leur écriture est aisée dans un premier temps mais la tâche est plus ardue quand la complexité et l'interdépendance augmentent.
- Les lexiques. Si leur maintenance est contrôlable, leur création est complexe car elle nécessite une abstraction importante si l'on désire atteindre une couverture satisfaisante. Utilisation de LMF (Lexical Markup Framework).
- Un corpus annoté. Si les règles de codage sont cohérentes, la maintenance est aisée, mais la tâche d'annotation est laborieuse (130 Km).
- Un corpus brut, qui pris isolément n'est pas d'une grande utilité (600 Mm).

Quel analyseur ?

- TagParser: analyseur de type montant
- Sont enchaînés sous forme d'un pipeline: un segmenteur, un analyseur morphologique, un chunker (qui effectue le tagging) et un module de calcul des relations syntaxiques.
- La sortie comporte trois types de résultat: les chunks, les relations syntaxiques et les entitées nommées.
- La communication d'un module à l'autre respecte les principes du Linguistic Annotation Framework (LAF+MAF+SynAF) dans le sens où chaque module ajoute une annotation de type déportée (standoff) sur la donnée transmise.

Chaîne d'analyse syntaxique (sans les traitmts sémantiques)



Ateliers de développement

- Quatre ateliers
  1 Gestion du cycle de vie du chunker
  2 Développement des relations
  3 Apprentissage des affinités de rattachement
  4 Vérification post-mortem

Les ateliers ne gèrent pas les mêmes informations.

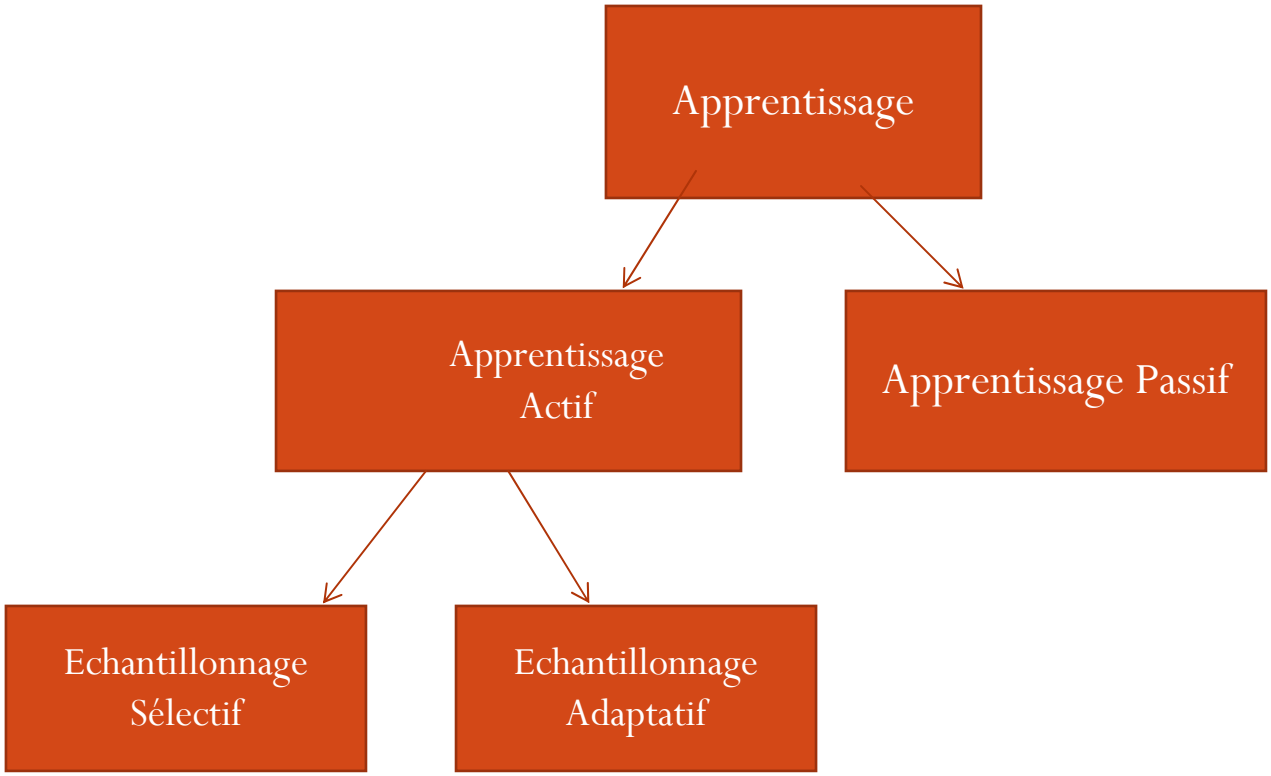
Certains ateliers sont dédiés au rappel et d'autres à la précision

Point de détail: historiquement, les ateliers n'ont pas été conçus après l'analyseur mais en même temps.

Atelier#1 Gestion du cycle de vie du chunker

- L'annotation des données est un pré-requis pour un grand nombre de programmes de TAL. Malheureusement, produire des données fiables en grandes quantités est une tâche laborieuse. Il a été montré que l'apprentissage actif peut être employé pour en réduire le coût sans baisse de qualité.
- Concrètement, il s'agit de combiner un corpus annoté avec un corpus brut par sélection dynamique de fragments.

Apprentissages

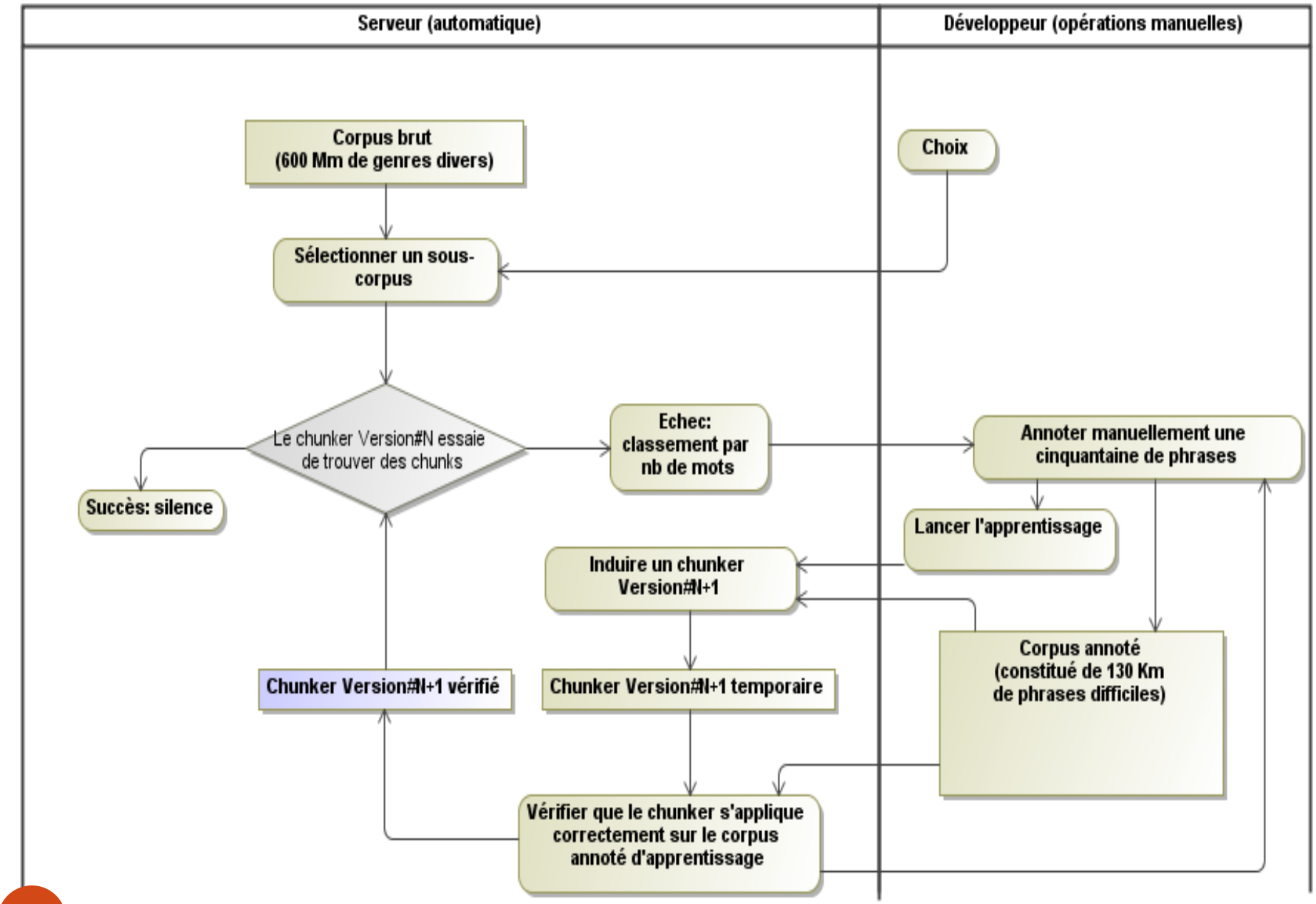


- Voir état de l'art: Bondu (RNTI 2007) , Castro et Nowak (NIPS 2005) ou l'atelier Active Learning NAACL-2009 http://nlp.cs.byu.edu/alnlp

Apprentissage actif: exemple

- A l 'état initial, avec un corpus vide, supposons que nous ajoutions la phrase « Albert marche. ».
- L'annotation donnée « à la main » sera:
  [Albert GN] [marche GV] .
  prénom verbeConjugué ponctuationFinale
- Un programme d'apprentissage peut alors construire automatiquement un chunker qui est capable de reconnaître « Albert marche » mais aussi « Adrienne dort »
- Supposons maintenant que nous confrontions ce chunker à la phrase « Adrienne dort profondément ». Le chunker (qui est lancé sans son mode de rattrapage) échoue puisqu'il n'a, à cette étape, aucune connaissance des adverbes.
- L'échec d'analyse est enregistré dans un fichier. Il faut alors annoter manuellement cette phrase qui va faire grossir le corpus annoté.
- D'un cycle à l'autre, le dispositif permet de construire un chunker du français.
- => dédié principalement au rappel

- Dans la réalité, c'est un peu plus compliqué.
- L'apprentissage est suivi d'une phase de vérification de la cohérence des annotations. Il est vérifié que le chunker induit donne effectivement les mêmes annotations que celles qui figurent dans le corpus annoté.
- Le point le plus délicat de l'apprentissage actif est la sélection des exemples à chaque itération. Nous posons deux hypothèses qui sont que l'effort d'annotation est proportionnel au nb de mots et que plus une phrase en échec est petite plus elle a de chances d'exhiber des phénomènes intéressants rapporté au nb de mots => les échecs sont triés par ordre de longueur croissante ce qui permet de choisir les phrases dont le retour sur investissement (effort rapporté à l'apport au système) est le plus élevé.

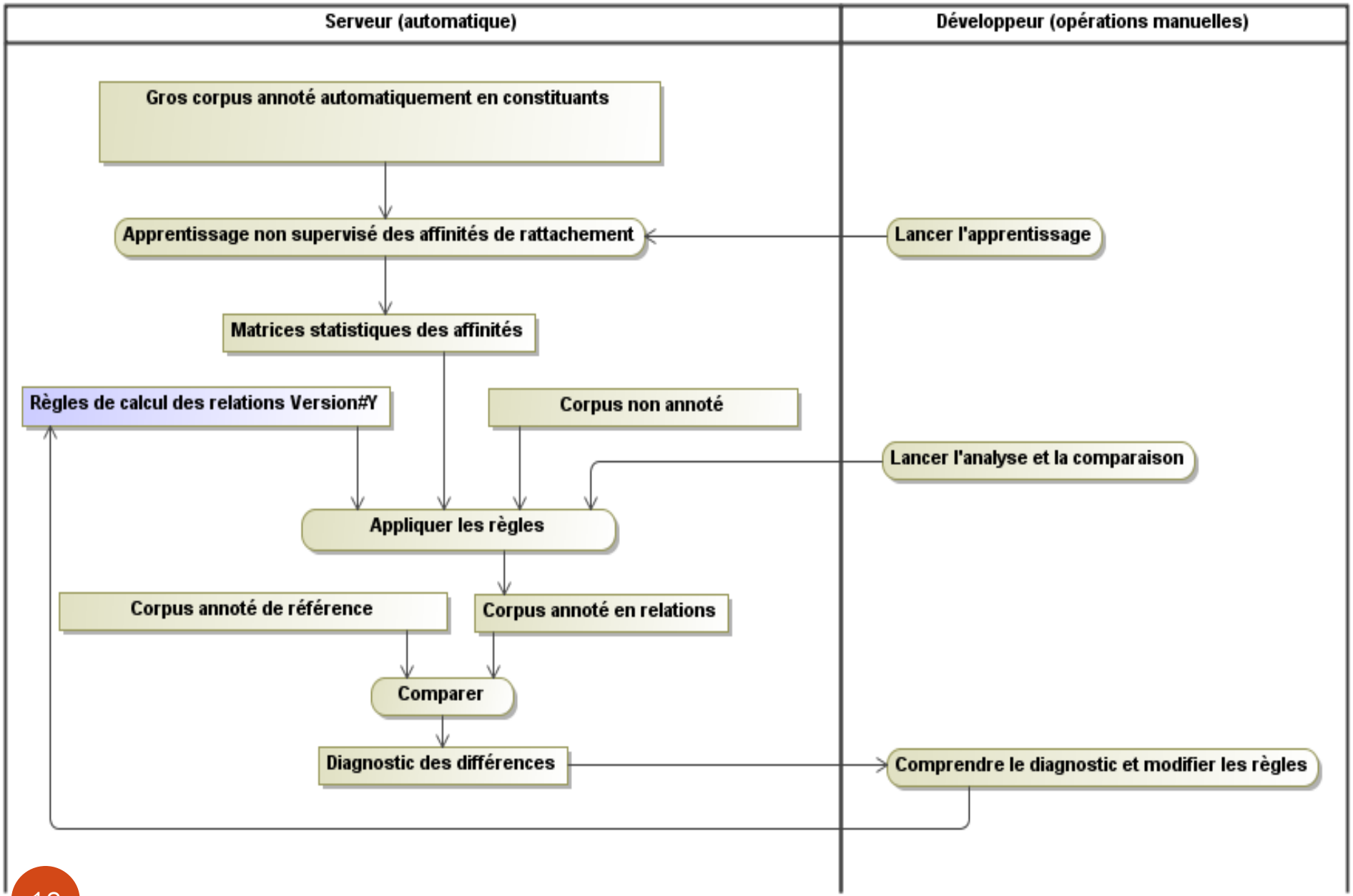

Serveur (automatique)
Développeur (opérations manuelles)
Corpus brut
(600 Mm de genres divers)
Choix
Sélectionner un sous-corpus
Le chunker Version#N essaie de trouver des chunks
Succès: silence
Echec: classement par nb de mots
Annoter manuellement une cinquantaine de phrases
Lancer l'apprentissage
Induire un chunker Version#N+1
Corpus annoté
(constitué de 130 Km de phrases difficiles)
Chunker Version#N+1 vérifié
Chunker Version#N+1 temporaire
Vérifier que le chunker s'applique correctement sur le corpus annoté d'apprentissage

Atelier#2 développement des relations

- Système disons plus « classique »
- La gestion n'est pas effectuée par apprentissage mais par édition manuelle de règles avec contrôle par test de non-régression
- Repose sur DicoValence (Van den Eynde & Mertens), Lefff-3 (Sagot) et des informations décrites par Tagmatica
- => dédié au rappel et à la précision

Atelier#3 Apprentissage d'affinités de rattachement

- Une fois le chunker suffisamment, il est possible de s'intéresser aux rattachements entre groupes.
- On détermine par exemple, quels sont les triplets nom#1 / prep / nom#2 qui figurent dans un corpus dans une situation où l'on peut déterminer de manière certaine que le GP modifie le GN.
- En début de phrase: « Le chien de ma mère … » avec « de ma mère » est le modifieur de « le chien ».
- Un programme d'apprentissage passif non supervisé opère à partir de l'analyse en constituants du gros corpus.
- => dédié à la précision


Serveur (automatique)
Développeur (opérations manuelles)
Gros corpus annoté automatiquement en constituants
Apprentissage non supervisé des affinités de rattachement
Lancer l'apprentissage
Matrices statistiques des affinités
Règles de calcul des relations Version#Y
Corpus non annoté
Lancer l'analyse et la comparaison
Appliquer les règles
Corpus annoté de référence
Corpus annoté en relations
Comparer
Diagnostic des différences
Comprendre le diagnostic et modifier les règles

Atelier#4 vérification post-mortem

- On lance la chaîne d'analyse au complet et on s'assure que la sortie respecte bien certaines propriétés formelles définies dans le guide d'annotation.
- Pas ex, il est vérifié qu'un GP commence effectivement par une prep.
- Il y a une quarantaine de règles.
- Elles sont évaluées périodiquement sur les portions du gros corpus sur lesquelles l'apprentissage actif a été appliqué avec succès.
- => dédié à la précision

Aussi ...

- Remontées d'anomalies par les développeurs et utilisateurs finaux

- sur les résultats de l'analyse syntaxiques, y compris les entités nommées
- depuis les trois post-traitements sémantiques: regroupement de variantes (à la Fastr), résolution des références pronominales, extraction des citations.

Mise en oeuvre

- Totalité du code écrit en Java de manière portable et industrielle sur une période de 7 ans.
- Totalement en UTF8
- Les différents modules sont des packages Java qui s'enchaînent via des appels en mémoire afin d'ajouter un annotation selon LAF. Aucune lecture/écriture sur disque.
- Conception objet avec héritage.
- Débit: ni très lent, ni très rapide: 1,3 M mots par heure

Qualité des résultats

- Sur le site d'évaluation de Elda pour la campagne#1 de l'ANR-Passage
- TagParser avec mode de rattrapage,

	Constituants			Relations		
	F-mesure	Précision	Rappel	F-mesure	Précision	Rappel
juin 2009	0,961	0,956	0,966	0,660	0,671	0,650

- Limites: la limite semble atteinte pour les constituants. Grande marge de progression possible pour les relations.

Merci !!!