# Génération de phrases multilingues par apprentissage automatique de modèles de phrases.

*Soutenance de thèse, Eric Charton*
*Sous la direction de Mr Juan-Manuel Torres Moreno*

ÉCOLE POLYTECHNIQUE MONTRÉAL

LABORATOIRE INFORMATIQUE D'AVIGNON
Université d'Avignon et des Pays de Vaucluse

UNIVERSITÉ D'AVIGNON

# Définition

La Génération Automatique de Texte (GAT) est le champ de recherche de la linguistique informatique qui étudie la possibilité d'attribuer à une machine la faculté de produire du texte intelligible.

# Plan

# Préambule

# Les précurseurs

## Une réflexion ancienne

- Claude Shanon *Génération de phrases par modèle de langage* ( [Théorie Mathématique de la Communication, 1948] )

# Les précurseurs

## Une réflexion ancienne

- Claude Shanon *Génération de phrases par modèle de langage* ( [Théorie Mathématique de la Communication, 1948] )
- Alan Turing *Le jeu de l'imitation* ( [Computing machinery and intelligence,1950] )

# Les précurseurs

## Une réflexion ancienne

- Claude Shanon *Génération de phrases par modèle de langage* ( [Théorie Mathématique de la Communication, 1948] )
- Alan Turing *Le jeu de l'imitation* ( [Computing machinery and intelligence,1950] )

## A partir des années 50, insérer la génération de phrases dans un processus de traduction

# Les précurseurs

## Une réflexion ancienne

- Claude Shanon *Génération de phrases par modèle de langage* ( [Théorie Mathématique de la Communication, 1948] )
- Alan Turing *Le jeu de l'imitation* ( [Computing machinery and intelligence,1950] )

## A partir des années 50, insérer la génération de phrases dans un processus de traduction

- Victor Yngve applique les théories des *Structures Syntaxiques* de Chomsky ( [Generation of English Sentences,1961] )

# Les précurseurs

## Une réflexion ancienne

- Claude Shanon *Génération de phrases par modèle de langage* ( [Théorie Mathématique de la Communication, 1948] )
- Alan Turing *Le jeu de l'imitation* ( [Computing machinery and intelligence,1950] )

## A partir des années 50, insérer la génération de phrases dans un processus de traduction

- Victor Yngve applique les théories des *Structures Syntaxiques* de Chomsky ( [Generation of English Sentences,1961] )
- Tentative d'implémenter les grammaires dans des systèmes de génération combinatoires [Mathews,1962]

# Que dire et comment le dire: la gestion de l'intention à communiquer

La GAT inclut à deux extrémités d'une chaîne de traitement:

## Que dire:

D'un côté une intention de communication qui peut prendre la forme d'une information structurée ou d'une donnée

La GAT inclut à deux extrémités d'une chaîne de traitement:

## Que dire:

D'un côté une intention de communication qui peut prendre la forme d'une information structurée ou d'une donnée

Exemple d'*Intention de Communication*

| New Maison |
| --- |
| Maison.Propriétaire=John;<br>Maison.Attributs=Blanche;Grande;<br>Maison.Situation=Rue de Paris;proche de l'école primaire; |

La GAT inclut à deux extrémités d'une chaîne de traitement:

## Que dire:

D'un côté une intention de communication qui peut prendre la forme d'une information structurée ou d'une donnée

Exemple d'*Intention de Communication*

| New Maison |
| --- |
| Maison.Propriétaire=John;<br>Maison.Attributs=Blanche;Grande;<br>Maison.Situation=Rue de Paris;proche de l'école primaire; |

## Comment le dire:

À l'autre extrémité, une phrase ou un groupe de phrases générées

Plusieurs étapes pour transformer cette représentation en phrases.

## *Planification*

- John possède une maison. La maison de John est blanche. La maison de John est grande. La maison de John est située rue de Paris. La maison de John est proche de l'école primaire.

Plusieurs étapes pour transformer cette représentation en phrases.

## *Planification*

- John possède une maison. La maison de John est blanche. La maison de John est grande. La maison de John est située rue de Paris. La maison de John est proche de l'école primaire.

## *Lexicalisation*:

- John [possède;détient] une [maison;demeure]. La [maison;demeure] de John est blanche. La [maison;demeure] de John est [grande;spacieuse].

Plusieurs étapes pour transformer cette représentation en phrases.

## *Planification*

- John possède une maison. La maison de John est blanche. La maison de John est grande. La maison de John est située rue de Paris. La maison de John est proche de l'école primaire.

## *Lexicalisation*:

- John [possède;détient] une [maison;demeure]. La [maison;demeure] de John est blanche. La [maison;demeure] de John est [grande;spacieuse].

## *Production d'expressions co-référentes*:

- John [possède;détient] une [maison;demeure]. [Elle—La{maison;demeure}] est blanche. [[Sa][maison;demeure]——[elle]] est [grande;spacieuse].

Plusieurs étapes pour transformer cette représentation en phrases.

## *Planification*

- John possède une maison. La maison de John est blanche. La maison de John est grande. La maison de John est située rue de Paris. La maison de John est proche de l'école primaire.

## *Lexicalisation*:

- John [possède;détient] une [maison;demeure]. La [maison;demeure] de John est blanche. La [maison;demeure] de John est [grande;spacieuse].

## *Production d'expressions co-référentes*:

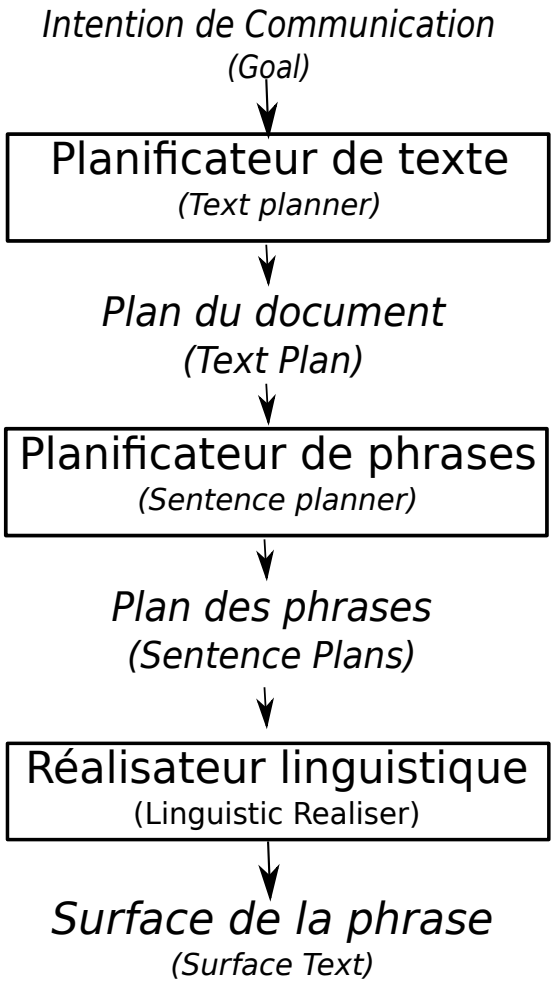
- John [possède;détient] une [maison;demeure]. [Elle—La{maison;demeure}] est blanche. [[Sa][maison;demeure]——[elle]] est [grande;spacieuse].

## *Agrégation*

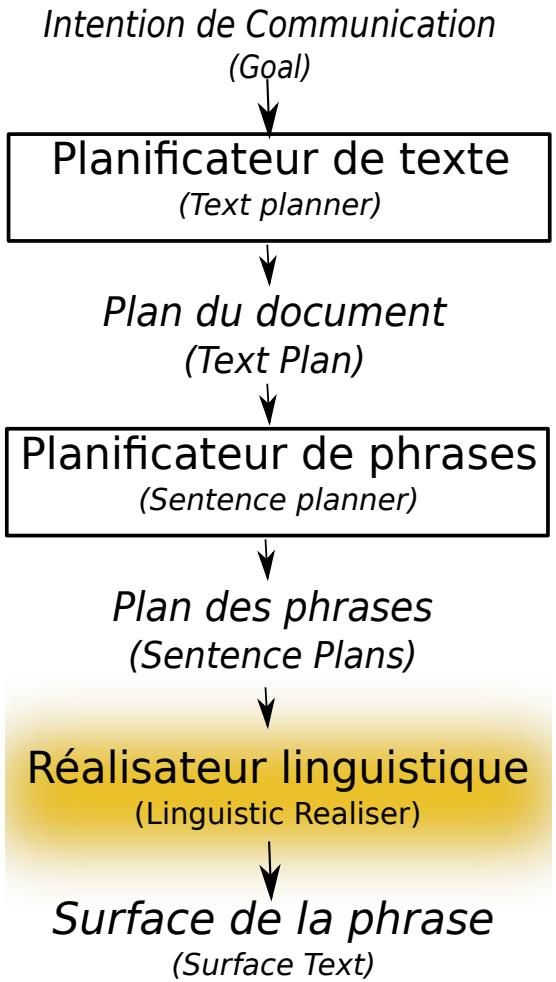
- John possède une grande maison blanche située rue de Paris, à proximité de l'école primaire.

# Architecture et fonctionnement d'un système de GAT

# L'architecture consensuelle en *pipeline*[Reiter2000]



# L'architecture consensuelle en *pipeline*[Reiter2000]



# Le composant de réalisation linguistique

## Approches par patrons

- Des modèles de phrases prédéfinis contenant des éléments variables
- Le modèle le plus fréquent [Deemter2005]

# Le composant de réalisation linguistique

## Approches par patrons

- Des modèles de phrases prédéfinis contenant des éléments variables
- Le modèle le plus fréquent [Deemter2005]

## Approches à base de règles et de grammaires

- Composants de réalisation inspirés d'une théorie linguistique dans une architecture modulaire en *pipeline* [Reiter1994]

# Le composant de réalisation linguistique

## Approches par patrons

- Des modèles de phrases prédéfinis contenant des éléments variables
- Le modèle le plus fréquent [Deemter2005]

## Approches à base de règles et de grammaires

- Composants de réalisation inspirés d'une théorie linguistique dans une architecture modulaire en *pipeline* [Reiter1994]

## Approches statistiques et n-grammes

- Systèmes probabilistes guidés reposant sur des assemblages de n-grammes [Langdike1998]

# Exemple de système à base de patrons

$$f(n, d, h) = \text{Le train [n] à destination de [d] partira à [h]} \quad (1)$$

$$f(n = 755, d = Paris, h = 12h) \quad (2)$$

*Le train 755 à destination de Paris partira à 12h.*

## Inconvénients des générateurs à base de patrons

- Tous les cas ne sont pas traités (ex *destination de Avignon*)
- Modèles contraignants
- Impossible de faire varier le temps

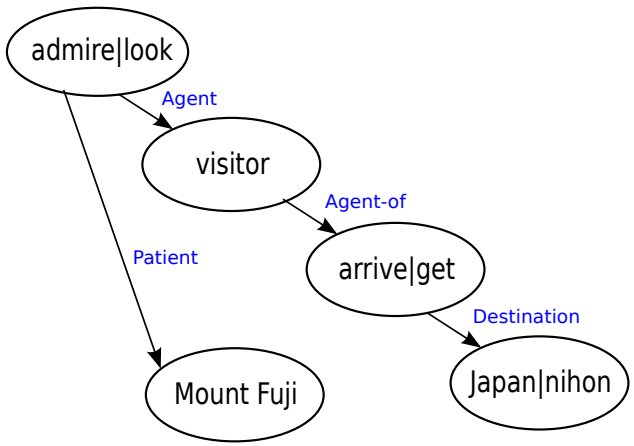
# Système à base grammaires

Une théorie linguistique est implémentée pour produire la structure syntaxique des phrases générées

- **SFG** (*Systemic functional grammar*) : axé sur la syntaxe et les classes de mots. Système PENMAN [Matthiessen1991]
- **TAG** (*Tree-Adjoining grammar*): grammaires hors contexte avec ré-écriture par un arbre. Nombreux systèmes [Stone1997,Meunier1998,Danlos2000]
- Théories rarement implémentées **LFG**, **GBG** ...
- **Règles de production dérivées des systèmes à base d'arbres**: SimpleNLG [Reiter, Gatt, 2009]

# Système statistiques

Un réalisateur linguistique rudimentaire est complété par une correction d'après un modèle de langage




Exemple de représentations symboliques fournies par NITROGEN.

| Rang | Phrase |
|---|---|
| 1 | Visitors who came in Japan admire Mount Fuji . |
| 2 | Visitors who came in Japan admires Mount Fuji . |
| 3 | Visitors who arrived in Japan admire Mount Fuji . |
| 7 | The visitor who came in Japan admire Mount Fuji . |
| 8 | Visitors who came to Japan admires Mount Fuji . |

# Représentativité des approches

Sur la liste de Bateman et Zock (www.nlg-wiki.org)

- **Systèmes à base de patrons** : une dizaine de systèmes, en réalité majoritairement appliqués
- **Systèmes à base de grammaires et de règles** : majoritaires avec des grammaires très variées (génératives, à base d'arbres), 70 systèmes
- **Systèmes à composants statistiques** : une dizaine, très différents

# Architecture de génération à base de corpus de modèles de phrases

# Proposition

Insérer dans une architecture *pipeline* un composant de génération de surface utilisant des modèles de phrases appris sur un corpus:

- Modélisation: Transformer, par une approche pragmatique, ces phrases en patrons décrits par une formule logique
- Génération: Identifier les phrases un formalisme de prédicats logiques du 1er ordre inspiré de la Discourse Representation Theory (DRT) [Kamp, 1981]

# Hypothèse

Dans le cadre de la logique des prédicats un langage $L$.
Il prévoit deux variables:

- $u$ qui est une description sémantique exprimée en *logique de prédicats*
- $p$ qui est une phrases syntaxiquement correcte

et un prédicat:

- $S$ qui définit la relation sémantique entre $u$ et $p$

$$\forall p \exists u \{S(p, u)\} \tag{3}$$

## Appliquée à la GAT

- Pour toute phrase $p$ contenue dans $L$, il existe une représentation sémantique formelle $u$.
- Pour une Intention de Communication représentée par $u$ il existe une probabilité variable qu'une phrase $p$ pré-existante puisse représenter $u$.

# Exemple d'entrée utilisée pour générer un texte

| Pers=Romain Gary | |
|---|---|
| Pers.pseudonyme | Émile Ajar |
| Pers.fonction | Écrivain |
| Pers.bornplace | Vilnius |
| Pers.deathplace | Paris |
| Pers.borndate | 8 mai 1914 |
| Pers.deathdate | 2 décembre 1980 |
| Pers.nationalité | Français |

$$DRT = \left( \begin{array}{l} (a, b, c, d, e, f, g) \\ (\text{Romain Gary(a)}, \text{Émile Ajar}(b), \text{écrivain}(c), \text{Vilnius}(d), 8 \text{ mai } 1914(d), \\ \text{français}(e), \text{Paris}(f), 2 \text{ décembre } 2980(g)) \\ (\text{Pseudonyme}(a, b), \text{Fonction}(a, c), \text{Bornplace}(a, d), \text{Borndate}(a, d), \\ \text{Bornplace}(a, d), \text{Deathdate}(a, g)\text{Deathplace}(a, f)) \end{array} \right) \quad (4)$$

# Collecte de phrases éventuellement compatibles avec la DRT

Collecte manuelle réalisée dans Wikipédia:

1. Honoré de Balzac, né Honoré Balzac, à Tours le 20 mai 1799 (1er prairial an VII) et mort à Paris le 18 août 1850
2. Alphonse de Lamartine, de son nom complet Alphonse Marie Louis de Prat de Lamartine, né à Mâcon le 21 octobre 1790 et mort à Paris le 28 février 1869.
3. Gérard de Nerval, pseudonyme de Gérard Labrunie, né à Paris le 22 mai 1808 et mort à Paris le 26 janvier 1855, était un poète français.
4. Jean-Paul Sartre (Jean-Paul Charles Aymard Sartre), né le 21 juin 1905 à Paris et mort le 15 avril 1980 à Paris, est un philosophe et écrivain français.

- **La phrase 3 est compatible avec IC: elle devient une Phrase Support**

# Remplacement des contenus d'une phrase abstraite

- Gérard de Nerval, pseudonyme de Gérard Labrunie, né à Paris le 22 mai 1808 et mort à Paris le 26 janvier 1855, était un poète français.

Abstraction d'une phrase support

- [Pers.pseudonyme], pseudonyme de [Pers], né à [Pers.bornplace] le [Pers.borndate] et mort à [Pers.deathplace] le [Pers.deathdate], était un [Pers.fonc] [Pers.nationalité].

Dans notre proposition de système, il serait finalement possible de transformer le modèle de phrase ci-dessus pour qu'il reçoive le contenu de IC:

- Émile Ajar, pseudonyme de Romain Gary, né à Vilnius le 8 mai 1914 et mort à Paris le 2 décembre 1980, était un écrivain français.

# Architecture proposée

Principe : le composant de réalisation est un algorithme de RI appliqué sur un Corpus de Phrases Modèles.

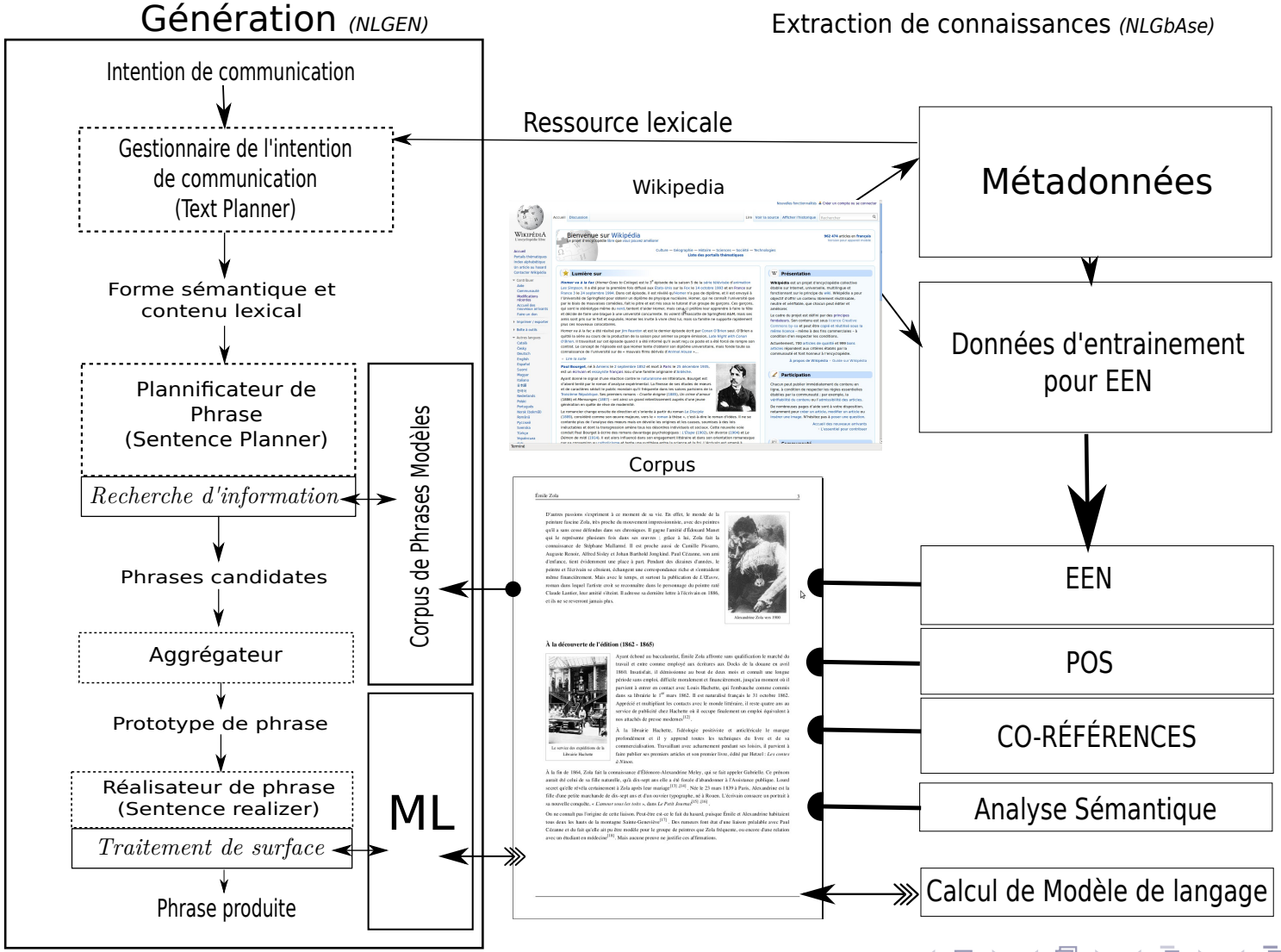
## Apprentissage de modèles de phrases

- Phrases apprises sur un corpus de grande taille
- Des méthodes d'étiquetage et d'analyse rendent les phrases abstraites

## Génération de phrases

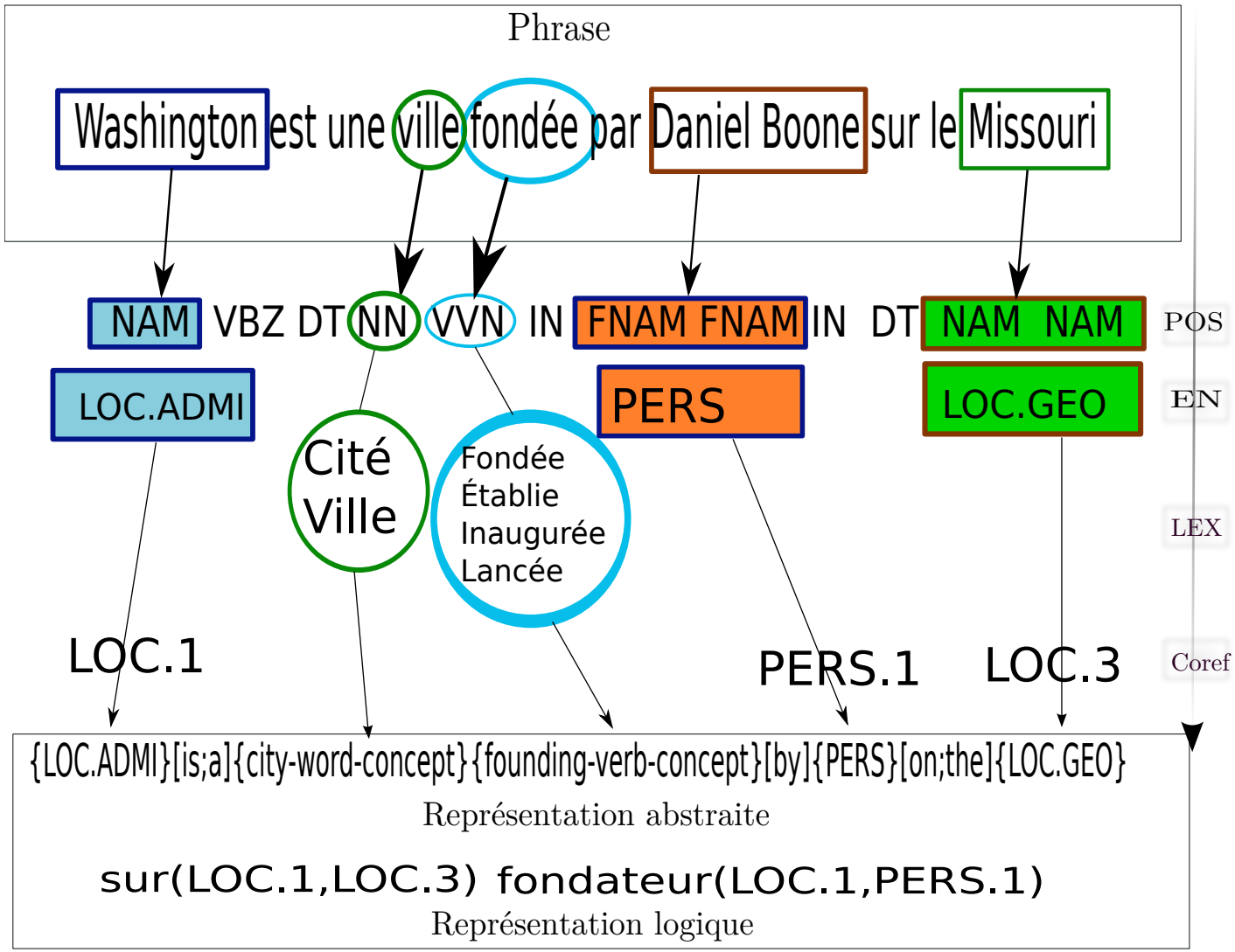
- Identifier la phrase modèle la plus appropriée pour exprimer une Intention de Communication
- Modifier le contenu informatif de la phrase en conservant sa syntaxe

# Architecture proposée

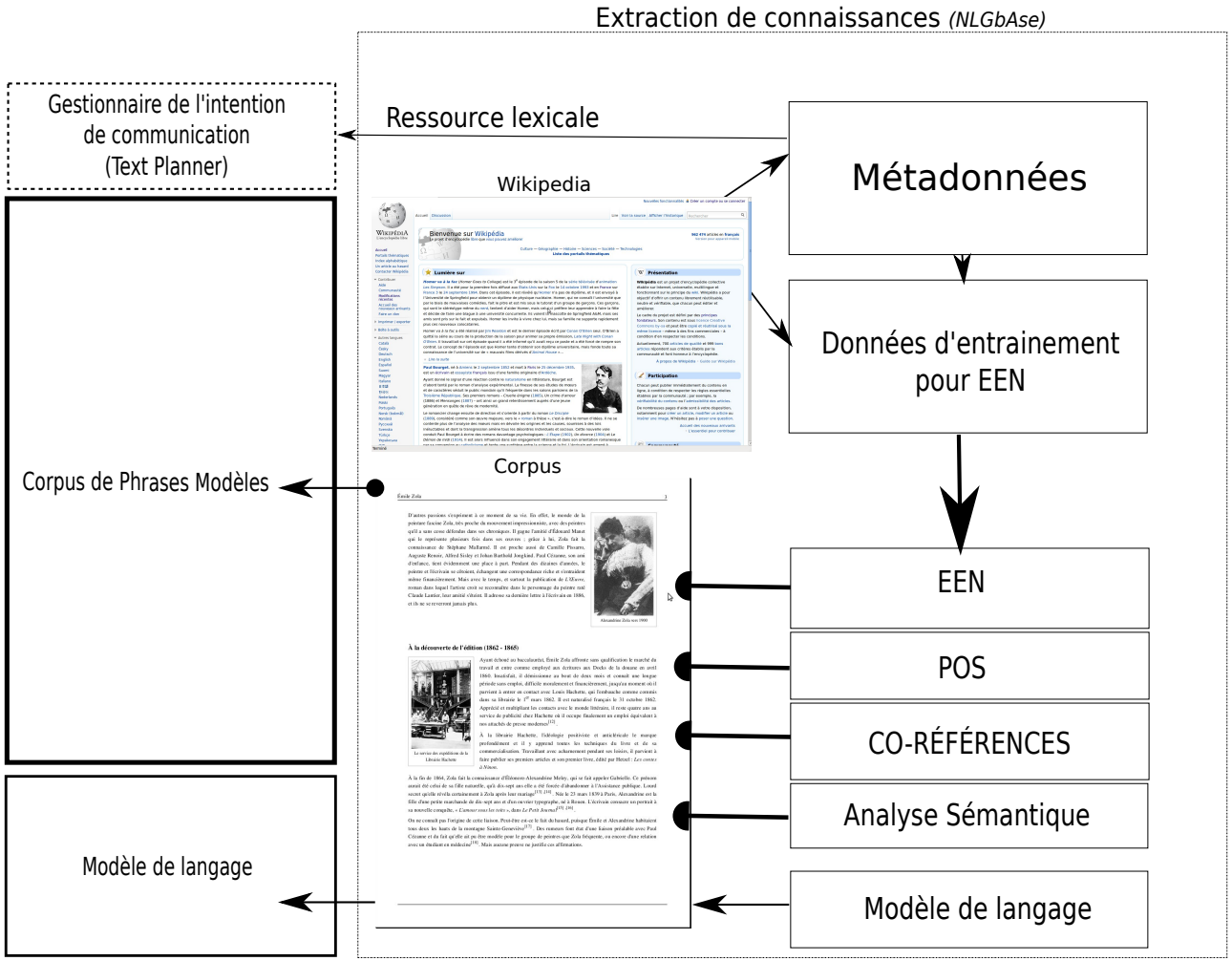


# Extraction de connaissances

# Les différents niveaux d'étiquetage possibles



# Architecture d'extraction de connaissances



# Ressources et outils d'étiquetage retenus

- Ressource lexicale : métadonnées et Wordnet
- Étiqueteur morpho-syntaxique: TreeTagger
- Étiqueteur d'entités nommées: méthode CRF
- Étiqueteur de co-références: méthode hybride
- Analyseur sémantique d'après des arbres de dépendances

# Corpus de ressources

Wikipédia: le seul corpus multilingue, de grande taille, et déjà structuré

## En tant que ressource d'apprentissage

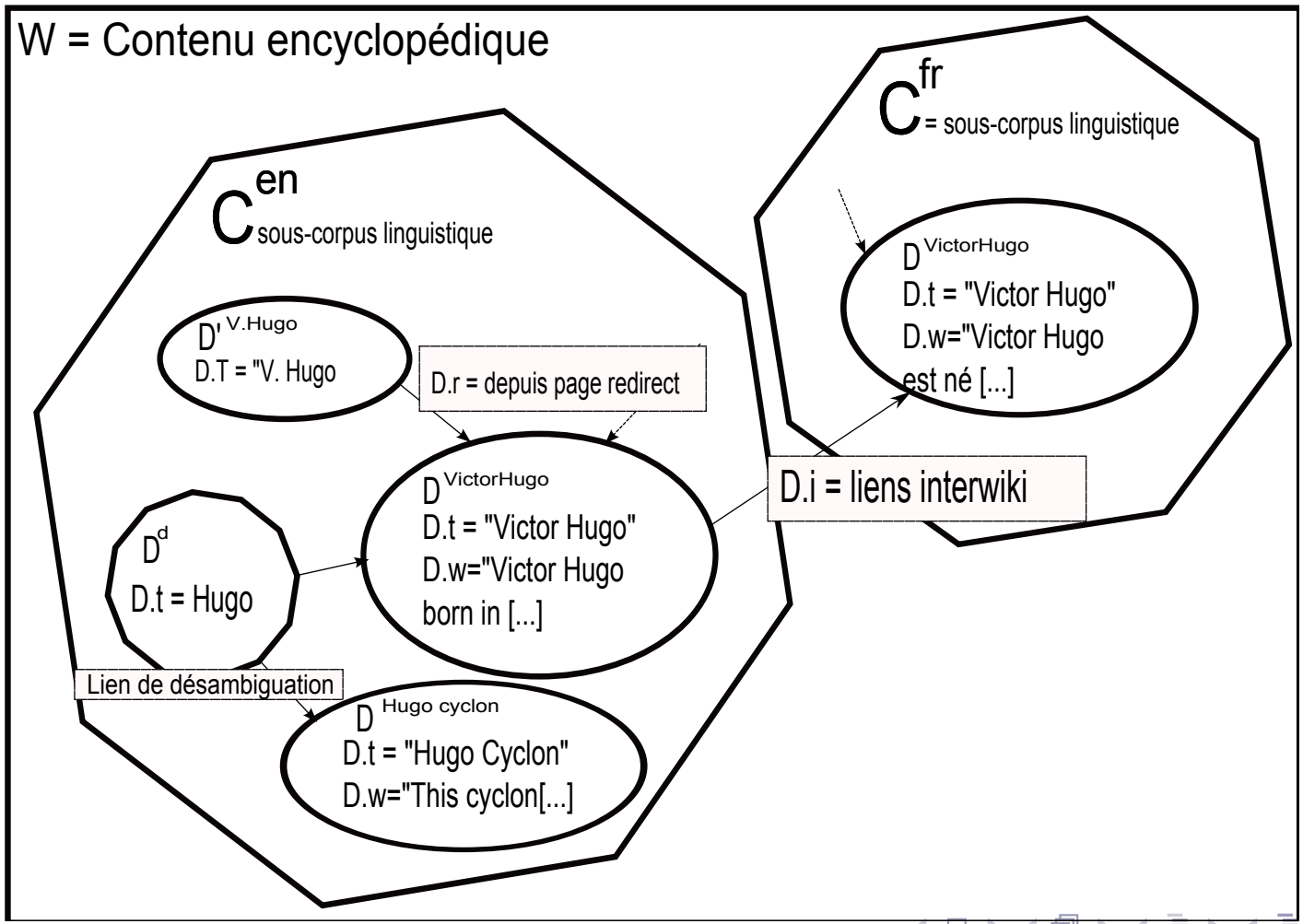
- Concepts aisés à transformer en ressource lexicale
- Vaste quantité de phrases pré-étiquetées par des liens internes
- Ressources de base pour la modélisation de phrases

## Pour les expériences de génération

- 10 millions de phrases en Français, 90 millions en Anglais, 3 millions en Espagnol
- Un vaste échantillonage de phrases disponible pour exprimer des intentions diversifiées

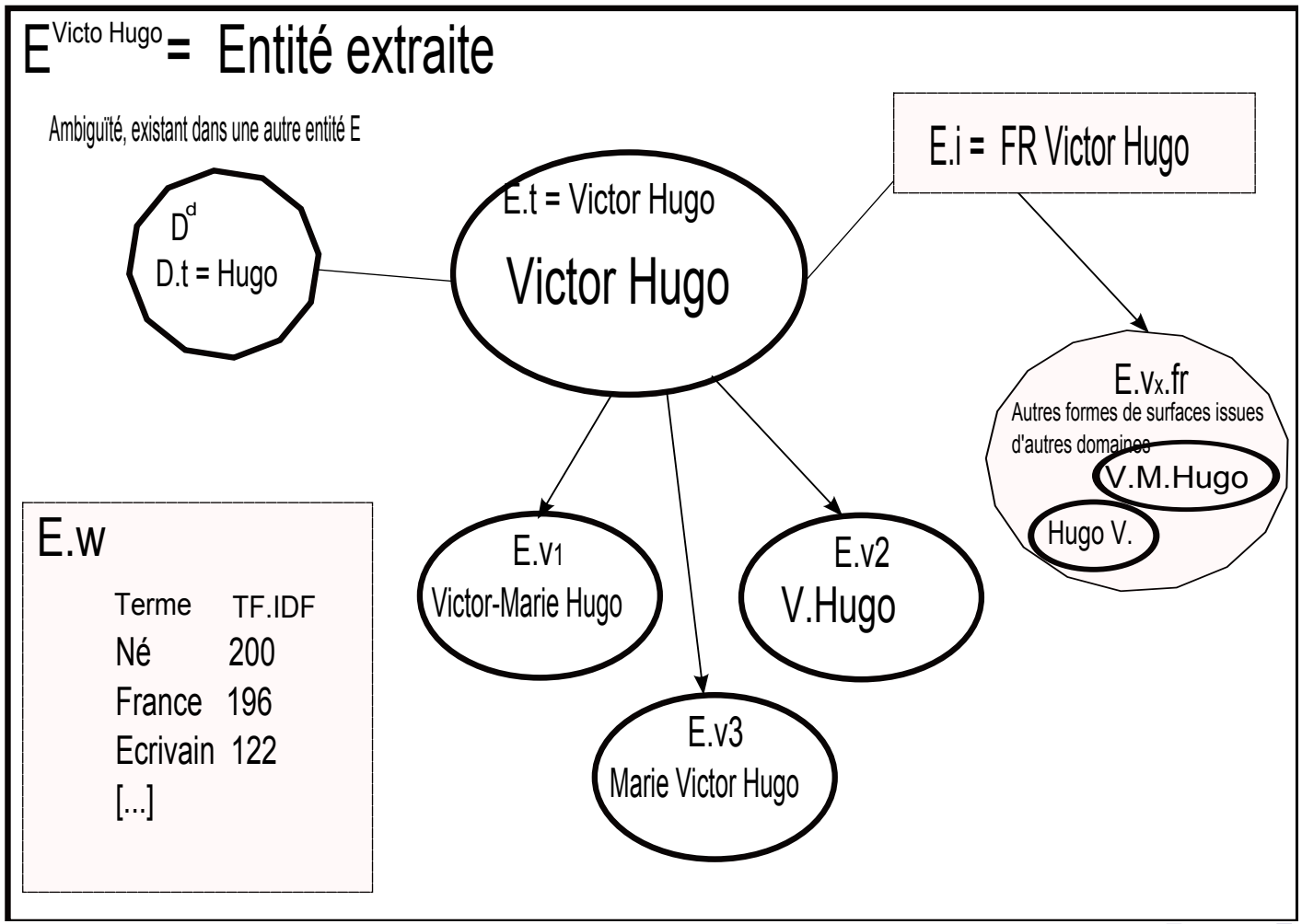
# Élaboration de métadonnées

Structure des données dans Wikipédia.



# Élaboration de métadonnées

Structure d'une *metadonnée*.



# Étiquetage d'entités nommées

Métadonnées: produisent automatiquement des corpus d'entrainement de CRF

| Format Wikipédia | $C_{d1}$ | $C_{d2}$ (texte) | $C_{d2}$ (POS) | $C_{d3}$ (EN) |
|---|---|---|---|---|
| The | The | The | DT | UNK |
| national | national | national | JJ | UNK |
| team | team | team | NN | UNK |
| of | of | of | IN | UNK |
| [[Republic of Kenya— | $< entbegin >$ | | *début lien loc* | |
| Kenya | Kenya | Kenya | NAM | LOC.ADMI |
| ]] | $< tag = loc.admi >$ | | *fin lien loc* | |
| is | is | is | VBZ | UNK |
| controlled | controlled | controlled | VVN | UNK |
| by | by | by | IN | UNK |
| the | the | the | DT | UNK |
| [[ | $< entbegin >$ | | *début lien org* | - |
| Kenya | Kenya | Kenya | NAM | ORG.DIV |
| Football | Football | Football | NAM | ORG.DIV |
| Federation | Federation | Federation | NAM | ORG.DIV |
| ]] | $< tag = org.div >$ | | *fin lien org* | - |
| . | . | . | SENT | UNK |

# Étiquetage d'entités nommées

## Évaluation des performances de l'étiqueteur CRF

Résultats système *baseline* appliqué dans la campagne ESTER 2

| EN | AMOUNT | FONC | LOC | ORG | PERS | PROD | TIME | tous |
|---|---|---|---|---|---|---|---|---|
| Qté | 239 | 196 | 1215 | 1267 | 1108 | 58 | 1025 | 5123 |
| précision | 0,85 | 0,61 | 0,77 | 0,79 | 0,93 | 0,53 | 0,91 | 0,86 |
| rappel | 0,56 | 0,559 | 0,81 | 0,63 | 0,75 | 0,12 | 0,60 | 0,718 |
| F-Score | 0,68 | 0,58 | 0,79 | 0,70 | 0,84 | 0,20 | 0,73 | 0,78 |

Résultats système multilingue entrainé d'après Wikipédia

| EN | AMOUNT | FONC | LOC | ORG | PERS | PROD | TIME | tous |
|---|---|---|---|---|---|---|---|---|
| Qté | 239 | 196 | 1215 | 1267 | 1108 | 58 | 1025 | 5123 |
| précision | **0,90** | **0,98** | **0,77** | **0,92** | **0,92** | **0,32** | **0,97** | **0,87** |
| rappel | 0,74 | 0,45 | 0,89 | 0,60 | 0,93 | 0,35 | 0,62 | 0,73 |
| F-Score | 0,81 | 0,62 | 0,82 | 0,73 | 0,93 | 0,33 | 0,76 | 0,80 |

# Détection de co-références

EN et POS

```
Tout      PRO:IND UNK
à         NOM     UNK
coup      NOM     UNK
un        DET:ART UNK
éclair    NOM     UNK
frappe    VER:presUNK
Doc       FNAM    PERS
et        KON     UNK
la        DET:ART UNK
DeLorean  NAM     PROD
de        PRP     UNK
plein     ADJ     UNK
fouet     NOM     UNK
et        KON     UNK
la        DET:ART UNK
machine   NOM     UNK
disparaît ADJ     UNK
.         SENT    UNK
```

1. Règle de détection des pronoms et des articles co-référents avec une entité
2. Association des entités nommées entre elles par système de pile
3. Algorithme de Hobb pour la mise en relation des entités co-référentes subsistantes [Hobb1978]

Co-références

```
Tout      PRO:IND UNK
à         NOM     UNK
coup      NOM     UNK
un        DET:ART UNK
éclair    NOM     UNK
frappe    VER:presUNK
Doc       FNAM    PERS.O
et        KON     UNK
la        DET:ART UNK
DeLorean  NAM     PROD.1
de        PRP     UNK
plein     ADJ     UNK
fouet     NOM     UNK
et        KON     UNK
la        DET:ART UNK
machine   NOM     PROD.1
disparaît ADJ     UNK
.         SENT    UNK
```

## Validation

Système présenté sous le nom de Poly-co lors de la campagne d'évaluation GREC2010 de la conférence INLG2010

# Détection de co-références

Résultats obtenus sur le corpus de test lors de la campagne GREC 2010.

| Système | Moyenne | B-3 | CEAF | MUC |
|---|---|---|---|---|
| UDel-NER | 72.71 | **80.51** | **77.53** | **60.09** |
| Poly-co | 66.99 | **76.92** | **70.29** | **53.77** |
| LingPipe | 58.23 | **71.19** | **61.58** | **41.92** |
| OpenNLP | 54.03 | **67.61** | **67.61** | **33.52** |

Résultats obtenus sur le corpus de développement de la campagne GREC 2010.

| Score | B3 | | | CEAF | | | MUC | | |
|---|---|---|---|---|---|---|---|---|---|
| Set | Préc. | Rappel | F-Score | Préc. | Rappel | F-Score | Préc. | Rappel | F-Score |
| Full set | 91.48 | 85.89 | **88,60** | 85.40 | 85.40 | **85.40** | 92.15 | 86.95 | **89.47** |
| Chef | 91.12 | 87.84 | **89.45** | 86.53 | 86.53 | **86.53** | 91.86 | 88.55 | **90.18** |
| Composers | 92.01 | 87.14 | **89.51** | 86.87 | 86.87 | **86.87** | 92.11 | 87.02 | **89.49** |
| Inventors | 91.27 | 82.63 | **86.74** | 82.73 | 82.73 | **82.73** | 92.48 | 85.29 | **88.74** |

# Analyse sémantique

Plusieurs propositions:

- Utilisation de détecteur de syntagmes associés à des règles
- Utilisation d'analyseur en dépendances associées à des règles [Zouaq2009]
- Utilisation d'analyseurs sémantiques de CoNLL2008 [Hajik2009]

## Méthodes retenues

- Pour l'analyse en Anglais, analyseur sémantique LTH [Johansson2008]
- Pour l'analyse en Français et Espagnol, analyseur en dépendance DeSR [Attardi2006] et règles

# Génération d'un Corpus de phrases modèles

# Application des étiqueteurs aux phrases du corpus

- Morphosyntaxe : nature des mots, temps de verbes
- Étiqueteur d'entités nommées : détection des entités
- Détecteur de co-références : identification des entités reliées
- Analyseur sémantique : relations logiques dans les phrases

Niveaux d'étiquetage disponibles dans CPM

| $x$ | $C$ | $CPM_{POS}$ | $CPM_{EN.COREF}$ | $CPM_{SEM}$ |
|---|---|---|---|---|
| 0 | Il | PRO:PER | PERS.0 | $s.0.p0$ |
| 1 | est | VER:pre | UNK | $s.0$ |
| 2 | parti | VER:pper | UNK | $s.0.D(p0)$ |
| 3 | , | PUN | UNK | $s.0$ |
| 4 | Henri | FNAM | PERS.0 | $s.0$ |
| 5 | , | PUN | UNK | $s.0$ |
| 6 | assassiné | NOM | UNK | $s.0.A(p1, p0)$ |
| 7 | par | PRP | UNK | $s.0$ |
| 8 | Ravaillac | NAM | PERS.1 | $s.0.p1$ |
| 9 | . | SENT | UNK | $s.0$ |

# Quantité de phrases étiquetables obtenues d'après trois éditions linguistiques de Wikipédia

| Langage | Corpus original $C_{d2}$ |
|---|---|
| Anglais | 74 711 331 |
| Français | 10 008 100 |
| Espagnol | 4 900 862 |

| Nombre de verbes | Nombre de phrases de $n$ verbes |
|---|---|
| 1 verbe | 1763010 |
| 2 verbes | 1263851 |
| 3 verbes | 876742 |
| 4 verbes | 576251 |
| 5 verbes | 363278 |
| 6 verbes | 218948 |
| 7 verbes | 129663 |
| 8 verbes | 77279 |
| 9 à 19 verbes | 45637 à 710 |

# Méthode de génération d'après CPM

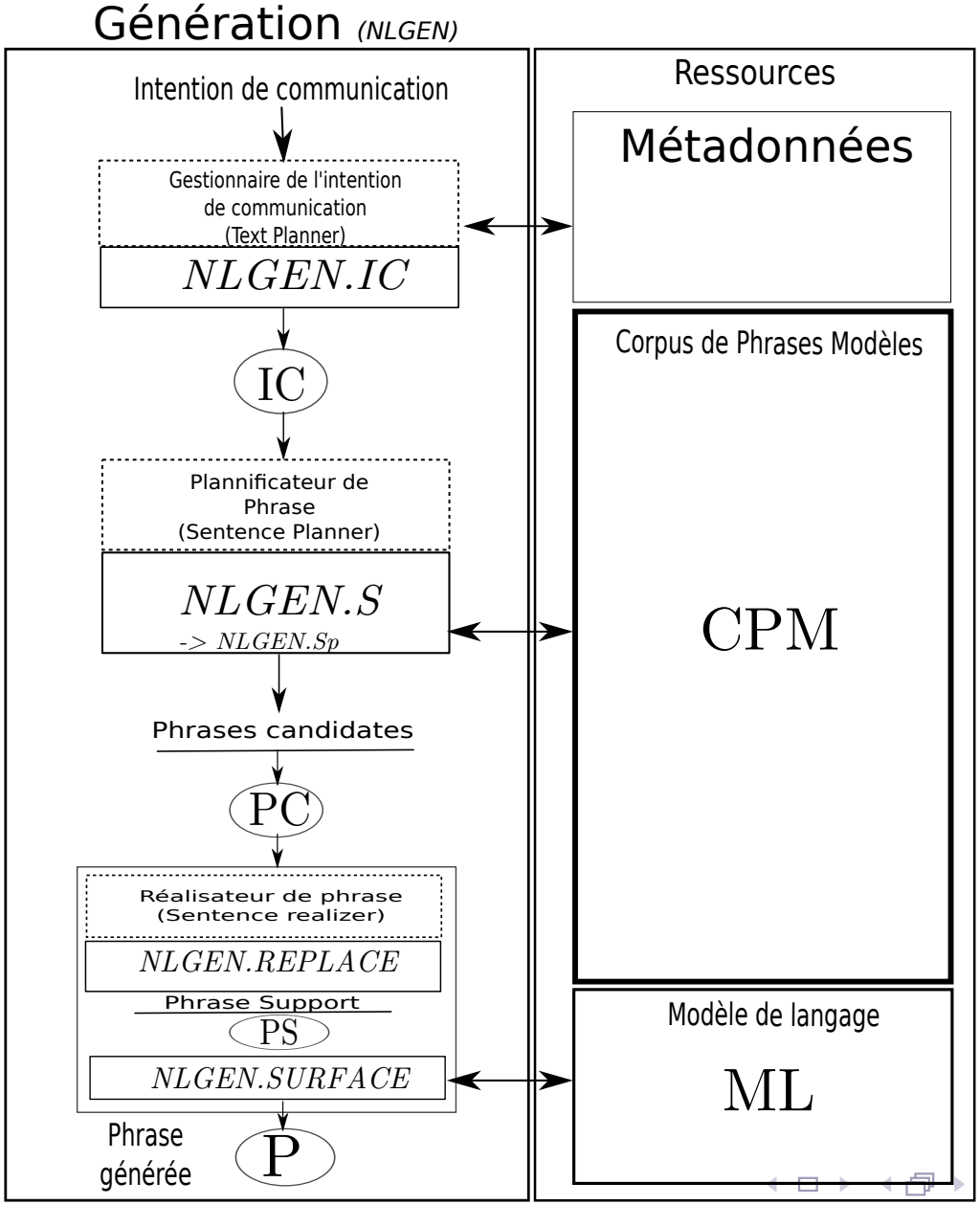
# Processus de génération

## Fonctions de l'algorithme de génération

1. *NLGEN.IC*:prépare l'intention de communication en la lexicalisant
2. *NLGEN.S*: détermine pour *IC* l'ensemble de Phrases Candidates *PC*, $PC \in CPM$, la Phrase Support $PS \in PC \rightarrow PS \sim IC$
3. *NLGEN.REPLACE*: modifie les informations abstraites contenues dans *PS* et les remplace par celles contenues dans *IC*
4. *NLGEN.SURFACE*: utilise un modèle de langage *ML* appris sur *C* pour corriger la surface de *PS*

## Fonctions de l'algorithme de recherche de phrases *NLGEN.S*

- *NLGEN.S.RTC* : mesure la compatibilité de temps
- *NLGEN.S.RL* : mesure la compatibilité lexicale par similarité cosinus
- *NLGEN.S.RS* : vérifie la compatibilité de forme logique


Génération (NLGEN)
Intention de communication
Gestionnaire de l'intention de communication (Text Planner)
NLGEN.IC
IC
Plannificateur de Phrase (Sentence Planner)
NLGEN.S
-> NLGEN.Sp
Phrases candidates
PC
Réalisateur de phrase (Sentence realizer)
NLGEN.REPLACE
Phrase Support
PS
NLGEN.SURFACE
Phrase générée
P
Ressources
Métadonnées
Corpus de Phrases Modèles
CPM
Modèle de langage
ML

# Recherche de la meilleure Phrase Support PS

*NLGEN.S.RTC*, *NLGEN.S.RL*, *NLGEN.S.RS* retournent une valeur pour chaque Phrase Candidate PC:

- $x = NLGEN.S.RS$ et $x \in \{0, 1\}$ (compatibilité de temps)
- $y = NLGEN.S.RL$ et $y \in [0, 1]$ (compatibilité lexicale)
- $z = NLGEN.S.RTC$ et $z \in \{0, 1\}$ (compatibilité logique)

Le produit des valeurs retournés par *NLGEN.S.RS*, *NLGEN.S.RL* et *NLGEN.S.RTC* obtenu pour chaque PC, donne une liste triée:

$$PS = \arg \max_{PC_{r=1\ldots p}} (PC_r x * PC_r y * PC_r z) \quad (5)$$

PS est la proposition de rang 1

# Transformation de la Phrase Support pour exprimer l'IC

Exemple d'IC : battu(Candidat,2012)

- *NLGEN.S*: si *NLGEN.S* retourne $PC = \emptyset$, un algorithme de repli *NLGEN.Sp* est activé

Exemple de PS : Le FONC n'a pas été réélu le DATE.

- *NLGEN.REPLACE*: modifie les informations abstraites contenues dans *PS* et les remplace par celles contenues dans *IC*

PS transformée : Le candidat n'a pas été réélu **le** 2012

- *NLGEN.SURFACE*: utilise le modèle de langage pour corriger la surface de *PS*

Exemple généré : Le candidat n'a pas été réélu en 2012

# NLGEN.sp: méthode de replis avec des structures SVO

On adopte une méthode de replis pour les cas ou $|PC| = 0$

## Les IC sont décomposées en séquences sujet verbe objet

- La probabilité de trouver une structure $\{sujet, verbe, objet\}$ est plus élevée
- On utilise un CPM composé uniquement de séquences SVO
- L'agrégateur de [Harbusch2009] est activé pour construire la phrase

Inconvénients: il faut écrire un agrégateur pour chaque langue. Structure stylistique proche de celle d'un générateur à base de règles.

# Évaluation des performances du système

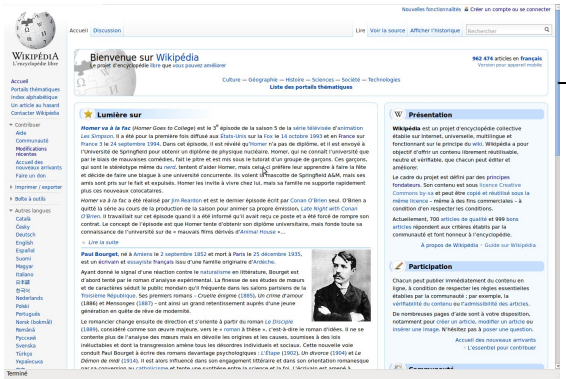
On souhaite mesurer:

1. La capacité du système à localiser une *phrase modèle (PM)* dans le *Corpus de Phrases Modèles (CPM)* quelle que soit la complexité de IC
2. La capacité du système à localiser *n* séquence *SVO* dans le *Corpus de Phrases Modèles SVO ($CPM_{SVO}$)* pour reconstruire par agrégation une IC
3. La qualité sémantique de la phrase finalement générée pour une IC donnée

# Protocole expérimental

1. Extraction aléatoire de $n$ phrases d'un corpus: les *Phrases d'Origine (PO)*
2. Une $IC_n$ est construite pour chaque *PO*
3. IC soumises aux deux générateurs *NLGEN.s* et *NLGEN.sp*
4. La qualité de la phrase générée est évaluée par comparaison avec *PO*

# Élaboration d'une IC



Random

1:La commune gère le terrain d' aviation de Chartres.
2:L' ARP est alors le premier parti des Pays-Bas.
3:La population est estimée en 2004 à 7882 habitants.
[...]

**IC de test pour la phrase 1**

IC_1
{ IC_L=[v(commune|ville|village);
A(gèr|admin;terrain|aéroport|aérodrome);
l(LOC.ADMI=Chartres)]
IC_S=[A(v,l)]
IC_C=[VER:pres]
}
{REF_1=*La commune gère le terrain d' aviation de Chartres.*}

# Métrique d'évaluation du processus de RI

But: vérifier la capacité du système à retrouver une phrase et que cette phrase soit sémantiquement adaptable à *IC*.

## Proposition: adapter la mesure de précision rappel

- Utiliser le rappel pour mesurer la capacité du système à retrouver une phrase d'après la forme logique d'une IC
- Utiliser la précision pour mesurer la capacité du CPM à retourner une PS valable pour représenter une IC

$$Rappel_{IC} = \frac{nombre\ de\ PS = PO}{|IC|} \tag{6}$$

$$Précision_{IC} = \frac{nombre\ de\ PS \equiv IC(\text{vérifiées par un juge humain})}{|PS|(\text{considérées correctes par le système})} \tag{7}$$

$$F - Score = \frac{2 \cdot (\text{Précision}_{\text{IC}} \cdot \text{Rappel}_{\text{IC}})}{(\text{Précision}_{\text{IC}} + \text{Rappel}_{\text{IC}})} \tag{8}$$

# Métrique d'évaluation du processus de génération

## Objectifs

Évaluer la capacité du système à adapter une Phrase Support valide pour qu'elle exprime une intention de communication

## Un juge humain attribue un score

- Une valeur de 0 à 2 est attribuée pour décrire la **qualité syntaxique**.
- Une valeur de 0 à 2 est attribuée pour décrire la **qualité sémantique**.

# Exemple d'apauvrissement sémantique (valeur de 1)

## Phrase source: Le fleuve traverse les communes de Palazzuolo sul Senio

- LOC.1;fleuve
- travers;franch;bord
- commune;ville;village;municipalité
- LOC.2=Palazzuolo sul Senio;

## Phrase Support proposée

- La ville est traversée par la LOC.1 , un des principaux fleuves d' LOC.2 ;
- La ville est traversée par la Clyde , un des principaux fleuves d' écosse

## Phrase produite

- La ville est traversée par le fleuve , un des principaux fleuves de Palazzuolo sul Senio

# Exemple de transformation conforme (valeur de 2)

## Phrase source: Capçanes est une commune de la province de Tarragone

- LOC.1=Capçanes
- commune;ville;village;municipalité
- province;région
- LOC.2=Tarragone

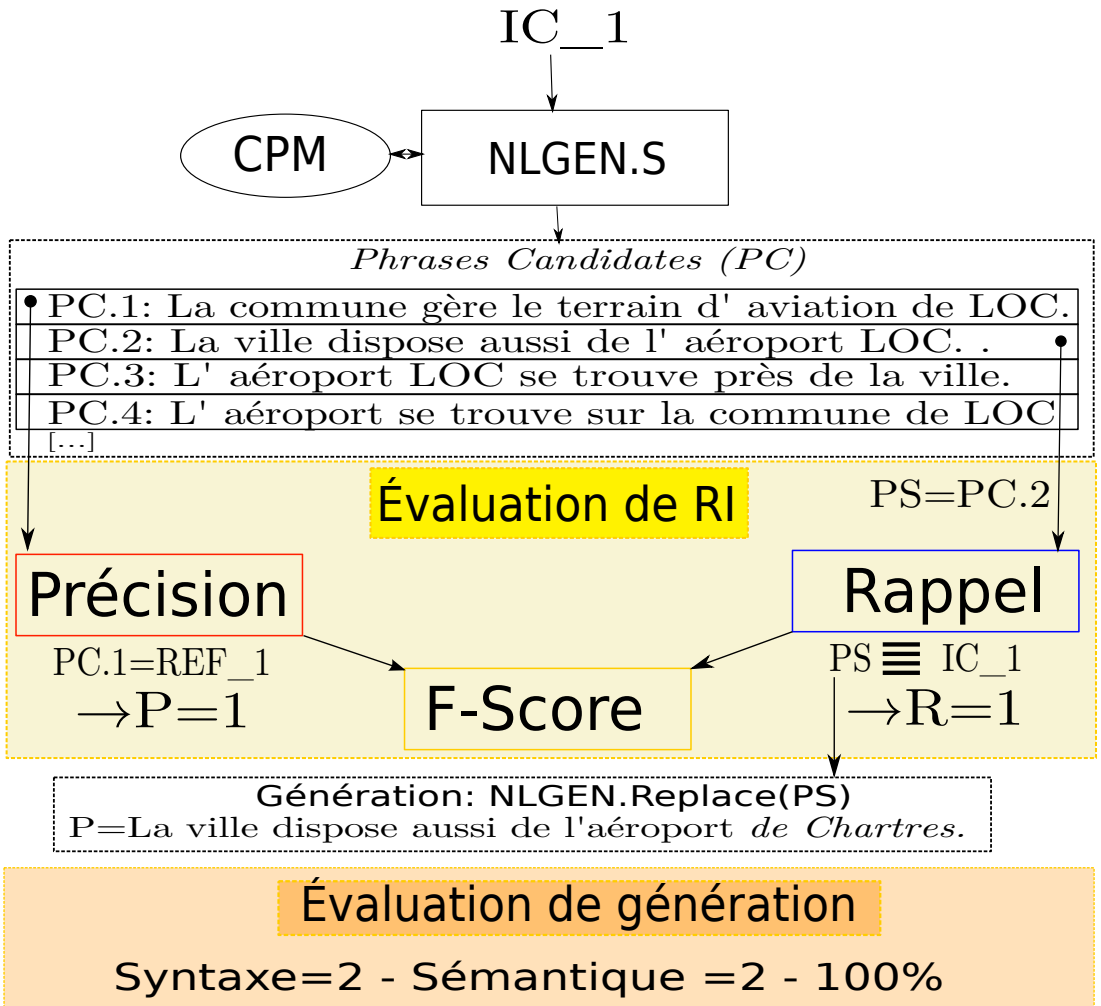
## Phrase Support proposée

- LOC.1 est une commune de la province d' LOC.2 , LOC.3 , en LOC.4 ;
- Rà¡gol est une commune de la province d' Almeria , Andalousie , en Espagne

## Phrase produite

- Capçanes est une commune de la province de Tarragone

# Algorithme de mesure complet



# Résultats

# Résultats de la RI

Résultats de recherche obtenus par *NLGEN.S* sur la totalité des *IC*

| Langue | Rappel | Précision | F-Score | *IC* |
|---|---|---|---|---|
| Français | 82 | 54 | 65 | 50 |
| Anglais | 91 | 61 | 73 | 50 |
| Espagnol | 71 | 43 | 53 | 25 |

Résultats de recherche obtenus par *NLGEN.Sp* sur chaque élément de la totalité des *IC* décomposées en *SVO*

| Langue | Rappel | Précision | F-Score | $IC_{SVO}$ |
|---|---|---|---|---|
| Français | 91 | 92 | 90 | 148 |
| Anglais | 92 | 94 | 91 | 162 |
| Espagnol | 81 | 84 | 82 | 84 |

# Résultats de génération pour les phrases complètes

Résultats de génération obtenus par *NLGEN.replace* et *NLGEN.surface* pour des phrases complètes extraites par *NLGEN.S*.

| | Syntaxe | (%) | | Sémantique | (%) | |
|---|---|---|---|---|---|---|
| Langue | Erronée | Erreurs | Correcte | Erroné | Ambigu | Conforme |
| Français | 0 | 2 | 98 | 8 | 14 | 78 |
| Anglais | 0 | 3 | 97 | 6 | 17 | 77 |
| Espagnol | 5 | 10 | 85 | 10 | 30 | 60 |

Résultats de génération obtenus avec agrégation de *SVO* par *NLGEN.agg*.

| | Syntaxe | (%) | | Sémantique | (%) | |
|---|---|---|---|---|---|---|
| Langue | Erronée | Erreurs | Correcte | Erroné | Ambigu | Conforme |
| Français | 2 | 8 | 90 | 5 | 14 | 81 |
| Anglais | 1 | 6 | 93 | 11 | 10 | 79 |
| Espagnol | 4 | 16 | 80 | 11 | 24 | 65 |

# Conclusions

Nous avons présenté un système de Génération Automatique de Texte (GAT) fonctionnant entièrement avec des méthodes statistiques

- Il utilise un Générateur de Surface exploitant des phrases préexistantes, contenues dans un Corpus de Phrases Modèles (CPM)
- Il fonctionne en trois langues
- Le CPM exploite des ressources lexicales et un système d'étiquetage et d'analyse de grand corpus

Nous avons montré qu'il était possible d'exprimer une IC complexe en transformant une phrase existante

- La rareté de ces phrases et le manque de précision des analyseurs sémantiques ne permet pas encore de répondre à tous les cas
- La méthode décrite ne prend pas en compte tous les aspects linguistiques (négation, modes (conditionnel, impératif, propositions multiples)
- Les erreurs introduites dans les phrases générées sont dues à des défectuosités mineures des composants d'analyse et d'étiquetage.

Notre proposition possède un potentiel de progression, particulièrement lié aux évolutions futures des méthodes d'analyse sémantique et des étiqueteurs.

# Perspectives

- L'amélioration de la fiabilité de l'analyseur sémantique introduit mécaniquement une augmentation des performances des algorithmes de génération. Il sera donc essentiel d'intégrer les dernières innovations en ce domaine et d'en mesurer l'influence.
- Nous souhaitons tenter de récupérer automatiquement une connaissance grammaticale (les morphologies, les conjugaisons de verbes) pour les intégrer dans un système de GAT classique.
- L'évaluation précise des capacités de génération dans des domaines sémantiques fermés devra être menée.

# Merci de votre attention

# Éléments complémentaires

# Exemple d'erreur syntaxique

## Phrase source: Dacia est le plus grand constructeur d'automobiles roumain.

- ORG
- fabriqu;constru;assembl
- automobi
- LOC

## Transformations successives de la Phrase Support proposée

1. NLGEN.S : Proto Motors est une marque de fabrique automobile Coréenne.
2. NLGEN.S : ORG est une marque de fabrique automobile LOC.
3. NLGEN.REPLACE : *DACIA* est une marque de fabrique automobile *Roumain*.
4. ML: *DACIA* est une marque de fabrique automobile *Roumain*.

# Aggrégation 1 : légère erreur de syntaxe

S=Jules Verne est populaire dans le monde entier et, selon l'Index Translationum, avec un total de 4162 traductions, il vient au deuxième rang des auteurs les plus traduits en langue étrangère après Agatha Christie.

[même verbe][g]Jules Vernes est populaire dans le monde entier, Jules Vernes traduit 4162 fois selon l'Index Translationum, Jules Vernes situé au deuxième rang des auteurs les plus traduits et Agatha Christie au premier rang.

# Aggrégation 2: déviation de sens

S=Une exoplanète, ou planète extrasolaire, est une planète qui orbite autour d'une étoile autre que le Soleil.

[même sujet][même verbe][f]Une planète extrasolaire est une planète et en orbite autour d'une étoile autre que le Soleil.

# Génération: déviation de sens

< *seq* > *Sequence* = 4 < /*seq* >
< *ic* >p= Le PowerBook 180 est un ordinateur portable d' Apple< /*ref* >< /*ic* >

-**Sortie du PROD PROD , conçu par la firme ORG**; Sortie du Gamma 60 , conçu par la firme Bull
-En TIME ORG lance une gamme d' ordinateurs PROD PROD , les ORG PROD ; En 1985 Tandy lance une gamme d' ordinateurs compatibles IBM , les Tandy 1000
-En TIME sort la version originale du jeu sur le micro-ordinateur 8-bits ORG PROD ; En 1989 sort la version originale du jeu sur le micro-ordinateur 8-bits Apple II

< *found* > 3 < /*found* >

Sortie du Powerbook 180, conçu par la firme Apple.

# Sens conforme

< *seq* > *Sequence* = 18 < /*seq* >
< *ic* >p= La population est estimée en 2004 à 7882 habitants < /*ic* >
-**La population est estimée à AMOUNT AMOUNT AMOUNT AMOUNT en TIME** ( ORG ) ; La population est estimée à 1 253 637 habitants en 2004 ( INSEE )
-Au TIME TIME TIME , sa population était estimée à AMOUNT AMOUNT 000 habitants ; Au premier janvier 2006 , sa population était estimée à 4 781 000 habitants
-Au recensement de la population en TIME , la ville comptait 12 AMOUNT AMOUNT ; Au recensement de la population en 2000 , la ville comptait 12 626 habitants
-Selon l' estimation officielle de TIME , sa population est de 15 AMOUNT AMOUNT ; Selon l' estimation officielle de 2005 , sa population est de 15 327 habitants
-La population taà¯wanaise était estimée à AMOUNT AMOUNT AMOUNT d' habitants en TIME TIME ; La population taà¯wanaise était estimée à 22 911 292 d' habitants en juillet 2007
...
< *found* > 10 < /*found* >

La population est estimée à 7882 habitants en 2004.

# DBpedia Versus Wikipedia

- DBpedia ne gère pas l'aspect synonymique (uniquement les redirections)
- DBpedia reprends les 100 k classes Française (600k anglaises) et ne permet pas d'utiliser une taxonomie restreinte
- L'absence de taxonomie restreinte rend impossible la réutilisation des liens internes pour générer des corpus d'apprentissage
- DBpedia est basé sur l'anglais et n'intègre pas la totalité des éléments français et espagnols (amélioration en 2010)