Traitement Automatique du Langage Naturel (TALN)

Outils d'analyse de données textuelles

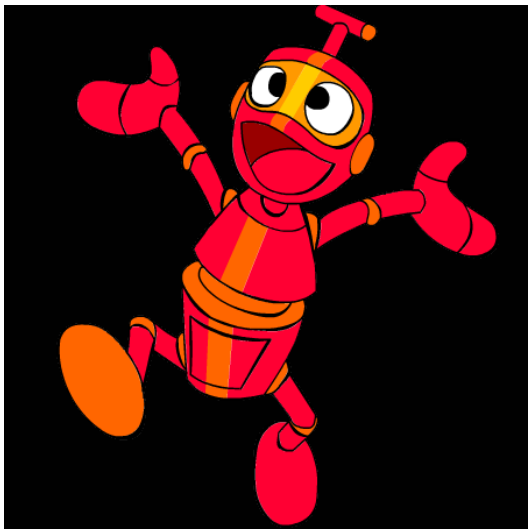
Laurent Audibert (LIPN - UMR CNRS 7030)

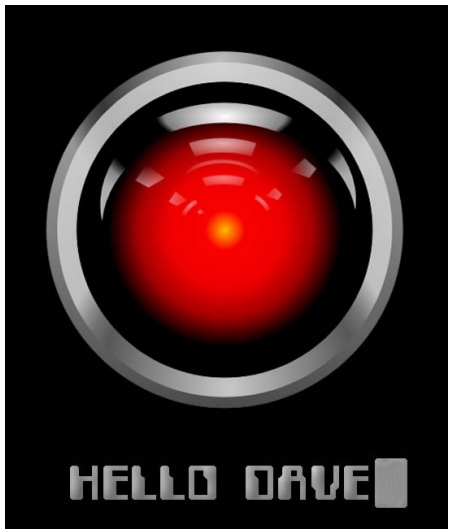
Université Paris 13 – Laboratoire d'Informatique de Paris-Nord (LIPN)

4 novembre 2010

Traitement Automatique du Langage Naturel : Un objectif…

- Objectif : Dialoguer *naturellement*
  avec une machine comme
  avec une personne



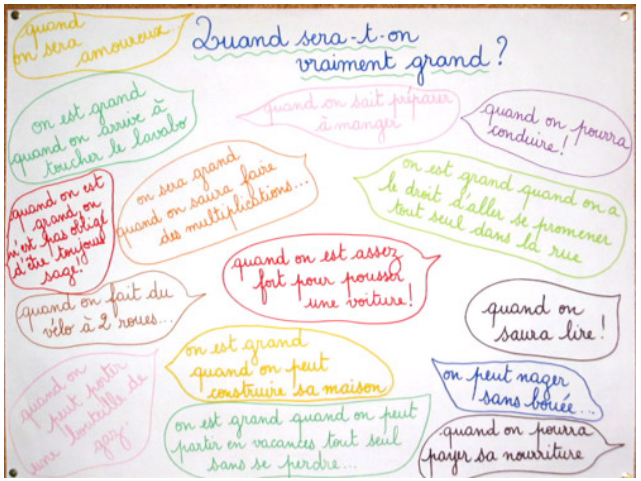


TALN : … difficile à atteindre

- En 1950, Turing prédit que ce sera possible « *dans 50 ans* »
  $\nrightarrow$ L'échéance est passée sans que la prédiction ne se réalise

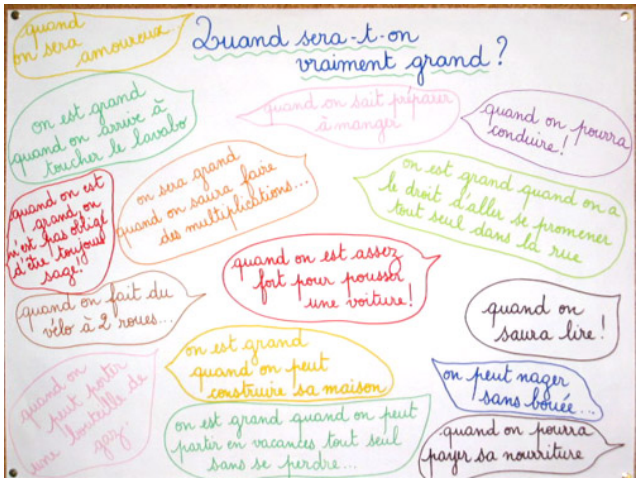
TALN : … difficile à atteindre

- En 1950, Turing prédit que ce sera possible « *dans 50 ans* »
  $\rightarrow\!\!\!\!\rightarrow$ L'échéance est passée sans que la prédiction ne se réalise
- Aujourd'hui les ordinateurs battent les grands maîtres d'échecs, mais n'ont pas les compétences langagières d'un enfant de 5 ans



TALN : … difficile à atteindre

- En 1950, Turing prédit que ce sera possible « *dans 50 ans* »
  $\rightarrow\!\!\!\!\rightarrow$ L'échéance est passée sans que la prédiction ne se réalise
- Aujourd'hui les ordinateurs battent les grands maîtres d'échecs, mais n'ont pas les compétences langagières d'un enfant de 5 ans



Le problème serait-il plus complexe que prévu ?

TALN : Définition

Définition (Traitement Automatique du Langage Naturel (TALN))

On regroupe sous le vocable de traitement automatique du langage naturel (TALN) l'ensemble des recherches et développements visant à modéliser et reproduire, à l'aide de machines, la capacité humaine à produire et à comprendre des énoncés linguistiques dans des buts de communication

- Discipline à cheval entre la linguistique et l'informatique
- Entretient des liens étroits avec les sciences cognitives
- Possède des zones de recouvrement avec l'Intelligence Artificielle

Cette section *Introduction au TALN* s'appuie sur les sources suivantes :
Véronis (2001) ; Tellier (2010) ; Yvon (2010)

Historique du TALN : Travaux en linguistique

1588-1648 - Le père Marin Mersenne [1] aborde l'étude de la phonation, d'un point de vue articulatoire, acoustique et mécanique

1660 - Publication de la *Grammaire générale et raisonnée* (*Grammaire de Port-Royal*) d'Antoine Arnauld et Claude Lancelot décrivant les règles du langage en termes de principes rationnels universels

1700-1900 - Règne de la linguistique comparative et historique

1916 - Ferdinand de Saussure publie son *Cours de linguistique générale*

1930-1940 - Le *cercle de Prague* prolonge les analyses de Saussure et promeut une *linguistique structurale*

1951-1954 - Harris publie ses travaux sur la linguistique distributionnaliste

1957 - L'œuvre de Noam Chomsky marque l'histoire de la syntaxe des 50 dernières années

1. Correspondant de Galilée, Descartes et de nombreux autres savants et philosophes

Historique du TALN : Naissance de l'ordinateur

1936 - Alan Turing publie l'article fondateur de la science informatique

1945 - Von Neumann propose un modèle d'architecture d'ordinateur extrêmement innovant et toujours d'actualité aujourd'hui

1946 - Premier ordinateur ne comportant plus de pièces mécaniques : l'ENIAC (Electronic Numerical Integrator and Computer)

1947 - Invention du transistor, étape décisive vers l'ordinateur moderne (John Bardeen, William Shockley et Walter Brattain, Nobel de physique en 1956)

1955 - Invention du circuit intégré, jetant les bases du matériel informatique moderne (Jack Kilby, Nobel de physique en 2000)

1971-1975 - Premiers ordinateurs personnels : kit MCS4 en 1971 (Intel), Micral en 1972 (entreprise française R2E), Altair 8800 en 1975 (constructeur américain MITS)

Historique du TALN : De l'euphorie des années 1950…

1950 - Essor fulgurant des recherches en traduction automatique (TA)
→ promesses prématurées et exagérées de la *machine à traduire*

1952 - Premier chercheur à temps plein en TA (Bar-Hillel, MIT)

1954 - Première expérience de TA du russe vers l'anglais [2]

1954 - Premiers numéros de la revue Mechanical Translation

1955 - Premier ouvrage sur la TA (Booth et Locke)

1959 - Création de l'Association pour l'étude et le développement de la Traduction Automatique et de la Linguistique Appliquée (ATALA : `http://www.linguist.jussieu.fr/~atala`) [3]

2. Vocabulaire de 250 mots et 6 règles de grammaire
3. Aujourd'hui : Association pour le Traitement Automatique des LAngues

Historique du TALN : … à l'age de raison

1960 - Rapport de Bar-Hillel exposant les énormes difficultés technologiques et linguistiques que pose la traduction

1964 - Le rapport ALPAC (Automatic Language Processind Advisory Committee) établit un constat d'échec des recherches en TA, et conduit à l'arrêt des financements et à la disparition quasi totale des recherches dans le domaine

1975 - Les recherche en TA prennent un nouvel essor sous l'impulsion de la Communauté Européenne avec le développement du système SYSTRAN

Historique du TALN : La linguistique computationnelle

Au cours des années 60, le Traitement Automatique des Langues (TAL) se démarque de la TA sous le nom de linguistique computationnelle

1962 - Création de l'*Association for Computational Linguistics* (ACL)

1965 - Première conférence internationale de linguistique computationnelle biannuelle : Coling

1975 - Création de la revue Computational Linguistics

Traitement Automatique du Langage Naturel : Principales Applications

Quelles sont les applications aujourd'hui utilisées mettant en œuvre les recherches en Traitement Automatique du Langage Naturel ?

Traduction automatique

Traduction automatique

- Intérêt applicatif évident, mais tâche particulièrement difficile
- Qualité actuelle pas exceptionnelle mais suffisante pour être utile
- Plusieurs systèmes de traduction en ligne existent déjà, ex. :
  - Reverso (`http://www.reverso.net`)
  - Babel Fish (`http://fr.babelfish.yahoo.com`)
  - Systran (`http://www.systran.fr/traduction-en-ligne-gratuite`)
  - Google traduction (`http://translate.google.fr`)
- Il est probable que la TA fasse l'objet d'améliorations importantes dans les années à venir

Correction orthographique, grammaticale…

Correction orthographique

- Intégrée à toute application informatique impliquant la rédaction
- Correction basée sur des lexiques
- Ex : traitement de texte, courrier électronique, navigateur Internet (zone de saisie)

Correction grammaticale

- Les meilleures applications fonctionnent bien mais sont payantes
- Actuellement aucune application libre pour le français

Atelier d'aide à la rédaction

- Orthographe, grammaire, style…
- Ex : *Antidote* pour le français

Reconnaissance de caractères (OCR)

Reconnaissance de caractères (OCR pour *Optical Character Recognition*)

- **1929 -** Première machine OCR créée par l'ingénieur allemand Gustav Tauschek
- Domaine actif de recherche en informatique depuis la fin des années 1950
- **TALN** Utilisé dans les post-traitements :
  - Règles linguistiques et contextuelles
  - Dictionnaires de mots, de syllabes, de trigrammes
- De nombreuses applications fonctionnelles libres ou payantes existent
- C'est aussi un service : Société Jouve (cf. `http://www.jouve.fr`)

Reconnaissance de la parole

Reconnaissance de la parole

- Discipline ayant fait des progrès considérables
- Grandes étapes :
  - Segmentation du flux continu de paroles en unités discrètes
  - Identification du phonème correspondant à chaque unité
  - Regroupement des unités pour constituer des mots
  - Prise en compte de la syntaxe pour finaliser le texte écrit
- Logiciels de dictée vocale (*Via Voice*, *Dragon Dictate*…)
- Reconnaissance de la parole ou commande vocale (Reconnaissance vocale de Windows, Systèmes de navigation routière GPS, Smartphone…)
- Prototype Google de sous-titrage automatique de Youtube

Synthèse de la parole

Synthèse de la parole

- Créer de la parole artificielle à partir d'un texte quelconque
- Ces systèmes ont largement franchi le seuil de l'intelligibilité permettant leur utilisation
- Difficultés : désambiguïsation des homographes hétérophones, gestion de la prosodie (intonation, rythme et intensité)…
- Démonstrations en ligne :
  - Acapela Group (`http://www.acapela-group.fr/text-to-speech-interactive-demo.html`)
  - Loquendo (`http://tts.loquendo.com/ttsdemo`)

Reconnaissance, Synthèse et Cie.

Reconnaissance et Synthèse de la parole pour interfaces vocales

- Réservation automatisée de billets (train, avion)
- Téléphonie mobile, messagerie vocale
- Systèmes de renseignements automatisés
- Système d'aiguillage en centre d'appels

Reconnaissance vocale, Traduction automatique et Synthèse vocale

- Language-to-Language Translation (Karlsruhe Institute of Technology, Allemagne)
- Traduction temps réel pour téléphone (Microsoft, stade expérimental)

OCR et Traduction automatique

- Prototype Google d'une fonction en temps réel permettant de prendre en photo un morceau de texte et d'en obtenir la traduction

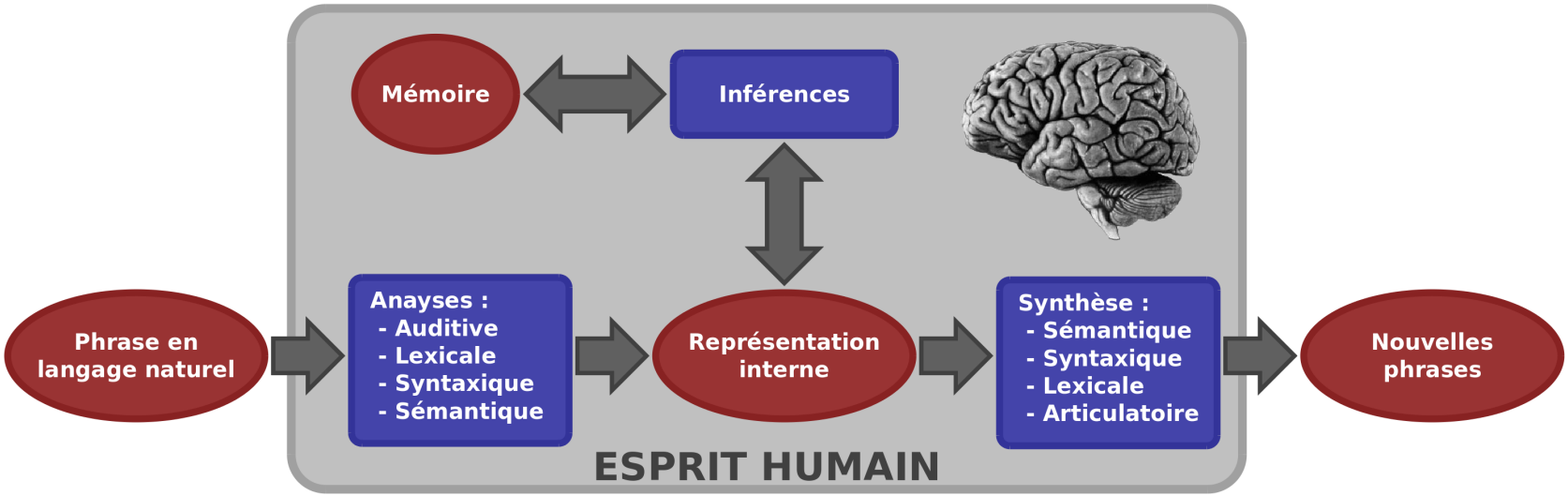
Recherche d'information

Recherche d'information

- Moteurs de recherche :
  - Google (`http://www.google.com`)
  - altavista (`http://fr.altavista.com`)
  - Yahoo ! (`http://fr.yahoo.com`)
  - bing (`http://www.bing.com`)
  - exalead (`http://www.exalead.com`)
  - Wikio (`http://www.wikio.fr`)
  - AlloCiné (`http://www.allocine.fr`)
- Mise en œuvre minimale de technologies du TALN (lemmatisation, détection des expressions composées, thesaurus, réseaux sémantiques…)

Schéma général

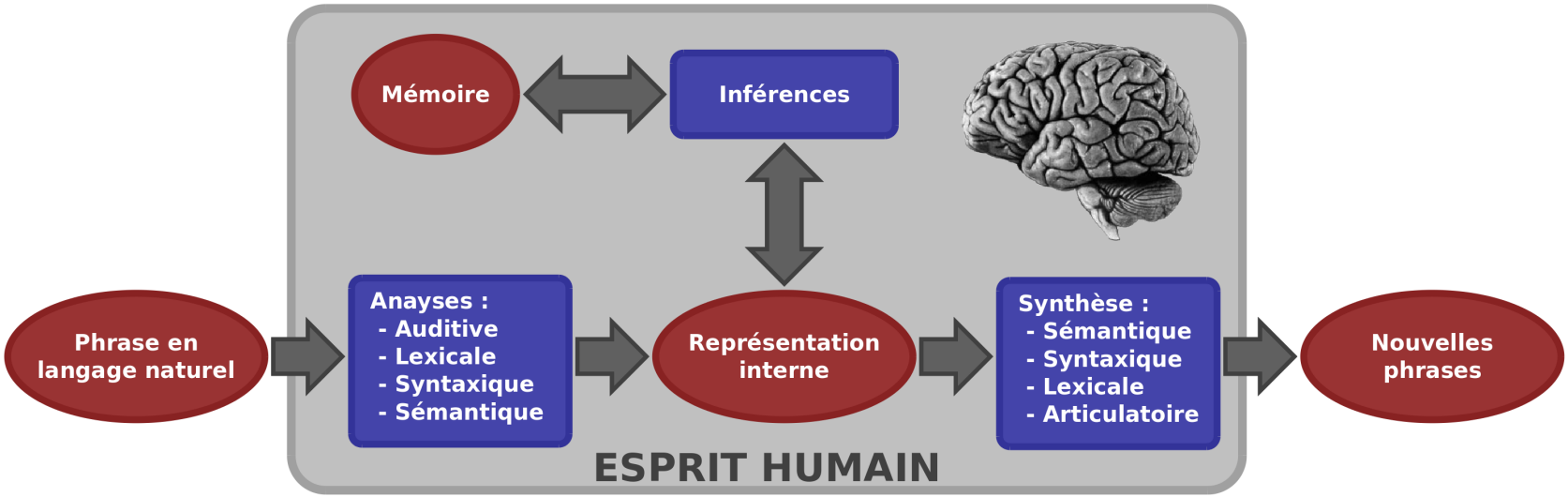
- Chaîne plausible de traitement des langues naturelles par une personne (Psycholinguistique) :



(Schéma très simplifié élaboré par des psychologues cognitivistes)

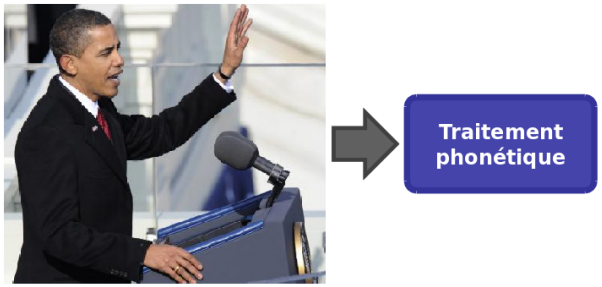
La partie *Niveaux de traitement* de cette section s'appuie sur les sources suivantes : Véronis (2001) ; Tellier (2010)

Schéma général



- Les systèmes de TALN complets tentent de reproduire cette architecture
- Certaines applications ne font intervenir qu'un sous-ensemble de ces traitements

Niveaux de traitement d'une application du TALN



(Architecture séquentielle de la suite des traitements d'une application du TALN)

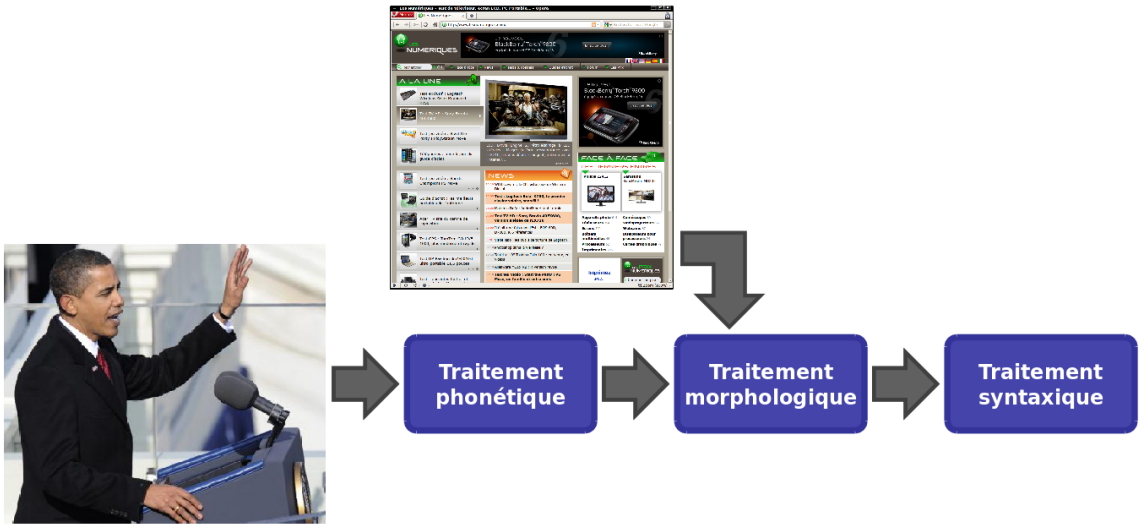
- Dans le cas où l'entrée du système est vocale il faut opérer un traitement phonétique

Niveaux de traitement d'une application du TALN

(Architecture séquentielle de la suite des traitements d'une application du TALN)

- L'étape suivante consiste à déterminer les informations grammaticales associées à chaque mot considéré isolément (traitement morphologique)
- Cette étape est la première si l'entrée du système est textuelle

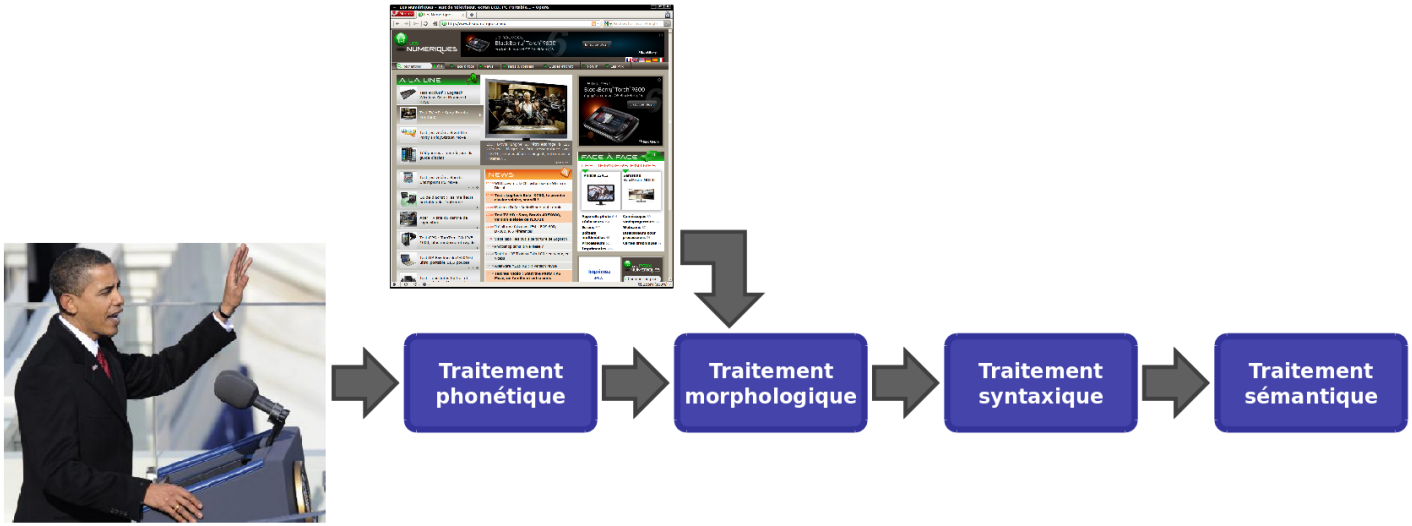
Niveaux de traitement d'une application du TALN



(Architecture séquentielle de la suite des traitements d'une application du TALN)

- Vient ensuite le traitement syntaxique consistant à extraire les relations grammaticales que les mots et groupes de mots entretiennent entre eux

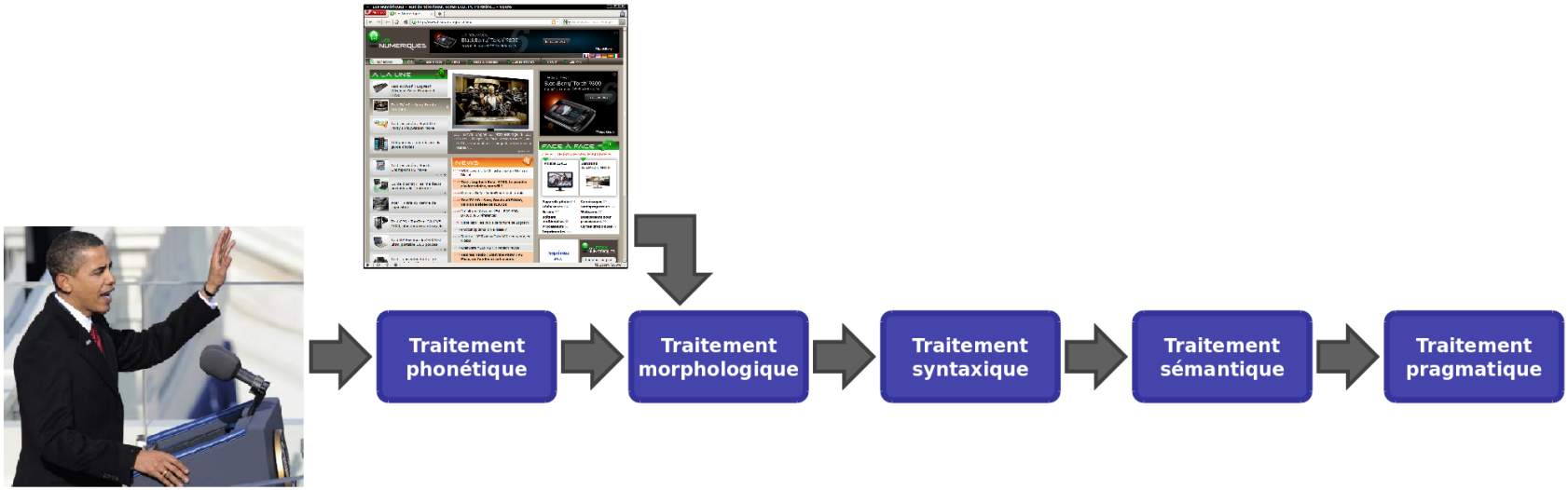
Niveaux de traitement d'une application du TALN



(Architecture séquentielle de la suite des traitements d'une application du TALN)

- Il est ensuite temps d'analyser le sens de la phrase (traitement sémantique)

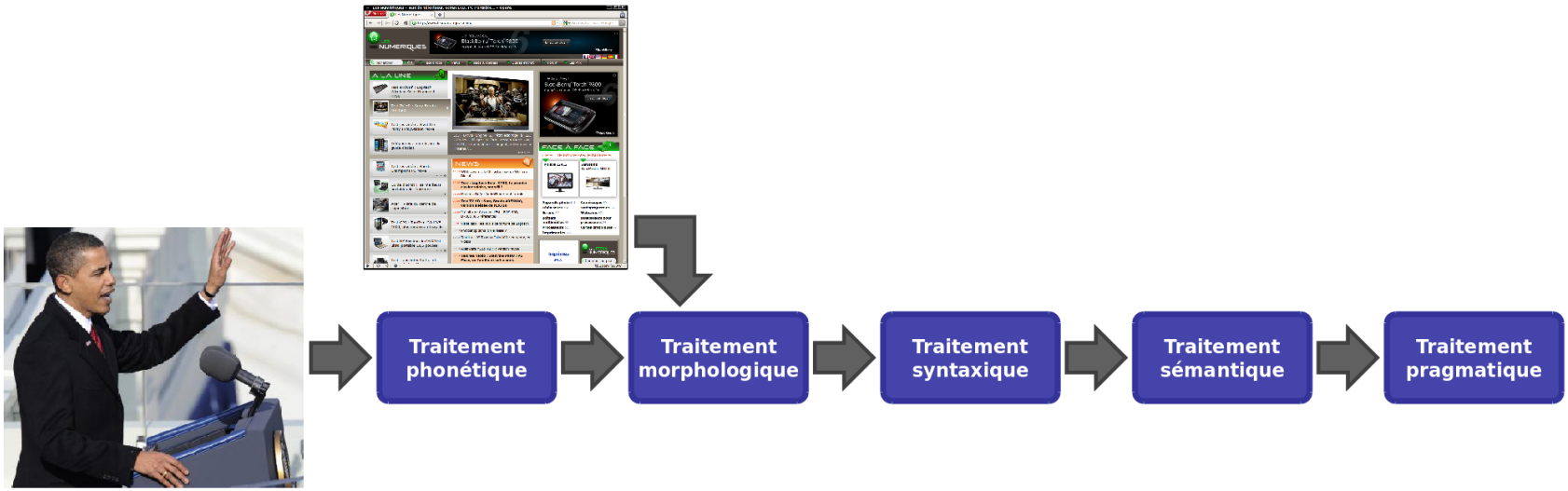
Niveaux de traitement d'une application du TALN



(Architecture séquentielle de la suite des traitements d'une application du TALN)

- Il faut enfin interpréter la phrase en fonction des connaissances générales sur le monde et de la situation de communication (traitement pragmatique)

Niveaux de traitement d'une application du TALN



(Architecture séquentielle de la suite des traitements d'une application du TALN)

- Cette architecture séquentielle est très simpliste car l'ambiguïté de la langue ne permet pas un cloisonnement strict entre ces différents niveaux de traitement

Problème de l'ambiguïté

- Comment déterminer le genre du mot *livre* dans les phrases suivantes :
  1. **J'ai lu un livre**
  2. **Il ne s'agit pas de livres mais de lires**

  $\rightarrow$ par un traitement morphologique ?

Problème de l'ambiguïté

- Comment déterminer le genre du mot *livre* dans les phrases suivantes :
  1. **J'ai lu un livre**
  2. **Il ne s'agit pas de livres mais de lires**

  $\rightarrow$ par un traitement morphologique ?
- Pour la première phrase, il faut repérer que *livre* est précèdé de l'article *un*

  $\rightarrow$ traitement syntaxique !

Problème de l'ambiguïté

- Comment déterminer le genre du mot *livre* dans les phrases suivantes :
  1. **J'ai lu un livre**
  2. **Il ne s'agit pas de livres mais de lires**

  → par un traitement morphologique ?
- Pour la première phrase, il faut repérer que *livre* est précèdé de l'article *un*
  → traitement syntaxique !
- Pour la seconde, il faut intégrer des connaissances sur le monde et la situation de communication (*livre* et *lire* sont deux monnaies)
  → traitement pragmatique ! !

Traitement phonétique

- À partir d'une entrée vocale il faut essentiellement extraire deux informations linguistiques :

  Les phonèmes - sons successifs qui constituent les mots (ex :*chapeau* comprend quatre phonèmes *ch / a / p / eau*)

  La prosodie - intonation, rythme et intensité permettant, par exemple, de distinguer une assertion, une question et une réponse

- Les phonèmes doivent être regroupés pour constituer des mots

Traitement morphologique

Morphologie : étude de la formation des mots à partir d'unités plus petites appelées morphèmes

- Par exemple, le mot lapins est composé de deux morphèmes :
  1. la base ou racine (*lapin*)
  2. un suffixe, la désinence [4] du pluriel (*s*)

Morphème : forme minimum douée de sens, libre ou liée à une autre forme

Morphème lexicaux : ou lexèmes, correspondent grossièrement aux entrées d'un dictionnaire

Morphème grammaticaux : ou affixes (préfixes, suffixes, infixes) n'apparaissent jamais isolés, mais se combinent aux lexèmes

4. Suffixe grammatical servant à la flexion

Traitement morphologique

- Les phénomènes morphologiques se subdivisent en deux groupes :
  la flexion : phénomènes purement grammaticaux (genre, nombre, personne, mode, temps) n'affectant pas la catégorie syntaxique
    - Ex : chat → chats ; chante → chantait

  la dérivation : permet de créer de nouvelles unités lexicales
    - Ex : constituer → constitution → constitutionnel → anticonstitutionnel → anticonstitutionnellement
- Dans un système de TALN, l'analyse morphologique a pour objectif de :
  1. Reconnaître la catégorie syntaxique et les propriétés grammaticales des mots
  2. Proposer une lemmatisation
  3. Reconnaître les entités nommées (noms de personnes, d'organisations, d'entreprises, de lieux, quantités, distances, valeurs, dates…)

Outils : POS Tagger I

TreeTagger

Auteur(s) :	Helmut Schmid
Description :	Étiquetage morphosyntaxique et lemmatisation (Outil probabiliste (HMM) basé sur des arbres de décision
Langue(s) :	Multilingue
Site web :	`http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger`
Licence :	gratuit pour l'évaluation, la recherche et l'enseignement
Plateforme :	Multiplateforme
Langage :	C++
Référence(s) :	Schmid (1994, 1995)
Institution :	Institute for Natural Language Processing (IMS), Universite de Stuttgart

Outils : POS Tagger II

LIA Tools : LIA_TAGG

Auteur(s) :	Frédéric Béchet
Description :	Étiquetage morphosyntaxique et lemmatisation
Langue(s) :	Français
Site web :	`http://pageperso.lif.univ-mrs.fr/~frederic.bechet/download_fred.htm`
Licence :	GPL
Plateforme :	Multiplateforme
Institution :	Laboratoire d'Informatique Fondamentale (LIF), Université Aix Marseille

Outils : POS Tagger III

Stanford POS Tagger

Description :	Maximum-entropy (CMM) part-of-speech (POS) tagger
Langue(s) :	Anglais, Arabe, Chinois, Allemand
Site web :	`http://www-nlp.stanford.edu/software/tagger.shtml`
Licence :	GPLv2
Plateforme :	Multiplateforme
Langage :	Java
Référence(s) :	Toutanova et Manning (2000) ; Toutanova, Klein, Manning, et Singer (2003)
Institution :	Stanford NLP Group, Université de Stanford

Outils : POS Tagger IV

Illinois Part of Speech Tagger

Auteur(s) :	Nick Rizzolo
Description :	This is a state of the art NER tagger that tags plain text with named entitites (people / organizations / locations / miscellaneous)
Langue(s) :	Anglais
Site web :	`http://cogcomp.cs.illinois.edu/page/software_view/3`
Licence :	Gratuit
Plateforme :	Multiplateforme
Langage :	Java
Institution :	Cognitive Computation Group (CCG), University of Illinois

Outils : Reconnaissance des entités nommées I

LIA Tools : LIA_NE

Auteur(s) :	Frédéric Béchet
Description :	Reconnaissance des entités nommées
Langue(s) :	Français
Site web :	`http://pageperso.lif.univ-mrs.fr/~frederic.bechet/download_fred.htm`
Licence :	GPL
Plateforme :	Multiplateforme
Institution :	Laboratoire d'Informatique Fondamentale (LIF), Université Aix Marseille

Outils : Reconnaissance des entités nommées II

Stanford Named Entity Recognizer

Description :	Conditional Random Field sequence model, together with well-engineered features for Named Entity Recognition
Langue(s) :	Anglais
Site web :	`http://www-nlp.stanford.edu/software/CRF-NER.shtml`
Licence :	GPLv2
Plateforme :	Multiplateforme
Langage :	Java
Référence(s) :	Finkel, Grenager, et Manning (2005)
Institution :	Stanford NLP Group, Université de Stanford

Outils : Reconnaissance des entités nommées III

Illinois Named Entity Tagger

Auteur(s) :	Lev Ratinov
Description :	This is a state of the art NER tagger that tags plain text with named entitites (people / organizations / locations / miscellaneous)
Langue(s) :	Anglais
Site web :	`http://cogcomp.cs.illinois.edu/page/software_view/4`
Licence :	Gratuit
Plateforme :	Multiplateforme
Langage :	Java
Référence(s) :	Ratinov et Roth (2009)
Institution :	Cognitive Computation Group (CCG), University of Illinois

Traitement syntaxique

- Comme les mots, les constituants regroupant plusieurs mots (*syntagme*) ont leurs propres catégories (syntagme nominal, syntagme verbal, syntagme prépositionnel…)

  Syntagme : mot ou une suite de mots consécutifs auxquels on peut associer une catégorie syntaxique
- Il existe plusieurs types de syntagmes :

  Les chunks - plus petites séquences de mots auxquelles on peut associer une catégorie

  Les termes - élément lexical, mot ou expression symbolisant un concept dans un domaine de spécialité (ex : *maladie de la vache folle*) et pouvant, par exemple, servir de mot clé dans une indexation

  Les clauses - séquences de mots contenant au moins un sujet et un prédicat

Traitement syntaxique

Syntaxe : étude des règles qui régissent la combinaison des mots en phrases. L'objectif de ces règles est de prédire :

1. la nature des constituants de la phrase
2. la structure hiérarchique de ces constituants
3. leurs fonctions syntaxiques

- Dans un système de TALN, l'analyse syntaxique peut avoir pour objectif de :
  1. Découper la phrase en syntagmes (chunks ou autres constituants)
  2. Reconnaître les termes
  3. Repérer des dépendances syntaxiques
  4. Proposer une organisation hiérarchique des syntagmes

Outils : Extracteur de termes I

YaTeA

Auteur(s) :	Thierry Hamon
Description :	YaTeA (Yet Another Term ExtrActor) permet d'identifier dans un corpus des groupes nominaux qui peuvent correspondre à des termes (c.-à-d. des candidats de terme)
Langue(s) :	Français, Anglais
Site web :	`http://search.cpan.org/~thhamon/Lingua-YaTeA-0.5`
Licence :	GPL
Plateforme :	Linux
Langage :	Perl
Référence(s) :	Aubin et Hamon (2006a, 2006b)
Institution :	Laboratoire d'Informatique de Paris-Nord, Université de Paris 13

Outils : Extracteur de termes II

ACABIT

Auteur(s) :	Béatrice Daille
Description :	ACABIT est un programme d'acquisition de terminologie qui prend en entrée un texte annoté linguistiquement et retourne une liste ordonnée de candidats termes.
Langue(s) :	Français, Anglais
Site web :	`http://www.bdaille.fr/index.php?option=com_content&task=blogcategory&id=5&Itemid=5`
Licence :	GPL
Langage :	Perl
Référence(s) :	Daille (2003)
Institution :	Laboratoire d'Informatique de Nantes Atlantique (LINA)

Outils : Extracteur de termes III

FASTR

Auteur(s) :	Christian Jacquemin
Description :	FASTR est un analyseur syntaxique permettant de reconnaître en corpus des variation terminologiques
Langue(s) :	Français, Anglais
Site web :	`http://www.limsi.fr/Individu/jacquemi/FASTR`
Licence :	GPL
Plateforme :	Linux
Référence(s) :	Jacquemin (1997) ; Jacquemin, Klavans, et Tzoukermann (1997)
Institution :	Laboratoire d'Informatique pour la Mécanique et les Sciences de l'Ingénieur (LIMSI), Universités UPMC et Paris-Sud 11

Outils : Extracteur de termes IV

TermExtractor

Description :	Extracteur de termes
Langue(s) :	Anglais
Site web :	`http://lcl2.di.uniroma1.it/termextractor`
Licence :	Gratuit
Référence(s) :	Sclano et Velardi (2007)
Institution :	Linguistic Computing Laboratory (LCL), Computer Science Department of the University of Roma

Outils : Extracteur de termes V

Termostat

Auteur(s) :	Patrick Drouin
Description :	TermoStat est un outil d'acquisition automatique de termes qui exploite une méthode de mise en opposition de corpus spécialisés et non-spécialisés en vue de l'identification des termes
Langue(s) :	Anglais, Français, Espagnol, Italien
Site web :	`http://olst.ling.umontreal.ca/~drouinp/termostat_web`
Institution :	Département de linguistique et de traduction, Université de Montréal

Outils : Analyseur Syntaxique I

Link Grammar

Auteur(s) :	Davy Temperley, Daniel Sleator, John Lafferty
Description :	Analyseur syntaxique basé sur une grammaire de dépendances
Langue(s) :	Anglais
Site web :	`http://www.link.cs.cmu.edu/link/`
Licence :	Compatible GPL
Plateforme :	Multiplateforme
Langage :	C
Référence(s) :	Sleator et Temperley (1993) ; Lafferty, Sleator, et Temperley (1992) ; Grinberg, Lafferty, et Sleator (1995)
Institution :	Université de Carnegie Mellon

Outils : Analyseur Syntaxique II

SYNTEX

Auteur(s) :	Didier Bourigault
Description :	SYNTEX est un analyseur procédural à cascade
Langue(s) :	Français
Site web :	`http://w3.erss.univ-tlse2.fr/textes/pagespersos/bourigault/syntex.html`
Licence :	Payant
Référence(s) :	Bourigault D. (2000)
Institution :	Traitement automatique des langues, Université Toulouse le Mirail

Outils : Analyseur Syntaxique III

C&C tools

Auteur(s) :	James Curran, Stephen Clark, Johan Bos
Description :	The C&C tools consist of a robust, wide-coverage Combinatory Categorial Grammar (CCG) parser and a number of Maximum Entropy taggers (pos tagging, chunking, named entity recognition), each of which can be run as a separate program, or combined in one go
Langue(s) :	Anglais
Site web :	`http://svn.ask.it.usyd.edu.au/trac/candc/wiki`
Licence :	Licence Universitaire
Plateforme :	Multiplateforme
Langage :	C++

Outils : Analyseur Syntaxique IV

Enju

Description :	Enju is a syntactic parser for English with a wide-coverage probabilistic HPSG grammar and an efficient parsing algorithm. This parser can effectively analyze syntactic/semantic structures of English sentences and provide a user with phrase structures and predicate-argument structures. Online demo is available ! UIMA Web Interface for Enju is also available. You can embed Enju in UIMA workflows.
Langue(s) :	Anglais
Site web :	`http://www-tsujii.is.s.u-tokyo.ac.jp/enju`
Plateforme :	Linux, Mac OS
Institution :	The University of Tokyo, Department of Computer Science, Tsujii laboratory

Objectifs de l'annotation linguistique

Information non structurée (*Unstructured Information*)

Source d'information (texte, audio, image ou vidéo) difficilement exploitable en raison de la nature non structurée du contenu

- Application de gestion de l'information non structurée (*Unstructured Information Management*, UIM) :
  $\rightarrow$ Organiser un grand volume d'information non structurée pour extraire, structurer et diffuser l'information

Objectifs de l'annotation linguistique

Information non structurée (*Unstructured Information*)

Source d'information (texte, audio, image ou vidéo) difficilement exploitable en raison de la nature non structurée du contenu

- Application de gestion de l'information non structurée (*Unstructured Information Management*, UIM) :
  $\rightarrow$ Organiser un grand volume d'information non structurée pour extraire, structurer et diffuser l'information

Objectifs de l'annotation linguistique

Enrichir les documents en associant des informations linguistiques aux différents segments de texte

(Par exemple avec les outils mentionnés dans la section précédente)

Objectifs de l'annotation linguistique

Information non structurée (*Unstructured Information*)

Source d'information (texte, audio, image ou vidéo) difficilement exploitable en raison de la nature non structurée du contenu

- Application de gestion de l'information non structurée (*Unstructured Information Management*, UIM) :
  $\rightarrow$ Organiser un grand volume d'information non structurée pour extraire, structurer et diffuser l'information

Objectifs de l'annotation linguistique

Enrichir les documents en associant des informations linguistiques aux différents segments de texte

(Par exemple avec les outils mentionnés dans la section précédente)

Problème : les outils développés sont hétérogènes et non interopérables

Pourquoi des plateformes d'annotations linguistiques ?

La recherche en TALN implique souvent :

- L'utilisation de corpus de grande taille et de formats variés
- Des cycles de modélisation/expérimentation/évaluation courts
- L'articulation d'outils et de ressources hétérogènes
- La réutilisation/capitalisation des outils et des ressources
- La sophistication des modèles linguistiques mis en œuvre

$\rightarrow$ Utilisation d'environnements de développement dédiés au TALN

$\rightarrow$ Floraison de plateformes plus ou moins génériques d'annotations

Cf. (Enjalbert, 2008)

Objectif des plateformes d'annotations linguistiques

Objectif des plateformes d'annotations linguistiques

Permettre l'intégration des outils existants et faciliter
leur interopérabilité par des mécanismes d'encapsulation

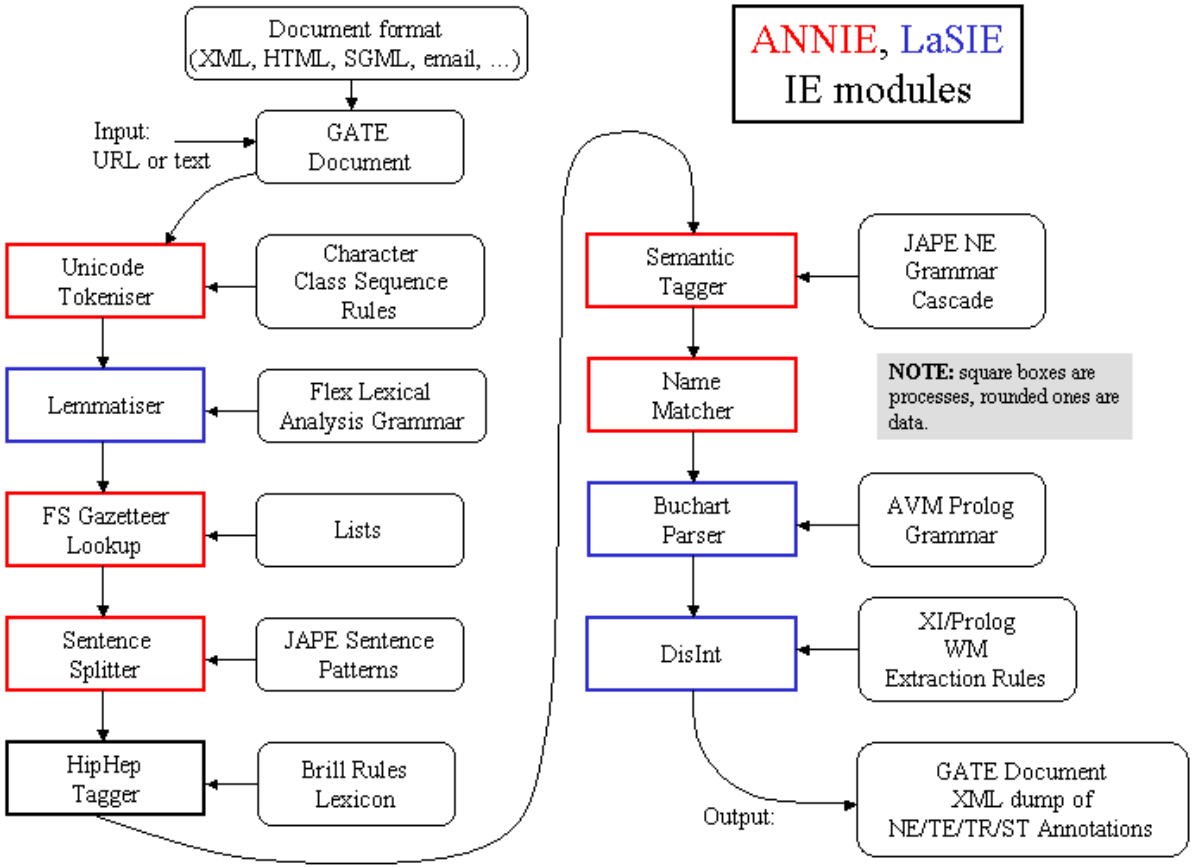
- Plateformes d'annotations linguistiques = chaîne de traitement
  $\rightarrow$ permettre l'enchaînement des traitements
- Réutilisation d'outils du TALN existants
  (Traitement $\approx$ encapsulation d'un outil du TALN)
- Résoudre les problèmes d'interopérabilité d'outils hétérogènes
- Générique et modulaire pour faciliter :
  - l'utilisation de nouveaux outils
  - l'interchangeabilité des outils

GATE (*General Architecture for Text Engineering*)

- Plateforme développée depuis 1995 à l'Université de Sheffield
- Infrastructure permettant le développement et le déploiement de composants pour le TALN
- GATE propose
  - une architecture
  - un framework en Java (incluant de nombreux modules)
  - un environnement de développement autonome

Institution :	Université de Sheffield
Site web :	`http://gate.ac.uk`
Licence :	open source
Plateforme :	Multiplateforme
Langage :	Java
Référence(s) :	Cunningham, Maynard, Bontcheva, et Tablan (2002)

GATE : pipeline de modules autonomes



- Fonctionne comme pipeline purement linéaire de modules autonomes

GATE : éléments de base

Language Ressource (LR) : Lexiques, taxonomies, ontologies, corpus et autres ressources (support de plusieurs formats XML, RTF, HTML, SGML…)

Processing Ressources (PR) : algorithmes et composants effectuant un traitement ayant pour but d'ajouter ou de transformer des annotations (sous forme d'attributs/valeurs)

Application ou contrôleur : agrégation de plusieurs PR sous forme de pipeline

Visual Ressource : permet la présentation des résultats à l'intérieur de l'environnement de développement GATE

Plugins : composition de plusieurs LR et PR spécialisés dans l'exécution d'une tâche précise

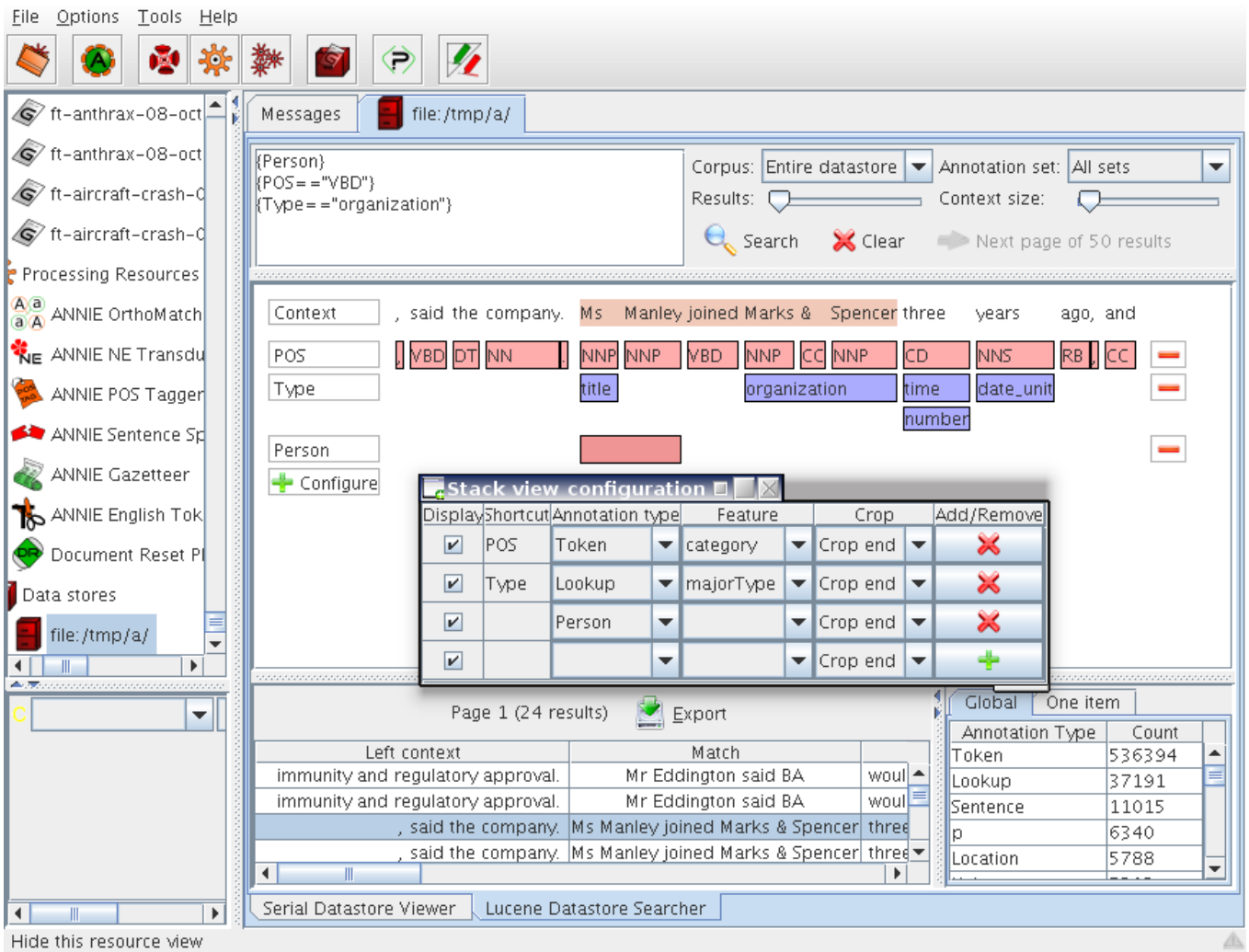
CREOLE (Collection of REusable Objects for Language Engineering) : large inventaire de ressources (LR, PR…) qui fonde GATE

GATE (*General Architecture for Text Engineering*)

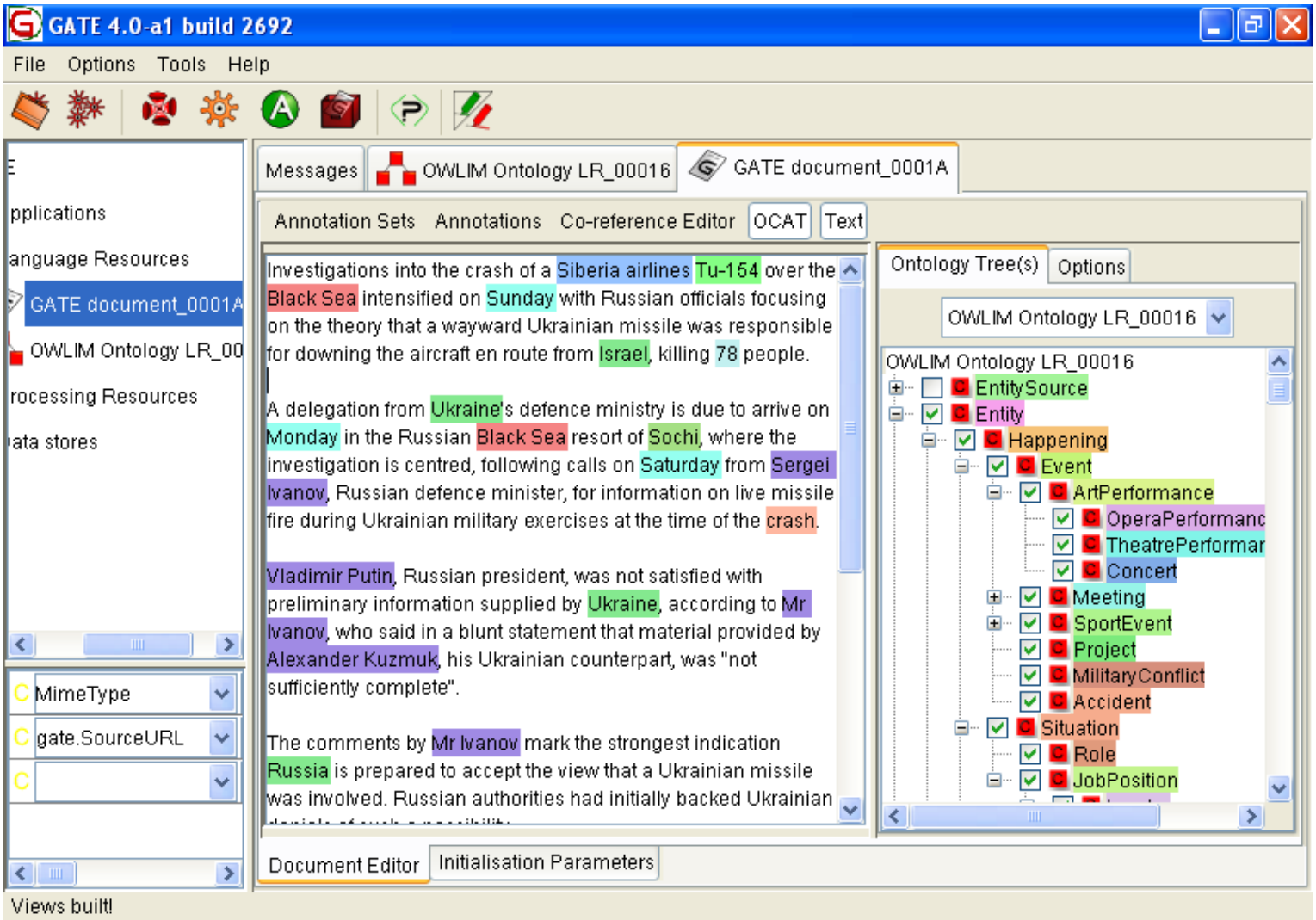
- Documentation importante :

  `http://gate.ac.uk/documentation.html`
- La plateforme GATE fournit :
  - Un adaptateur permettant d'utiliser sous forme de PR (*Processing Resource*) dans GATE un TAE (*Text Analysis Engine*) primitif ou composé d'UIMA
  - Un adaptateur permettant d'utiliser sous forme de TAE dans UIMA un pipeline de traitement de GATE (*CorpusController*)

GATE : capture d'écran



GATE : capture d'écran

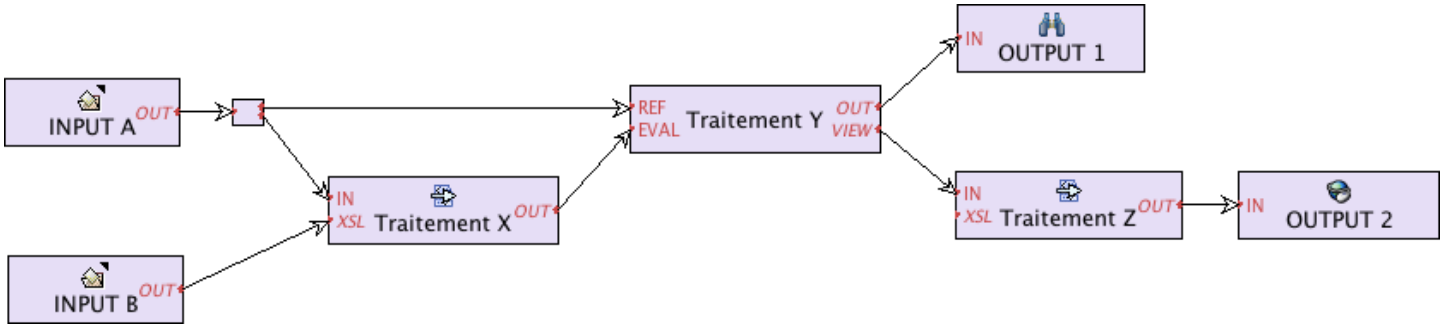


LinguaStream

- Développé au GREYC depuis 2001
- Repose sur le principe d'enrichissement incrémental des documents
- Modèle séquentiel de chaînes de traitements plus puissant que les pipelines linéaires de GATE, proche du système de flux d'UIMA
- Interopérabilité assurée par l'usage d'une représentation unifiée des annotations : structures de traits (Cf. Roussanaly, 2003 ; Véronis, 2001, p. 131)

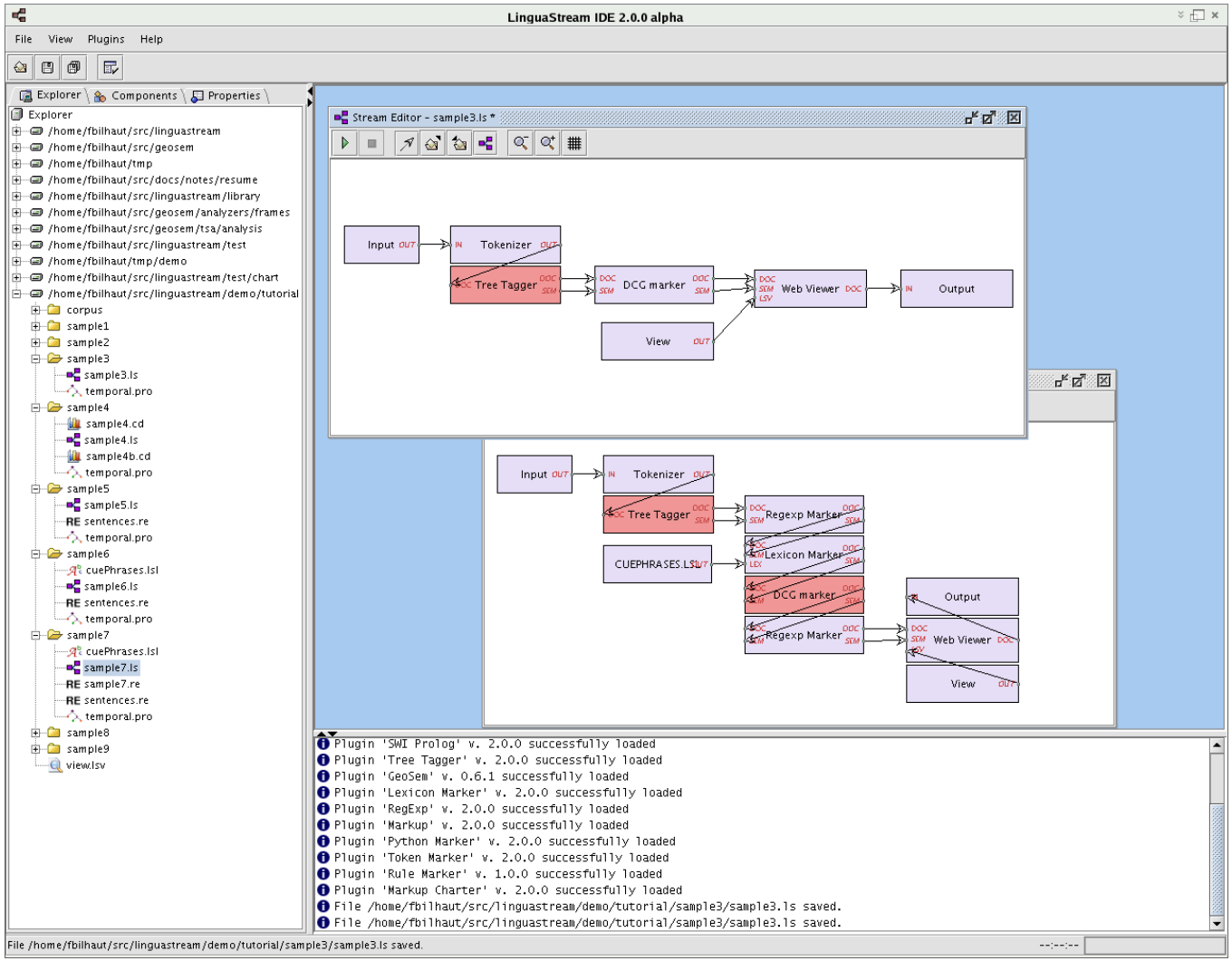
Institution :	Groupe de Recherche en Informatique, Image, Automatique et Instrumentation de Caen (GREYC)
Site web :	`http://linguastream.org`
Licence :	gratuit pour chercheurs et universitaires
Plateforme :	Multiplateforme
Langage :	Java
Référence(s) :	Bilhaut (2003) ; Widlöcher et Bilhaut (2008)

LinguaStream

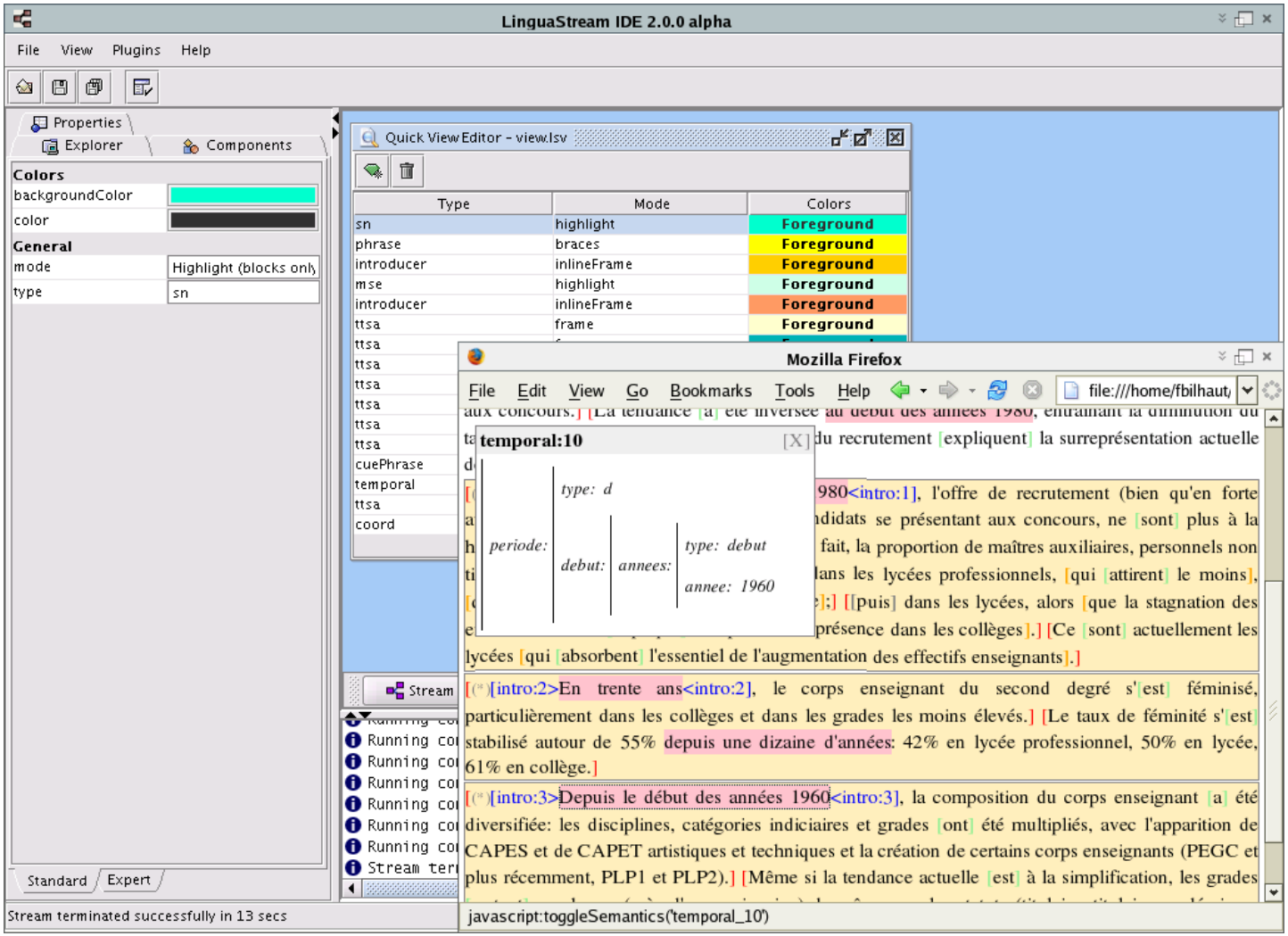


- Propose différents méta-modèles d'analyse (expressions régulières, grammaire locale d'unification, transducteur déterministe, grammaire de contraintes…)
- Environnement de développement intégré permettant une construction de la chaîne de traitement par assemblage visuel des composants (programmation non indispensable)
- Palette d'une cinquantaine de composants intégrés en standard
- Facilités d'intégration de composants exogènes

LinguaStream : Construction d'une chaîne de traitement



LinguaStream : Annotations sous forme de structures de traits



Intex, Nooj et Unitex

- Plateformes mettant en œuvre des modèles à base d'automates enchaînés en cascades
- Intex a été developpé par Max Silberztein de 1993 à 2002
- Depuis 2002, Max Silberztein travaille sur Nooj, le successeur d'Intex
- Unitex est au cœur d'une polémique (Condé & Viprey, 2002 ; MSHE, 2002) :
  - Unitex est un *portage partiel* d'Intex réalisée sans le consentement de son auteur
  - Unitex a été développé en 2002 pour pallier le fait qu'Intex n'était pas *open source*
- Documentation complète et communauté d'utilisateurs importante

Intex

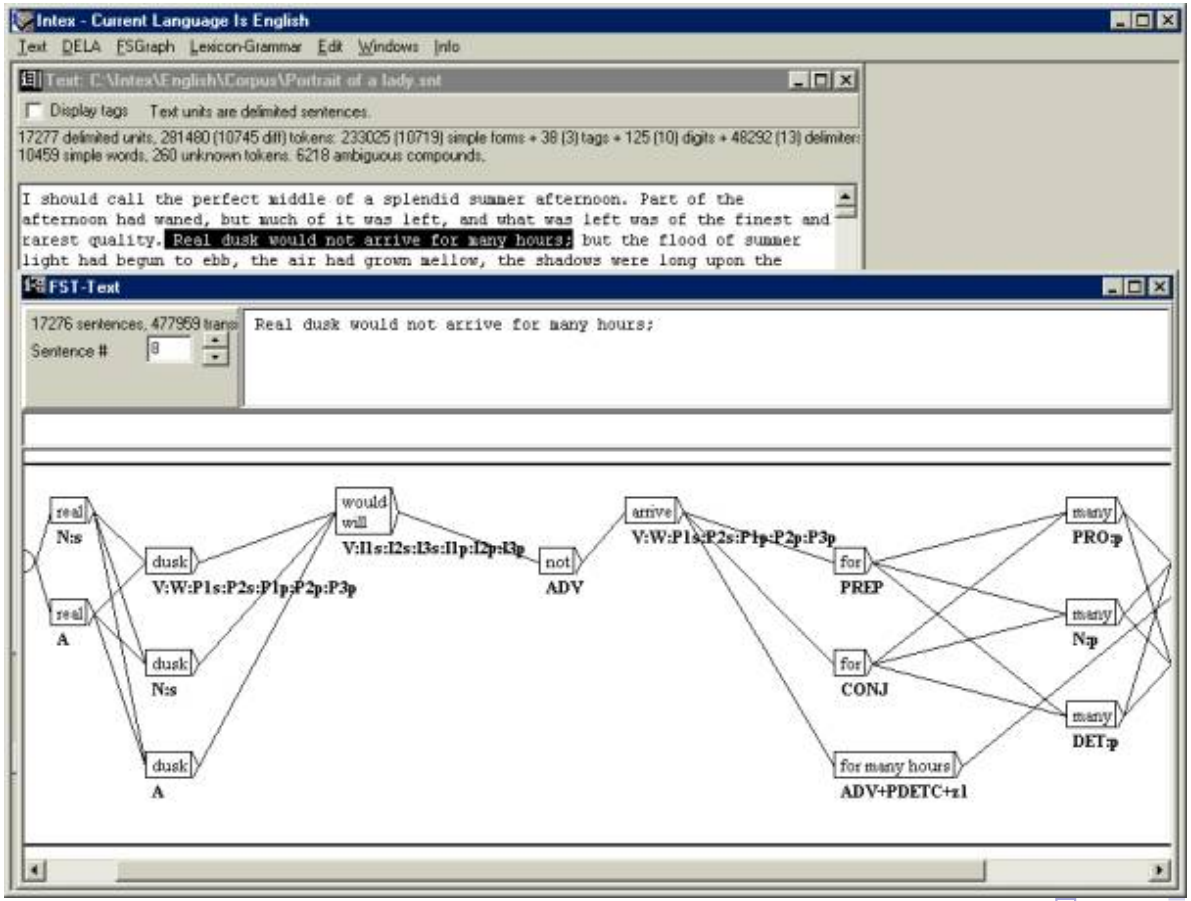
- Developpé par Max Silberztein de 1993 à 2002 [5] au LADL (Université Paris 7) puis à l'Université de Franche-Comté
- Permet de construire des descriptions formalisées des langues naturelles en s'appuyant sur :
  - des dictionnaires électroniques
  - des grammaires représentées par des graphes à états finis
  - des lexiques-grammaires

Auteur(s) :	Max Silberztein
Institution :	Université de Franche-Comté
Site web :	`http://intex.univ-fcomte.fr`
Licence :	freeware pour les étudiants et chercheurs affiliés à une université
Plateforme :	Windows
Langage :	C
Référence(s) :	Silberztein (1993)

5. Depuis 2002, Max Silberztein travaille sur Nooj, le successeur d'Intex

Intex : texte représenté par un transducteurs à états finis

- Tous les objets traités (textes, dictionnaires, grammaires) sont à un moment ou à un autre représentés par des transducteurs à états finis



Nooj

- Permet de formaliser cinq niveaux de phénomènes linguistiques
  1. Orthographe (machines à états finis)
  2. Morphologie
  3. Lexique
  4. Syntaxe (grammaires hors contexte)
  5. Sémantique (réseaux de transition augmentés (*Augmented Transition Networks* ou ATN)
- Nombreux séminaires, ateliers et workshop (Nooj et Intex)

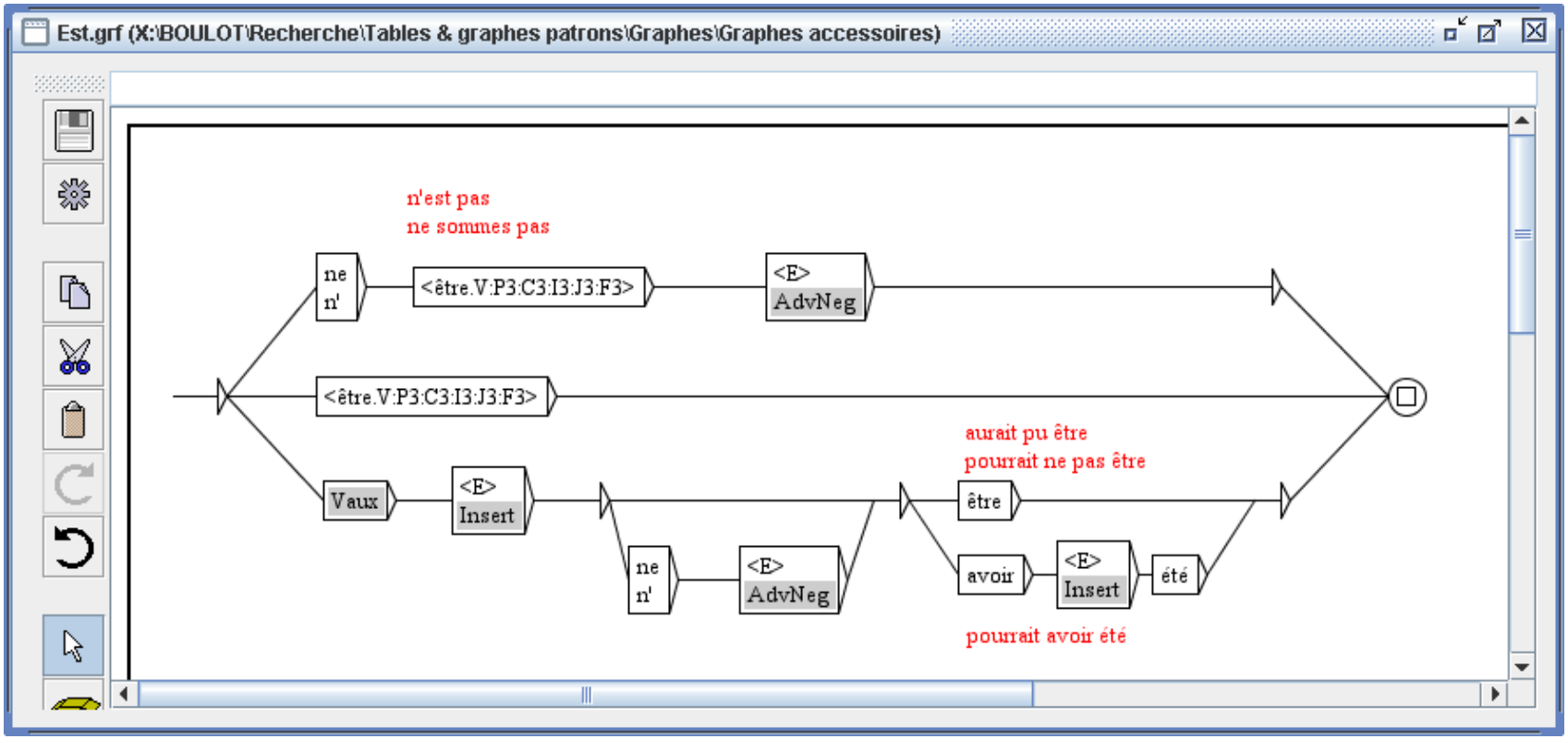
Auteur(s) :	Max Silberztein
Institution :	Université de Franche-Comté
Site web :	`http://www.nooj4nlp.net`
Licence :	freeware
Plateforme :	Windows (Microsoft .NET)
Langage :	C#
Référence(s) :	Silberztein (2003)

Unitex

- Développé en 2002 afin de reproduire les fonctionnalités d'Intex qui n'était pas *open source*
- Unitex est principalement développé par Sébastien Paumier à l'Institut Gaspard-Monge (IGM)

Auteur(s) :	Sébastien Paumier
Institution :	Institut Gaspard-Monge (IGM),Université de Paris-Est Marne-la-Vallée
Site web :	`http://igm.univ-mlv.fr/~unitex`
Licence :	Open source (LGPL)
Plateforme :	Multiplateforme
Langage :	Java, C/C++
Référence(s) :	Paumier (2010)

Unitex : capture d'écran



Antelope

- Plateforme industrielle développée par la société Proxem
- Permet de mettre en œuvre des analyses de niveau lexical, syntaxique et sémantique
- Encore en cours de développement mais d'ores et déjà utilisable

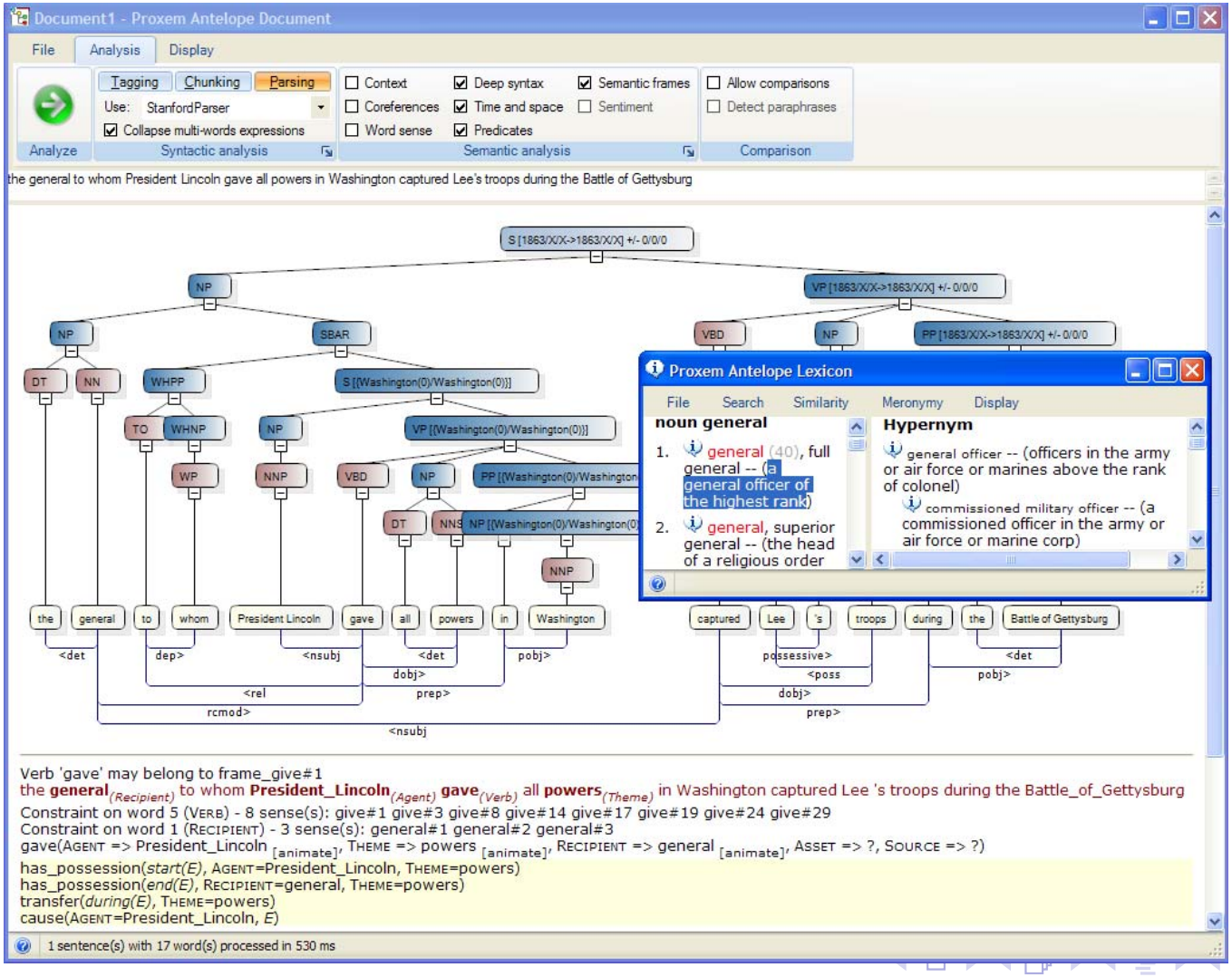
Institution :	Société Proxem
Site web :	`http://www.proxem.com`
Licence :	Disponible sans contrainte pour la recherche et l'enseignement
Plateforme :	Windows (Microsoft .NET)
Langage :	C#
Référence(s) :	Chaumartin (2008)

Antelope : fonctionnalités

Antelope comprend, entre autres, les fonctionnalités suivantes :

- Taggers (étiqueteurs morphosyntaxiques) et chunkers
- Analyseurs syntaxiques en dépendance pour l'anglais (Link Grammar, Stanford Parser) et le français (TagParser)
- Lexique sémantique basé sur WordNet
- Résolveur d'anaphores
- Détecteur de compléments circonstanciels de temps et d'espace
- Module de désambiguïsation lexicale
- Module d'extraction de contexte
- Module d'extraction de la syntaxe profonde
- Étiqueteur de rôles sémantiques (interface syntaxe / sémantique basée sur VerbNet)
- …

Antelope : capture d'écran

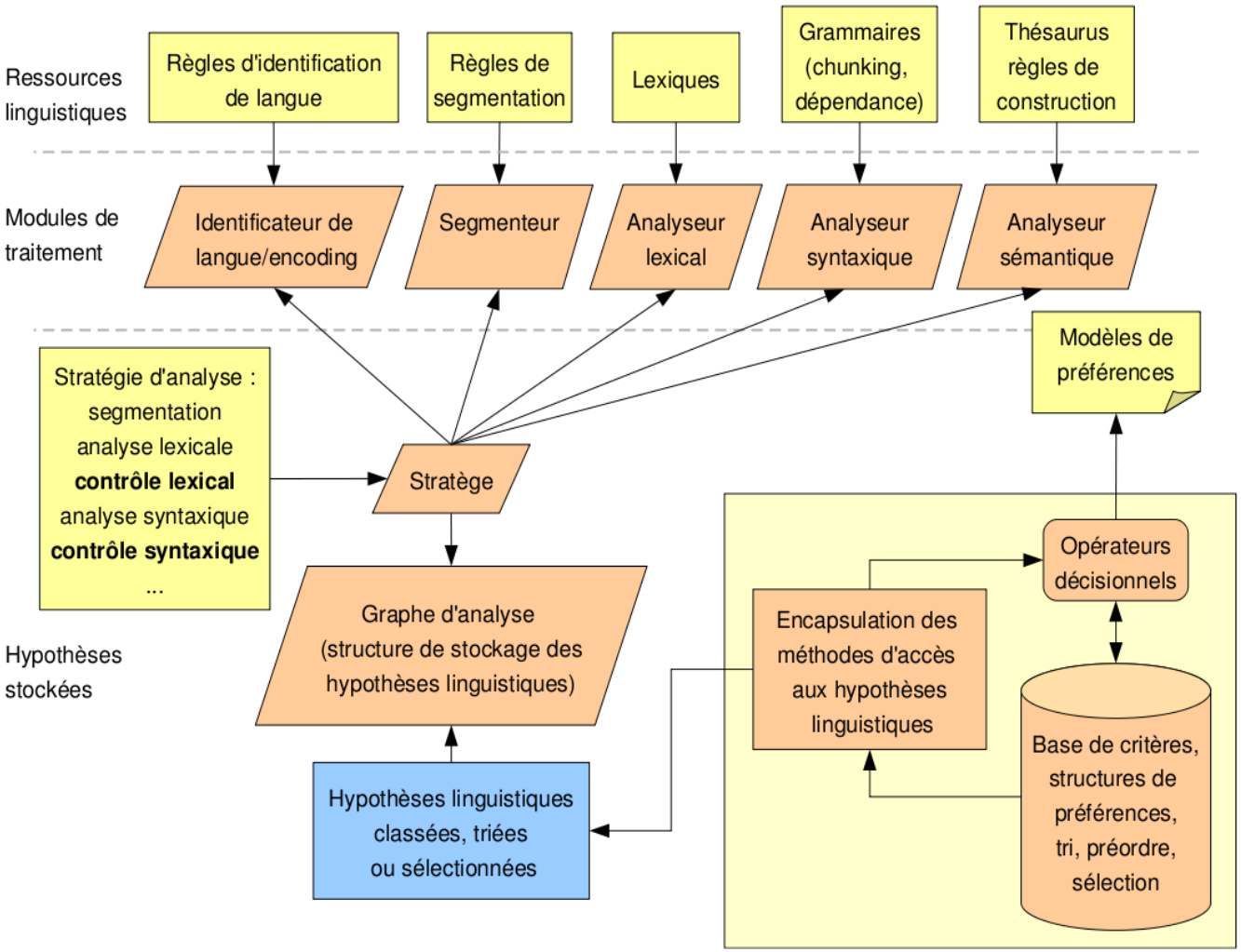


TiLT

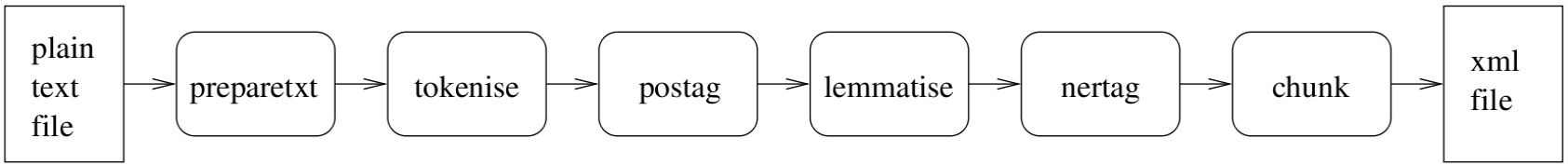
- Plateforme industrielle développée au laboratoire Orange Labs
- Intègre une importante panoplie de ressources et de modules d'analyse : correction lexicale, morphologie, analyse syntaxique (chunking et dépendances), analyse sémantique (réseaux sémantiques)
- Attention particulière sur la prise en compte du multilinguisme
- Utilisé dans des applications opérationnelles
- Fortes contraintes en termes de robustesse et de portabilité

Institution :	Orange Labs
Plateforme :	Linux et Windows XP
Langage :	C/C++
Référence(s) :	Guimier De Neef, Boualem, Chardenon, Filoche, et Vinesse (2002) ; Heinecke et al. (2008)

TiLT : schéma de l'architecture fonctionnelle



LT-TTT2



- LT-TTT2 n'est pas une plateforme à proprement parler
- LT-TTT2 est un pipeline d'outils développés au LTG (*Edinburgh Language Technology Group*)

Auteur(s) :	Claire Grover
Institution :	Edinburgh Language Technology Group (LTG)
Site web :	`http://www.ltg.ed.ac.uk/software/lt-ttt2`
Licence :	The University of Edinburgh Fee-Free Use (no modifications)
Plateforme :	Linux
Langage :	Anglais

Open NLP

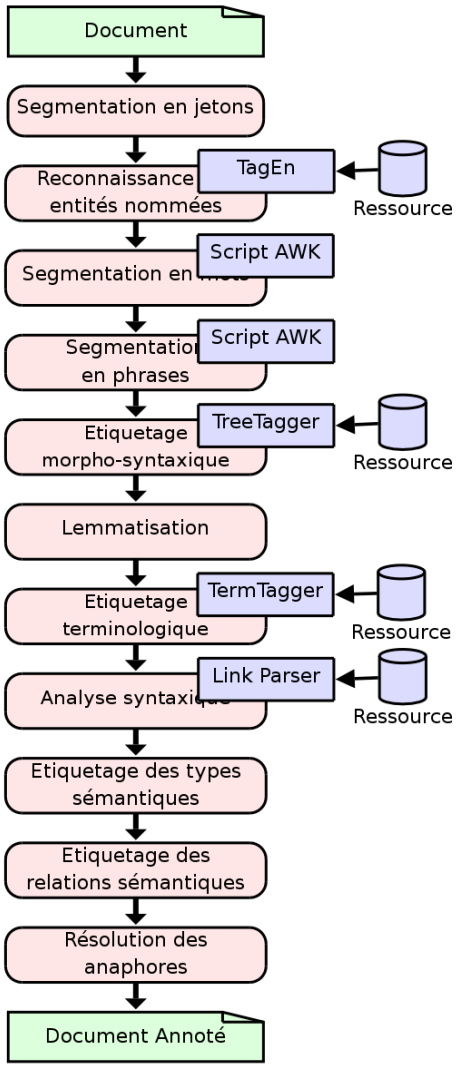
- OpenNLP n'est pas une plateforme à proprement parler
- OpenNLP est un regroupement de projets libres liés au TALN
- OpenNLP contient deux projets phares :

  Maxent : librairie Java d'apprentissage basé sur un modèle de maximisation d'entropie

  OpenNLP Tools : ensemble d'outils de TALN développés en Java et basés sur la librairie *Maxent* (segmentation, étiquetage morphosyntaxique, chunking, analyse syntaxique en constituants, détection d'entités nommées, extraction des coréférences)

Institution :	SourceForge.net
Site web :	`http://opennlp.sourceforge.net`
Licence :	open source projects
Plateforme :	Multiplateforme
Langage :	Java (essentiellement), C++, Python…

Ogmios (Alvis-NLPPlatform)



- Plateforme développée en 2006 au LIPN dans le cadre du projet Alvis
- Fonctionne comme pipeline figé et linéaire de modules autonomes
- Possibilité de redévelopper un module en Perl
- Annotations déportées dans un fichier xml respectant une DTD figée

Auteur(s) :	Thierry Hamon
Institution :	Laboratoire d'Informatique de Paris-Nord
Site web :	`http://search.cpan.org/~thhamon/Alvis-NLPPlatform-0.6`
Licence :	GPL
Plateforme :	Linux
Langage :	Perl
Référence(s) :	Hamon, Derivière, et Nazarenko (2007)

Unstructured Information Management Architecture (UIMA)

Cf. section *Apache UIMA*

Unstructured Information Management Architecture (UIMA)

- Pas une plateforme mais une librairie dédiée au développement de chaînes de traitement de l'information non structurée
- Implémentation initiée par IBM
- Devenu un projet *Open Source* en incubation à la fondation Apache en 2006
- Projet Apache de niveau supérieur (*Top level project*) depuis 2010

Institution :	Fondation Apache
Site web :	`http://uima.apache.org`
Licence :	open source
Plateforme :	Multiplateforme
Langage :	Java ou C++
Référence(s) :	Ferrucci et Lally (2004)

Apache UIMA

- Spécifications en cours de normalisation à l'OASIS [6] (*Organization for the Advancement of Structured Information Standards*)
- A pour ambition de s'imposer en tant que norme et standard industriels
- Documentation importante et existence de tutoriaux :
  - Apache UIMA Documentation (`http://uima.apache.org/documentation.html`)
  - UIMA-FR (`http://www.uima-fr.org`)

6. `http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=uima`

Apache UIMA

- Apache UIMA propose un framework contenant :
  - Ensemble de librairies permettant le développement de composants UIMA
  - Ensemble d'outils permettant le déploiement des composants
- Facilite l'intégration et le déploiement de composant d'annotation
- Les différents composants (*Analysis Engine*) s'échangent un ensemble d'annotations déportées (CAS) modélisé par un système de types (*Type Systeme*)

UIMA - *Analysis Engine* (AE)

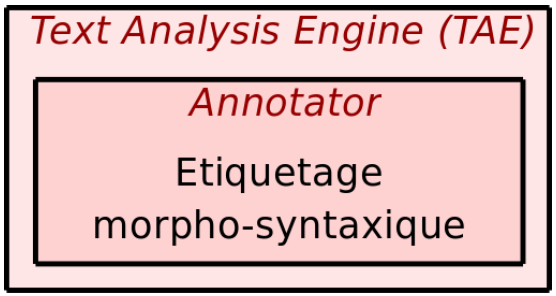
Analysis Engine (AE)

Composant fondamental de traitement

AE primitif :

- Partie déclarative (spécifications) en XML
- *Annotator* : implémentation en Java, en C++ ou sous forme de service Internet

Appelé TAE (*Text Analysis Engine*) quand il manipule des documents textuels



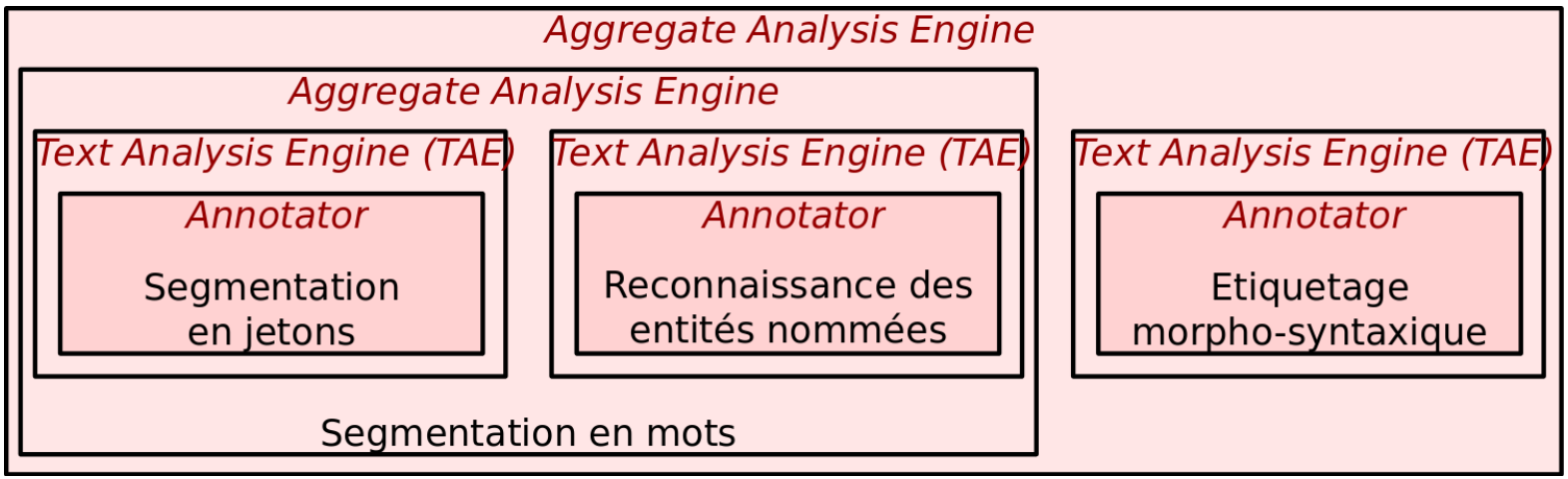
UIMA favorise la réutilisation et l'agrégation de composants en s'appuyant sur leur description XML

UIMA - *Analysis Engine* (AE)

Analysis Engine (AE)

Composant fondamental de traitement

AE complexe : composition ordonnée d'un ensemble d'AE complexes ou primitifs (*Aggregate Analysis Engine*)



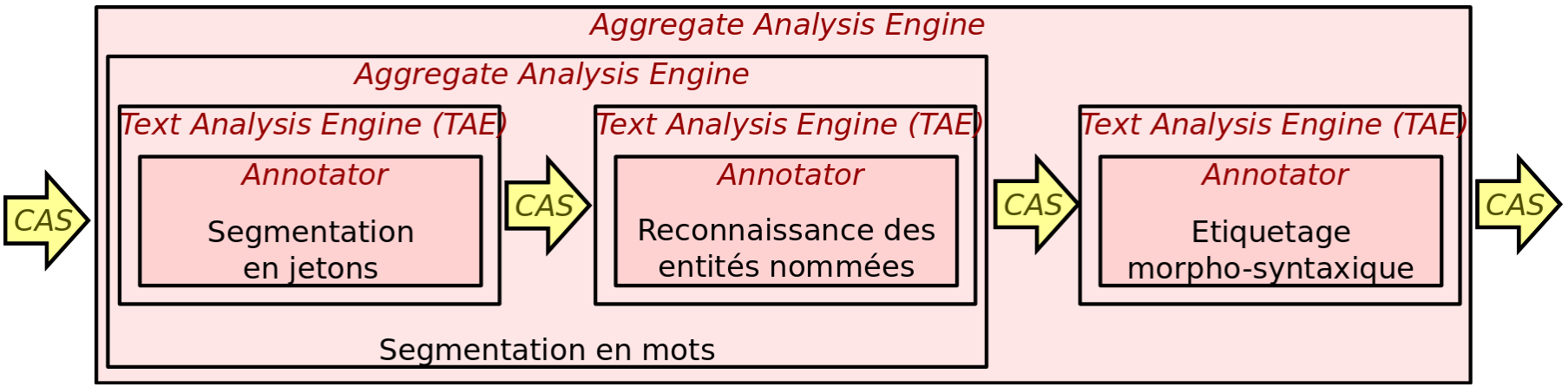
UIMA favorise la réutilisation et l'agrégation de composants en s'appuyant sur leur description XML

UIMA - *Common Analysis System* (CAS)

Common Analysis System (CAS)

Objet commun aux différents composants contenant le document original, ses méta-données (annotations) et une ou plusieurs interfaces pour accéder aux données

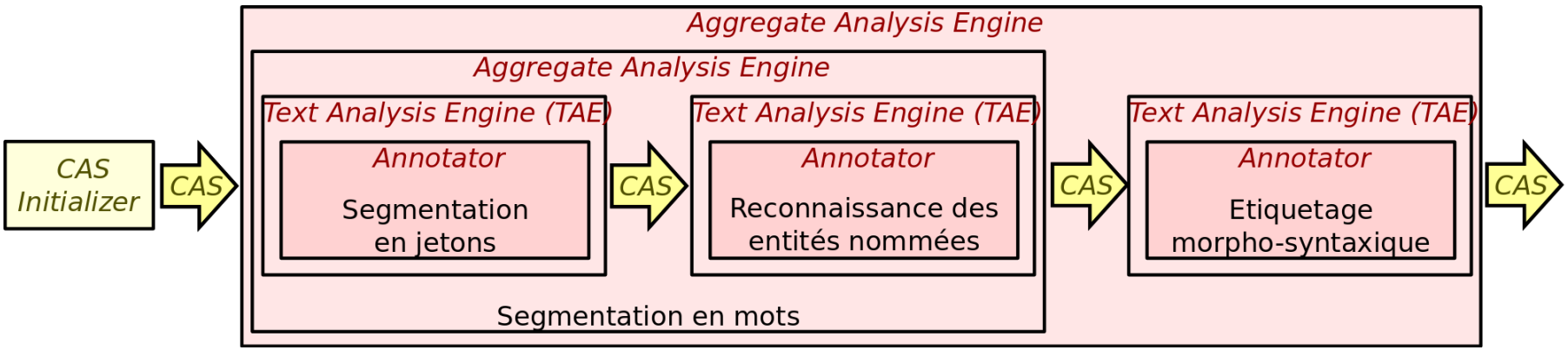
- Un TAE ne fait que compléter un CAS
- Pour plus de flexibilité, les annotations sont déportées



UIMA - *Common Analysis System Initialiser* (CAS)

Common Analysis System Initialiser (CAS)

Un *CAS Initialiser* est propre à un format de document source et a pour tâche de produire un objet CAS

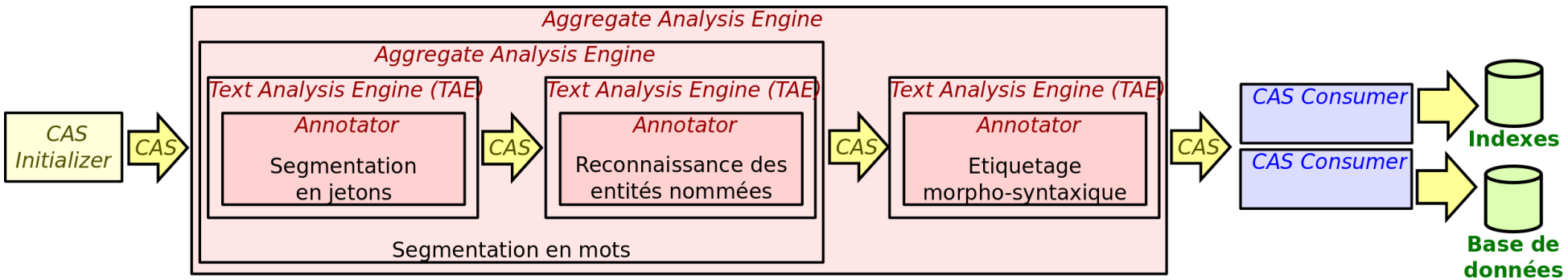


UIMA - *Common Analysis System Consumer* (CAS Consumer)

CAS Consumer

Intervient à la fin de la chaîne des différents AE pour produire, à partir des CAS, une ressource exploitable par une autre application (index, base de données…)

- Consomment des CAS mais n'en produisent pas
- Le rôle peut aller de la simple mémorisation des CAS à des inférences portant sur la totalité des CAS consommés

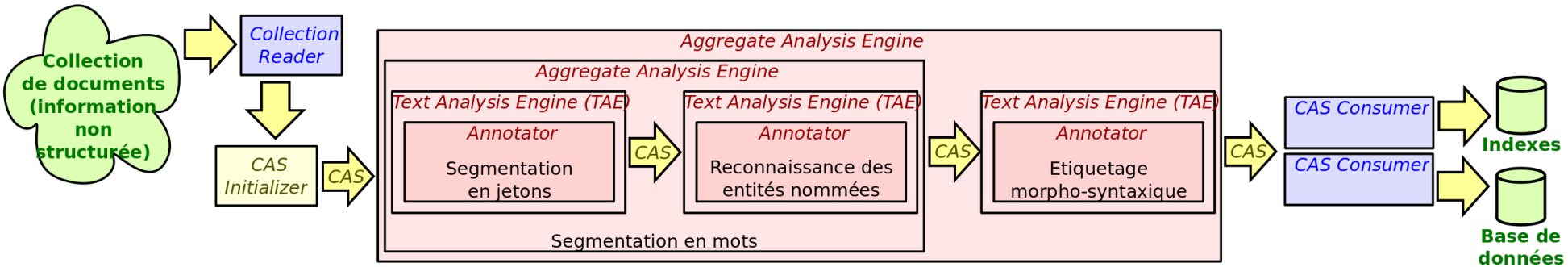


UIMA - *Collection Reader*

Collection Reader

Itère sur la collection des documents pour alimenter les *CAS Initialiser*

- La seule méthode d'un composant *Collection Reader* est « *passer au document suivant* »

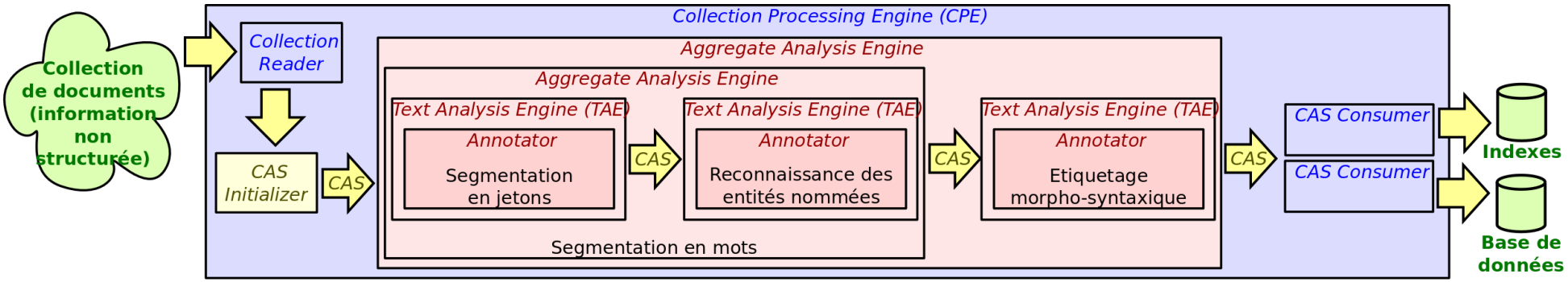


UIMA - *Collection Processing Engine* (CPE)

Collection Processing Engine (CPE)

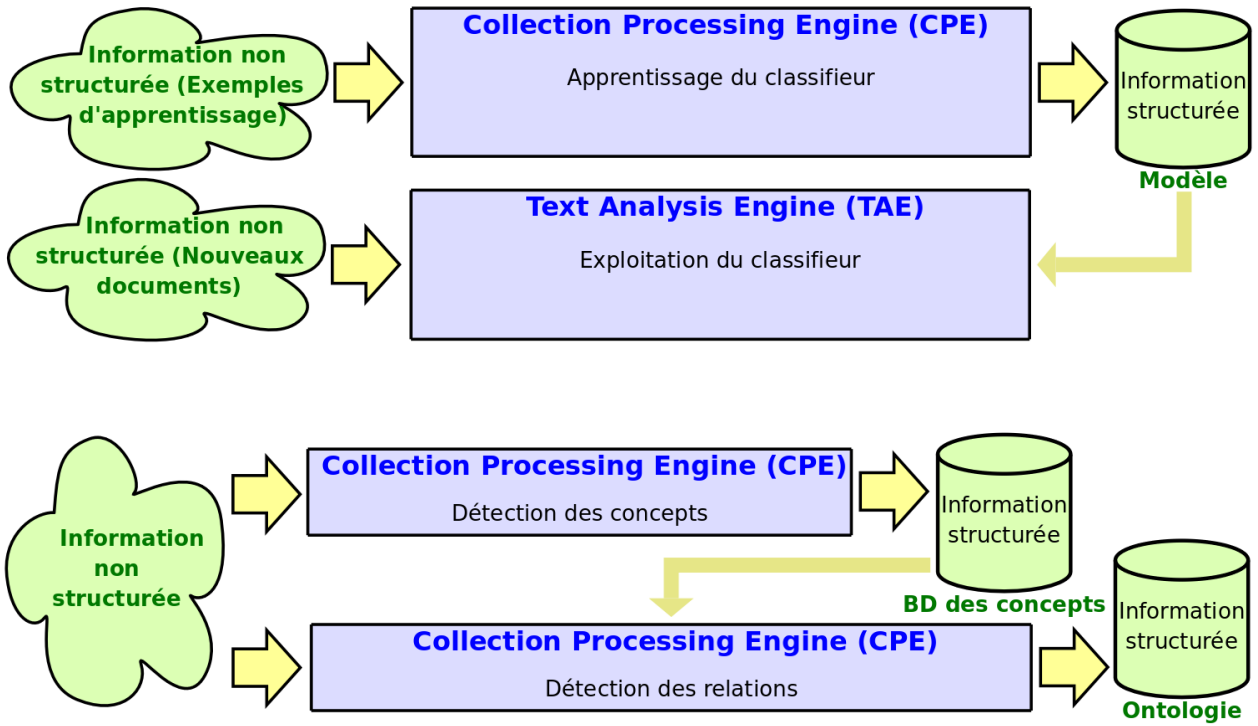
Composant complexe rassemblant tous les composants participant au traitement du *Collection Reader* jusqu'aux *CAS Consumer*

- Contrôle le flux entre ses différents composants



UIMA - Utilisation de ressources d'informations structurées

- UIMA permet à un AE d'accéder à une ressources d'informations structurées



UIMA - *Collection Processing Management* (CPM)

Collection Processing Management (CPM)

Composant permettant de déployer et d'exécuter un CPE dans un environnement UIMA

Le CPM permet :

- le démarrage, la pause et la reprise des traitements
- l'exécution d'un sous-ensemble des TAE en respectant les contraintes inhérentes aux méta-données du CAS
- la définition d'une stratégie concernant la gestion des documents provoquant des erreurs
- le monitoring des performances (temps, mémoire…)
- la parallélisation des traitements sur différents documents

Où trouver des outils de TALN ?

Sites rassemblant ou inventoriant de tels outils :

Natural Language Software Registry : `http://registry.dfki.de`

OpenNLP : `http://opennlp.sourceforge.net/projects.html`

Standford NLP : `http://nlp.stanford.edu/software`

CCG Software : `http://cogcomp.cs.illinois.edu/page/software`

CLARIN : `http://www.clarin.eu/view_tools`

ATALA : `http://www.atala.org/-Outils-pour-le-TAL-`

IMS : `http://www.ims.uni-stuttgart.de/projekte/gramotron/resources.html`

ISLanD : `https://www.greyc.fr/node/8?q=node/884`

Où trouver des composants UIMA ?

Sites rassemblant ou inventoriant des composants UIMA :

Apache UIMA : `http://uima.apache.org/annotators.html`

UIMA-FR : `http://uima-fr.org/download`

Références I

Apache UIMA. (2010a). *UIMA Overview & SDK Setup* .
(http://uima.apache.org/downloads/releaseDocs/2.3.0-incubating/docs/pdf/overview_and_setup.pdf).

Apache UIMA. (2010b). *UIMA References*.
(http://uima.apache.org/downloads/releaseDocs/2.3.0-incubating/docs/pdf/references.pdf).

Apache UIMA. (2010c). *UIMA Tools Guide and Reference*.
(http://uima.apache.org/downloads/releaseDocs/2.3.0-incubating/docs/pdf/tools.pdf).

Références II

Apache UIMA. (2010d). *UIMA Tutorial and Developers' Guides.* (`http://uima.apache.org/downloads/releaseDocs/2.3.0-incubating/docs/pdf/tutorials_and_users_guides.pdf`).

Aubin, S., & Hamon, T. (2006a). *Deliverable d6.3 - technical description of the term extractor - yatea* (Rapport technique). ALVIS (Superpeer semantic Search Engine).

Aubin, S., & Hamon, T. (2006b). Improving term extraction with terminological resources. *CoRR, abs/cs/0609019*.

Bilhaut, F. (2003). The linguastream platform. In *Proceedings of the 19th spanish society for natural language processing conference (sepln)* (p. p. 339-340). Alcalá de Henares, Spain.

Références III

Bourigault D., F. C. (2000). Approche linguistique pour l'analyse syntaxique de corpus. In *Cahiers de grammaire, 25, université toulouse le mirail* (p. pp.131-151).

Chaumartin, F.-R. (2008). Antelope : une plate-forme industrielle de traitement linguistique. *Traitement Automatique des Langues*, *49*(2).

Condé, C., & Viprey, J.-M. (2002). *A propos d'intex et d'unitex.* (`http://mshe.univ-fcomte.fr/intex/UX_COMM.htm`).

Références IV

Cunningham, H., Maynard, D., Bontcheva, K., & Tablan, V. (2002). GATE : A framework and graphical development environment for robust NLP tools and applications. In *Proceedings of the 40th annual meeting of the association for computational linguistics, philadelphia, pa, usa.*

Curran, J., Clark, S., & Bos, J. (2007, June). Linguistically motivated large-scale nlp with c&c and boxer. In *Proceedings of the 45th annual meeting of the association for computational linguistics companion volume proceedings of the demo and poster sessions* (pp. 33–36). Prague, Czech Republic : Association for Computational Linguistics.

Références V

Daille, B. (2003). Conceptual structuring through term variations. In D. M. F. Bond A. Korhonen & A. Villacicencio (Eds.), *Proceedings acl 2003 workshop on multiword expressions : Analysis, acquisition and treatment* (p. p. 9-16).

Encyclopédie Wikipédia. (2009). *Articles en ligne sur Wikipédia.* (`http://fr.wikipedia.org`).

Enjalbert, P. (2008). Plate-formes pour le traitement automatique des langues - Préface. *Traitement Automatique des Langues*, *49*(2).

Ferrucci, D., & Lally, A. (2004). Uima : an architectural approach to unstructured information processing in the corporate research environment. In *Natural language engineering* (Vol. 10, p. p. 327-348).

Références VI

Finkel, J. R., Grenager, T., & Manning, C. (2005). Incorporating non-local information into information extraction systems by gibbs sampling. In *Acl '05 : Proceedings of the 43rd annual meeting on association for computational linguistics* (pp. 363–370). Morristown, NJ, USA : Association for Computational Linguistics.

Grinberg, D., Lafferty, J., & Sleator, D. (1995, September). A robust parsing algorithm for link grammars. In *Proceedings of the fourth international workshop on parsing technologies*. Prague. (This paper describes the modifications of the parsing algorithm used to allow null links)

Références VII

Guimier De Neef, E., Boualem, M., Chardenon, C., Filoche, P., & Vinesse, J. (2002). Natural language processing software tools and linguistic data developed by france télécom r&d. In *European conference on multilingual technologies*. Pune, India.

Hamon, T., Derivière, J., & Nazarenko, A. (2007, 05 06). Ogmios : une plate-forme d'annotation linguistique. In *Actes de TALN 2007 Traitement Automatique des Langues Naturelles* (p. 103-112). Toulouse France : IRIT Press. (ALVIS STREP Project)

Heinecke, J., Smits, G., Chardenon, C., Guimier De Neef, E., Maillebuau, E., & Boualem, M. (2008). Tilt : plate-forme pour le traitement automatique des langues naturelles. *Traitement Automatique des Langues*, *49*(2).

Références VIII

Jacquemin, C. (1997). *Variation terminologique : reconnaissance et acquisition automatiques de termes et de leurs variantes en corpus*. Memoire d'habilitation a diriger des recherches en informatique fondamentale, Universite de Nantes.

Jacquemin, C., Klavans, J. L., & Tzoukermann, E. (1997). Expansion of multi-word terms for indexing and retrieval using morphology and syntax. In *In proceedings of the 35th annual meeting of the acl* (pp. 24–31). ACL.

Références IX

Lafferty, J., Sleator, D., & Temperley, D. (1992, October). Grammatical trigrams : A probabilistic model of link grammar. In *Proceedings of the aaai conference on probabilistic approaches to natural language*. (This paper introduces a statistical language model based on link grammars)

MSHE. (2002). *Commentaires sur la documentation unitex.* (`http://mshe.univ-fcomte.fr/intex/Unitex.htm`). (Maison des Sciences de l'Homme et de l'Environnement (MSHE))

Mustapha Es-salihe, & Stéphane Bond. (2006). *Étude des frameworks UIMA, Gate et OpenNLP.* (`http://www.crim.ca/fr/R-D/Technologies_Internet/documents/Etude-UIMA-GATE-OpenNLP.pdf`).

Références X

Paumier, S. (2010). *Unitex 2.1 - user manual.* (`http://www-igm.univ-mlv.fr/~unitex`). (Institut Gaspard-Monge (IGM))

Ratinov, L., & Roth, D. (2009). Design challenges and misconceptions in named entity recognition. In *Conll*.

Roussanaly, A. (2003). *FS : API Feature Structures.* (`http://webloria.loria.fr/~azim/FS`).

Schmid, H. (1994). Probabilistic part-of-speech tagging using decision trees. In *Proceedings of the international conference on new methods in language processing*. Manchester, UK.

Références XI

Schmid, H. (1995). Improvements in part-of-speech tagging with an application to german. In *In proceedings of the acl sigdat-workshop* (pp. 47–50).

Sclano, F., & Velardi, P. (2007, Octobre). Termextractor : a web application to learn the common terminology of interest groups and research communities. In *9th conf. on terminology and artificial intelligence tia 2007*. Sophia Antinopolis.

Silberztein, M. (1993). *Dictionnaires électroniques et analyse automatique de textes : le système intex*. Paris : Masson.

Silberztein, M. (2003). *Nooj manual*. (http ://www.nooj4nlp.net).

Références XII

Sleator, D., & Temperley, D. (1993). Parsing english with a link grammar. In *Third international workshop on parsing technologies*. (This is a shorter but more up-to-date version of the technical report above. It contains a better introduction to link grammars, and gives a more detailed description of the relationship between link grammar and other formalisms.)

sourceforge. (2010). *opennlp.maxent*. (`http://maxent.sourceforge.net/`).

Tellier, I. (2010). *Introduction au taln et à l'ingénierie linguistique*. (`http://www.univ-orleans.fr/lifo/Members/Isabelle.Tellier/`).

Références XIII

Toutanova, K., Klein, D., Manning, C. D., & Singer, Y. (2003). Feature-rich part-of-speech tagging with a cyclic dependency network. In *Naacl '03 : Proceedings of the 2003 conference of the north american chapter of the association for computational linguistics on human language technology* (pp. 173–180). Morristown, NJ, USA : Association for Computational Linguistics.

Toutanova, K., & Manning, C. D. (2000). Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In *Emnlp '00 : Proceedings of the 2000 joint sigdat conference on empirical methods in natural language processing and very large corpora* (pp. 63–70). Morristown, NJ, USA : Association for Computational Linguistics.

Références XIV

Véronis, J. (2001). *INF Z18 - informatique et linguistique I.* (`http://sites.univ-provence.fr/veronis`).

Widlöcher, A., & Bilhaut, F. (2008). Articulation des traitements en tal. *Traitement Automatique des Langues*, *49*(2).

Yvon, F. (2010). *Une petite introduction au traitement automatique des langues naturelles.* (`http://www.univ-orleans.fr/lifo/Members/Isabelle.Tellier/`).