

Implication Textuelle et Réécriture

Soutenance de thèse

Paul Bédaride

UHP Nancy/Loria

18 octobre 2010

Vandœuvre

- **Pourquoi ?**

Tâche de base pour les systèmes :

- d'extraction d'information
- de recherche d'information
- de question-réponse
- de synthèse de documents

- **Définition :**

Capacité humaine à déduire une hypothèse à partir d'un texte

- **Exemples :**

Texte		Hypothèses
Le ministre a quitté Bagdad.	⇒	Le ministre était à Bagdad.
	⇒	Le ministre n'est pas à Bagdad.
	≠	Le ministre est à Paris.

- ① Introduction
 - Implication Textuelle
 - Approches existantes
 - Réécriture
- ② Afazio : un système basé sur la réécriture
 - Architecture du système
 - Le calcul sémantique
 - Exemple
- ③ Évaluation sur des suites de tests contrôlés
- ④ Vers une analyse plus approfondie des systèmes
 - Motivation
 - ARTE : un schéma d'annotation fin pour le RTE
 - Fouille d'erreurs
- ⑤ Conclusion
 - Contributions de la thèse
 - Perspectives

Recognising Textual Entailment Challenge

- Introduit par Ido Dagan et Oren Glickman en 2006
- Un nouveau challenge chaque année
- Suite de développement composée de 800 problèmes
- Suite de tests composée de 800 problèmes
- Distribution uniforme des implications et des non-implications
- Problèmes construits à partir de systèmes :
 - d'extraction d'information
 - de recherche d'information
 - de question-réponse
 - de synthèse de documents
- Campagne du RTE 2007 :
 - 23 systèmes avec 41 soumissions
 - Exactitude (pourcentage de réponses correctes) entre 54% et 75%

- **Texte** : Depuis qu'il a vu le jour en 2004, Katamari Damacy a continué de devenir un des plus grands succès de l'histoire des jeux vidéos.
- **Hypothèse** : Katamari Damacy est sorti en 2004.
- **Réponse** : Oui
- **Système** : Extraction d'information

- Implications basées sur des connaissances syntaxiques :
Jean achète un vélo. \Rightarrow Un vélo est acheté par Jean.
Jean achète un vélo. \Rightarrow C'est Jean qui achète un vélo.
- Implications basées sur des connaissances lexicales :
Un chat mange une pomme. \Rightarrow Un animal mange un fruit.
Le verre est vide. \Rightarrow Le verre n'est pas plein.
- Implications basées sur des connaissances encyclopédiques :
Jean a vu Edith Piaf au Zénith. \Rightarrow Edith Piaf a chanté à Nancy.
- Implications basées sur les quantificateurs :
Tous les animaux mangent des fraises.
 \Rightarrow Tous les chats mangent des fruits.
Aucun animal ne mange des fruits.
 \Rightarrow Aucun chat ne mange des fraises.

- Implication Textuelle (RTE)

Capacité humaine à déduire une hypothèse à partir d'un texte

- Implication Logique (Afazio, NutCracker)

L'hypothèse doit être vraie dans **tous les mondes** où le texte est vrai

- Exemple :

Texte : Environ deux jours avant la conférence,

Marie était dans le bureau de Jean à Nancy.

Hypothèse : Jean travaille à Nancy.

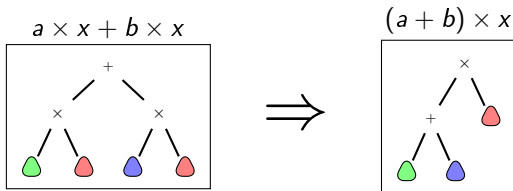
Implication Textuelle : **Oui**

Implication Logique : **Non**

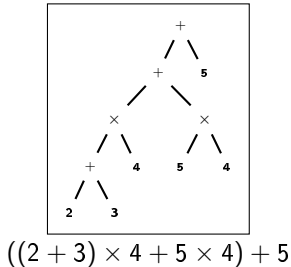
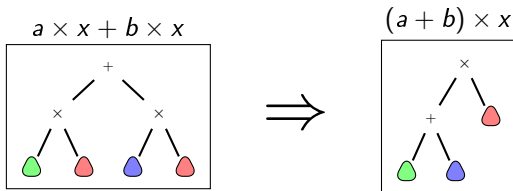
- Chevauchements de mots [Valentin Jijkoun and Maarten de Rijke]
 - $$\frac{\# \text{ lemmes en commun du Texte et de l'Hypothèse}}{\# \text{ lemmes de l'Hypothèse}} > \text{seuil}$$
- Alignement de structures linguistiques [Milen Kouylekov and Bernardo Magnini]
 - Trouver la **séquence d'éditions** permettant d'obtenir la **structure syntaxique** de l'**Hypothèse** à partir de celle du **Texte**
 - La **somme des scores** des éditions ne doit pas dépasser un certain **seuil**
- Méthode symbolique [Johan Bos] (Afazio)
 - Dérivation de la **représentation sémantique** du **Texte** et de l'**Hypothèse** à partir de leurs **analyses syntaxiques**
 - Test de l'**implication** entre les sémantiques à l'aide d'**outils de preuve**

Réécriture

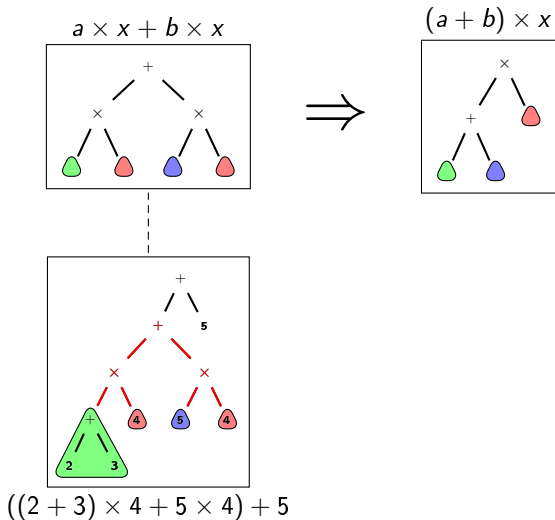
Exemple : la factorisation de formules arithmétiques



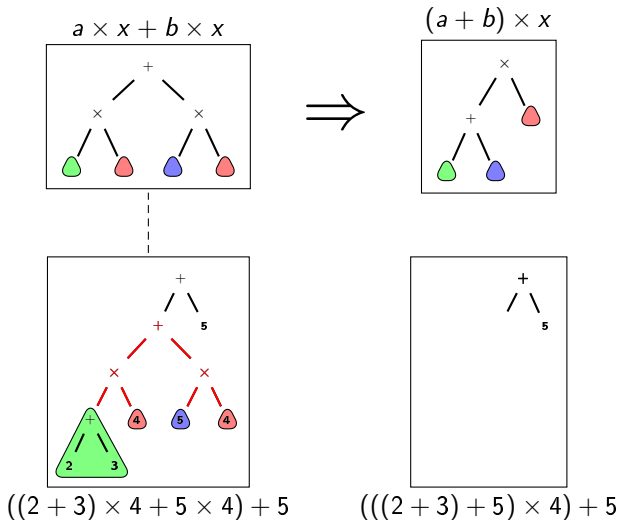
Exemple : la factorisation de formules arithmétiques



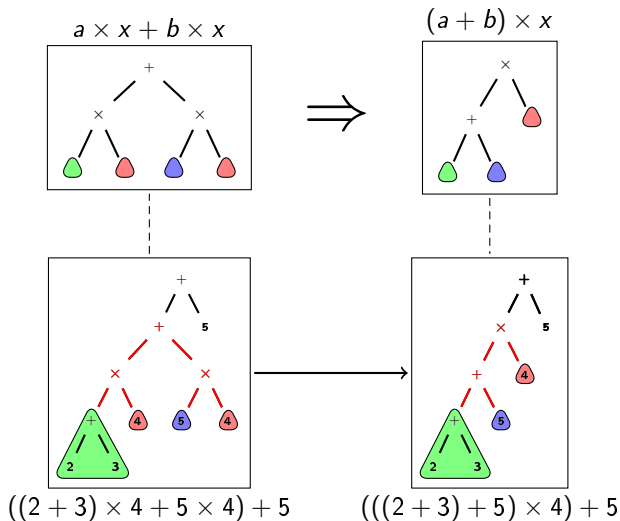
Exemple : la factorisation de formules arithmétiques



Exemple : la factorisation de formules arithmétiques



Exemple : la factorisation de formules arithmétiques



- **Modèle de calcul** permettant d'appliquer des **transformations** sur des objets
- Un **ensemble** de règles de réécriture forme un **système de réécriture**
- Un système de réécriture peut ne pas satisfaire :
 - la propriété de confluence
 - la propriété de terminaison
- La **stratégie** d'application des règles est **importante** car elle permet de forcer la **confluence** et la **terminaison** d'un système.

- **Modèle de calcul** permettant d'appliquer des **transformations** sur des objets
- Un **ensemble** de règles de réécriture forme un **système de réécriture**
- Un système de réécriture peut ne pas satisfaire :
 - la propriété de confluence
 - la propriété de terminaison
- La **stratégie** d'application des règles est **importante** car elle permet de forcer la **confluence** et la **terminaison** d'un système.
- Exemple :
 - Système de réécriture :

01	\Rightarrow_{r_1}	0
10	\Rightarrow_{r_2}	1
0	\Rightarrow_{r_3}	00

- **Modèle de calcul** permettant d'appliquer des **transformations** sur des objets
- Un **ensemble** de règles de réécriture forme un **système de réécriture**
- Un système de réécriture peut ne pas satisfaire :
 - la propriété de confluence
 - la propriété de terminaison
- La **stratégie** d'application des règles est **importante** car elle permet de forcer la **confluence** et la **terminaison** d'un système.
- Exemple :
 - Système de réécriture :

01	\Rightarrow_{r_1}	0
10	\Rightarrow_{r_2}	1
0	\Rightarrow_{r_3}	00
 - Application du système sur le mot 1101

- **Modèle de calcul** permettant d'appliquer des **transformations** sur des objets
- Un **ensemble** de règles de réécriture forme un **système de réécriture**
- Un système de réécriture peut ne pas satisfaire :
 - la propriété de confluence
 - la propriété de terminaison
- La **stratégie** d'application des règles est **importante** car elle permet de forcer la **confluence** et la **terminaison** d'un système.
- Exemple :
 - Système de réécriture :
$$\begin{array}{lll} 01 & \Rightarrow_{r_1} & 0 \\ 10 & \Rightarrow_{r_2} & 1 \\ 0 & \Rightarrow_{r_3} & 00 \end{array}$$
 - Application du système sur le mot 1101
 - r_1, r_2, r_3 : 1101

- **Modèle de calcul** permettant d'appliquer des **transformations** sur des objets
- Un **ensemble** de règles de réécriture forme un **système de réécriture**
- Un système de réécriture peut ne pas satisfaire :
 - la propriété de confluence
 - la propriété de terminaison
- La **stratégie** d'application des règles est **importante** car elle permet de forcer la **confluence** et la **terminaison** d'un système.
- Exemple :
 - Système de réécriture :
$$\begin{array}{lll} 01 & \Rightarrow_{r_1} & 0 \\ 10 & \Rightarrow_{r_2} & 1 \\ 0 & \Rightarrow_{r_3} & 00 \end{array}$$
 - Application du système sur le mot 1101
 - r_1, r_2, r_3 : 11**0**1

- **Modèle de calcul** permettant d'appliquer des **transformations** sur des objets
- Un **ensemble** de règles de réécriture forme un **système de réécriture**
- Un système de réécriture peut ne pas satisfaire :
 - la propriété de confluence
 - la propriété de terminaison
- La **stratégie** d'application des règles est **importante** car elle permet de forcer la **confluence** et la **terminaison** d'un système.
- Exemple :
 - Système de réécriture :
$$\begin{array}{lll} 01 & \Rightarrow_{r_1} & 0 \\ 10 & \Rightarrow_{r_2} & 1 \\ 0 & \Rightarrow_{r_3} & 00 \end{array}$$
 - Application du système sur le mot 1101
 - $r_1, r_2, r_3 : 11\mathbf{0}1 \rightarrow_{r_1} 110$

- **Modèle de calcul** permettant d'appliquer des **transformations** sur des objets
- Un **ensemble** de règles de réécriture forme un **système de réécriture**
- Un système de réécriture peut ne pas satisfaire :
 - la propriété de confluence
 - la propriété de terminaison
- La **stratégie** d'application des règles est **importante** car elle permet de forcer la **confluence** et la **terminaison** d'un système.
- Exemple :
 - Système de réécriture :
$$\begin{array}{lll} 01 & \Rightarrow_{r_1} & 0 \\ 10 & \Rightarrow_{r_2} & 1 \\ 0 & \Rightarrow_{r_3} & 00 \end{array}$$
 - Application du système sur le mot 1101
 - $r_1, r_2, r_3 : 11\mathbf{01} \rightarrow_{r_1} 1\mathbf{10}$

- **Modèle de calcul** permettant d'appliquer des **transformations** sur des objets
- Un **ensemble** de règles de réécriture forme un **système de réécriture**
- Un système de réécriture peut ne pas satisfaire :
 - la propriété de confluence
 - la propriété de terminaison
- La **stratégie** d'application des règles est **importante** car elle permet de forcer la **confluence** et la **terminaison** d'un système.
- Exemple :
 - Système de réécriture :
$$\begin{array}{lcl} 01 & \Rightarrow_{r_1} & 0 \\ 10 & \Rightarrow_{r_2} & 1 \\ 0 & \Rightarrow_{r_3} & 00 \end{array}$$
 - Application du système sur le mot 1101
 - $r_1, r_2, r_3 : 11\mathbf{0}1 \rightarrow_{r_1} 1\mathbf{1}0 \rightarrow_{r_2} 11$

- **Modèle de calcul** permettant d'appliquer des **transformations** sur des objets
- Un **ensemble** de règles de réécriture forme un **système de réécriture**
- Un système de réécriture peut ne pas satisfaire :
 - la propriété de confluence
 - la propriété de terminaison
- La **stratégie** d'application des règles est **importante** car elle permet de forcer la **confluence** et la **terminaison** d'un système.
- Exemple :
 - Système de réécriture :
$$\begin{array}{lll} 01 & \Rightarrow_{r_1} & 0 \\ 10 & \Rightarrow_{r_2} & 1 \\ 0 & \Rightarrow_{r_3} & 00 \end{array}$$
 - Application du système sur le mot 1101
 - $r_1, r_2, r_3 : 11\mathbf{0}1 \rightarrow_{r_1} 1\mathbf{1}0 \rightarrow_{r_2} 11$
 - $r_2, r_1, r_3 : 1101$

- **Modèle de calcul** permettant d'appliquer des **transformations** sur des objets
- Un **ensemble** de règles de réécriture forme un **système de réécriture**
- Un système de réécriture peut ne pas satisfaire :
 - la propriété de confluence
 - la propriété de terminaison
- La **stratégie** d'application des règles est **importante** car elle permet de forcer la **confluence** et la **terminaison** d'un système.
- Exemple :
 - Système de réécriture :
$$\begin{array}{lll} 01 & \Rightarrow_{r_1} & 0 \\ 10 & \Rightarrow_{r_2} & 1 \\ 0 & \Rightarrow_{r_3} & 00 \end{array}$$
 - Application du système sur le mot 1101
 - $r_1, r_2, r_3 : 11\mathbf{01} \rightarrow_{r_1} 1\mathbf{10} \rightarrow_{r_2} 11$
 - $r_2, r_1, r_3 : 11\mathbf{01}$

- **Modèle de calcul** permettant d'appliquer des **transformations** sur des objets
- Un **ensemble** de règles de réécriture forme un **système de réécriture**
- Un système de réécriture peut ne pas satisfaire :
 - la propriété de confluence
 - la propriété de terminaison
- La **stratégie** d'application des règles est **importante** car elle permet de forcer la **confluence** et la **terminaison** d'un système.
- Exemple :
 - Système de réécriture :
$$\begin{array}{lll} 01 & \Rightarrow_{r_1} & 0 \\ 10 & \Rightarrow_{r_2} & 1 \\ 0 & \Rightarrow_{r_3} & 00 \end{array}$$
 - Application du système sur le mot 1101
 - $r_1, r_2, r_3 : 11\mathbf{01} \rightarrow_{r_1} 1\mathbf{10} \rightarrow_{r_2} 11$
 - $r_2, r_1, r_3 : 11\mathbf{01} \rightarrow_{r_2} 111$

- **Modèle de calcul** permettant d'appliquer des **transformations** sur des objets
- Un **ensemble** de règles de réécriture forme un **système de réécriture**
- Un système de réécriture peut ne pas satisfaire :
 - la propriété de confluence
 - la propriété de terminaison
- La **stratégie** d'application des règles est **importante** car elle permet de forcer la **confluence** et la **terminaison** d'un système.

- Exemple :

- Système de réécriture :

01	\Rightarrow_{r_1}	0
10	\Rightarrow_{r_2}	1
0	\Rightarrow_{r_3}	00

- Application du système sur le mot 1101

- $r_1, r_2, r_3 : 11\mathbf{01} \rightarrow_{r_1} 1\mathbf{10} \rightarrow_{r_2} 11$
- $r_2, r_1, r_3 : 11\mathbf{01} \rightarrow_{r_2} 111$
- $r_3, r_1, r_2 : 1101$

- **Modèle de calcul** permettant d'appliquer des **transformations** sur des objets
- Un **ensemble** de règles de réécriture forme un **système de réécriture**
- Un système de réécriture peut ne pas satisfaire :
 - la propriété de confluence
 - la propriété de terminaison
- La **stratégie** d'application des règles est **importante** car elle permet de forcer la **confluence** et la **terminaison** d'un système.

- Exemple :

- Système de réécriture :
$$\begin{array}{lll} 01 & \Rightarrow_{r_1} & 0 \\ 10 & \Rightarrow_{r_2} & 1 \\ 0 & \Rightarrow_{r_3} & 00 \end{array}$$

- Application du système sur le mot 1101

- $r_1, r_2, r_3 : 11\mathbf{0}1 \rightarrow_{r_1} 1\mathbf{1}0 \rightarrow_{r_2} 11$
- $r_2, r_1, r_3 : 11\mathbf{0}1 \rightarrow_{r_2} 111$
- $r_3, r_1, r_2 : 11\mathbf{0}1$

- **Modèle de calcul** permettant d'appliquer des **transformations** sur des objets
- Un **ensemble** de règles de réécriture forme un **système de réécriture**
- Un système de réécriture peut ne pas satisfaire :
 - la propriété de confluence
 - la propriété de terminaison
- La **stratégie** d'application des règles est **importante** car elle permet de forcer la **confluence** et la **terminaison** d'un système.

- Exemple :

- Système de réécriture :
$$\begin{array}{lll} 01 & \Rightarrow_{r_1} & 0 \\ 10 & \Rightarrow_{r_2} & 1 \\ 0 & \Rightarrow_{r_3} & 00 \end{array}$$

- Application du système sur le mot 1101

- $r_1, r_2, r_3 : 11\mathbf{0}1 \rightarrow_{r_1} 1\mathbf{1}0 \rightarrow_{r_2} 11$
- $r_2, r_1, r_3 : 11\mathbf{0}1 \rightarrow_{r_2} 111$
- $r_3, r_1, r_2 : 11\mathbf{0}1 \rightarrow_{r_3} 11001$

- **Modèle de calcul** permettant d'appliquer des **transformations** sur des objets
- Un **ensemble** de règles de réécriture forme un **système de réécriture**
- Un système de réécriture peut ne pas satisfaire :
 - la propriété de confluence
 - la propriété de terminaison
- La **stratégie** d'application des règles est **importante** car elle permet de forcer la **confluence** et la **terminaison** d'un système.

- Exemple :

- Système de réécriture :
$$\begin{array}{lll} 01 & \Rightarrow_{r_1} & 0 \\ 10 & \Rightarrow_{r_2} & 1 \\ 0 & \Rightarrow_{r_3} & 00 \end{array}$$

- Application du système sur le mot 1101

- $r_1, r_2, r_3 : 11\mathbf{0}1 \rightarrow_{r_1} 1\mathbf{1}0 \rightarrow_{r_2} 11$
- $r_2, r_1, r_3 : 11\mathbf{0}1 \rightarrow_{r_2} 111$
- $r_3, r_1, r_2 : 11\mathbf{0}1 \rightarrow_{r_3} 11\mathbf{0}01$

- **Modèle de calcul** permettant d'appliquer des **transformations** sur des objets
- Un **ensemble** de règles de réécriture forme un **système de réécriture**
- Un système de réécriture peut ne pas satisfaire :
 - la propriété de confluence
 - la propriété de terminaison
- La **stratégie** d'application des règles est **importante** car elle permet de forcer la **confluence** et la **terminaison** d'un système.

- Exemple :

- Système de réécriture :
$$\begin{array}{lll} 01 & \Rightarrow_{r_1} & 0 \\ 10 & \Rightarrow_{r_2} & 1 \\ 0 & \Rightarrow_{r_3} & 00 \end{array}$$

- Application du système sur le mot 1101
 - $r_1, r_2, r_3 : 11\mathbf{0}1 \rightarrow_{r_1} 1\mathbf{1}0 \rightarrow_{r_2} 11$
 - $r_2, r_1, r_3 : 11\mathbf{0}1 \rightarrow_{r_2} 111$
 - $r_3, r_1, r_2 : 11\mathbf{0}1 \rightarrow_{r_3} 11\mathbf{0}01 \rightarrow_{r_3} 110001$

- **Modèle de calcul** permettant d'appliquer des **transformations** sur des objets
- Un **ensemble** de règles de réécriture forme un **système de réécriture**
- Un système de réécriture peut ne pas satisfaire :
 - la propriété de confluence
 - la propriété de terminaison
- La **stratégie** d'application des règles est **importante** car elle permet de forcer la **confluence** et la **terminaison** d'un système.

- Exemple :

- Système de réécriture :
$$\begin{array}{lll} 01 & \Rightarrow_{r_1} & 0 \\ 10 & \Rightarrow_{r_2} & 1 \\ 0 & \Rightarrow_{r_3} & 00 \end{array}$$

- Application du système sur le mot 1101
 - $r_1, r_2, r_3 : 11\mathbf{0}1 \rightarrow_{r_1} 1\mathbf{1}0 \rightarrow_{r_2} 11$
 - $r_2, r_1, r_3 : 11\mathbf{0}1 \rightarrow_{r_2} 111$
 - $r_3, r_1, r_2 : 11\mathbf{0}1 \rightarrow_{r_3} 11\mathbf{0}01 \rightarrow_{r_3} 11\mathbf{0}001$

- **Modèle de calcul** permettant d'appliquer des **transformations** sur des objets
- Un **ensemble** de règles de réécriture forme un **système de réécriture**
- Un système de réécriture peut ne pas satisfaire :
 - la propriété de confluence
 - la propriété de terminaison
- La **stratégie** d'application des règles est **importante** car elle permet de forcer la **confluence** et la **terminaison** d'un système.

- Exemple :

- Système de réécriture :
$$\begin{array}{lll} 01 & \Rightarrow_{r_1} & 0 \\ 10 & \Rightarrow_{r_2} & 1 \\ 0 & \Rightarrow_{r_3} & 00 \end{array}$$

- Application du système sur le mot 1101

- $r_1, r_2, r_3 : 11\mathbf{0}1 \rightarrow_{r_1} 1\mathbf{1}0 \rightarrow_{r_2} 11$
- $r_2, r_1, r_3 : 11\mathbf{0}1 \rightarrow_{r_2} 111$
- $r_3, r_1, r_2 : 11\mathbf{0}1 \rightarrow_{r_3} 11\mathbf{0}01 \rightarrow_{r_3} 11\mathbf{0}001 \rightarrow_{r_3} \dots$

- La **confluence** des systèmes de réécriture vers la solution voulue est assurée par l'utilisation d'une **stratégie** de réécriture appliquant les règles les plus spécifiques en premier
 - Exemple : le passif long avant le passif court
- La stratégie de réécriture est calculée **automatiquement** en testant si les règles de réécriture se filtrent entre elles
 - Exemple : le passif court filtre le passif long et doit donc être appliqué après
- La **terminaison** est assurée par la définition de **niveaux de ressources**, et l'utilisation de règles **consommant des ressources** d'un niveau pour créer des ressources du niveau supérieur

- 1 Introduction
 - Implication Textuelle
 - Approches existantes
 - Réécriture
- 2 Afazio : un système basé sur la réécriture
 - Architecture du système
 - Le calcul sémantique
 - Exemple
- 3 Évaluation sur des suites de tests contrôlés
- 4 Vers une analyse plus approfondie des systèmes
 - Motivation
 - ARTE : un schéma d'annotation fin pour le RTE
 - Fouille d'erreurs
- 5 Conclusion
 - Contributions de la thèse
 - Perspectives

- **Entrée** : le **texte** et l'**hypothèse**
 - Analyse en **constituants** du texte et de l'hypothèse
 - Analyse en **dépendances**
 - Étiquetage des **rôle sémantiques**
 - Dérivation des **formules logiques**
 - Test de la **validité de l'implication** entre les formules associées au **Texte** et à l'**Hypothèse** à l'aide d'un **prouveur de théorèmes** et d'un **constructeur de modèles**
- | | | |
|--|---|------------------------|
| | } | Stanford |
| | } | Systèmes de réécriture |
| | } | Paradox Équinox |

- Logique du première ordre
- Représentation **néo-Davidsonienne** des verbes
- **Une seule** formule associée à une analyse
 - **Portée** des **quantificateurs** = **syntaxe**
- Calcul sémantique basé sur la **réécriture** à partir des structures linguistiques dérivées antérieurement (constituants et dépendances syntaxiques, étiquetage sémantique)

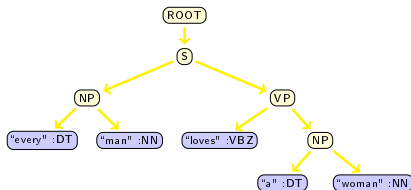
« Every man loves a woman »

- Texte

Exemple

- Texte
- Analyse en constituants

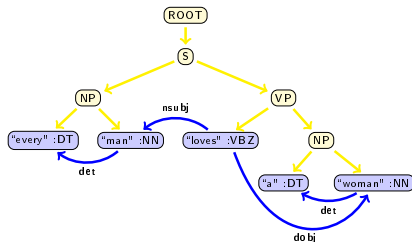
« Every man loves a woman »



Exemple

- Texte
- Analyse en constituants
- Dérivation des dépendances

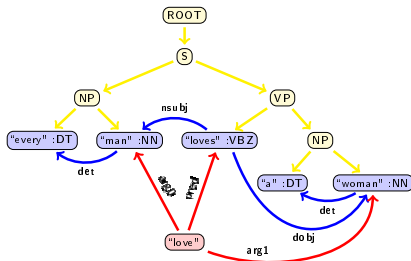
« Every man loves a woman »



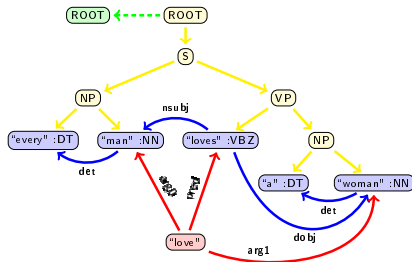
- Texte
- Analyse en constituants
- Dérivation des dépendances
- Étiquetage de rôles sémantiques

Réécriture

« Every man loves a woman »



« Every man loves a woman »

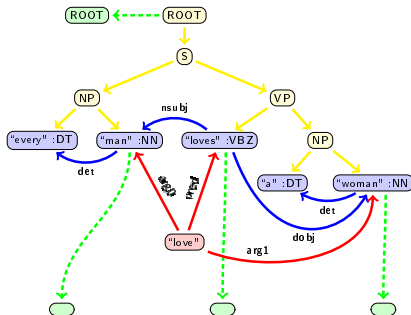


Réécriture

- Texte
- Analyse en constituants
- Dérivation des dépendances
- Étiquetage de rôles sémantiques
- Initialisation de la racine

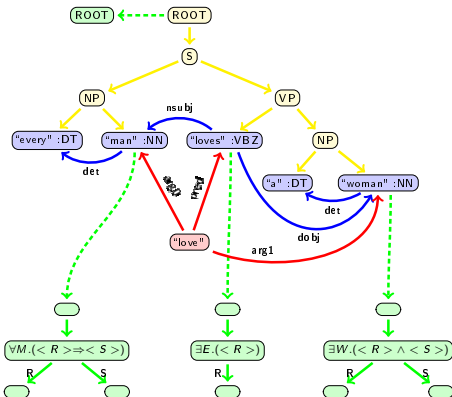
- Texte
- Analyse en constituants
- Dérivation des dépendances
- Étiquetage de rôles sémantiques
- Initialisation de la racine
- Initialisation des têtes

« Every man loves a woman »



- Texte
- Analyse en constituants
- Dérivation des dépendances
- Étiquetage de rôles sémantiques
- Initialisation de la racine
- Initialisation des têtes
- Création des fragments

« Every man loves a woman »

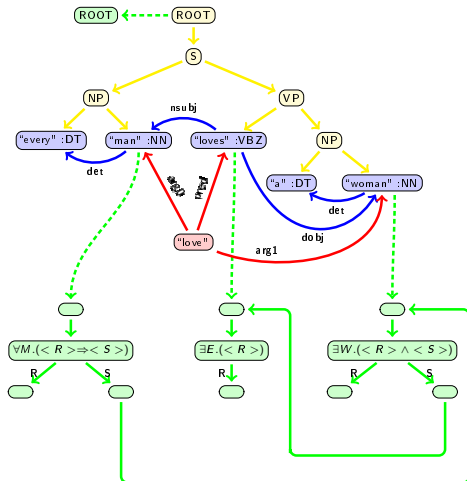


Exemple

Réécriture

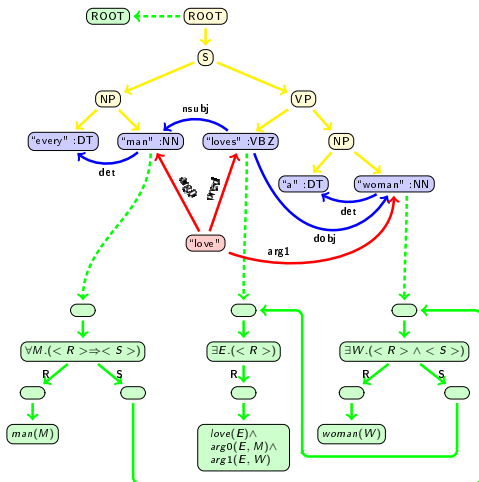
- Texte
- Analyse en constituants
- Dérivation des dépendances
- Étiquetage de rôles sémantiques
- Initialisation de la racine
- Initialisation des têtes
- Création des fragments
- Liaison des fragments

« Every man loves a woman »



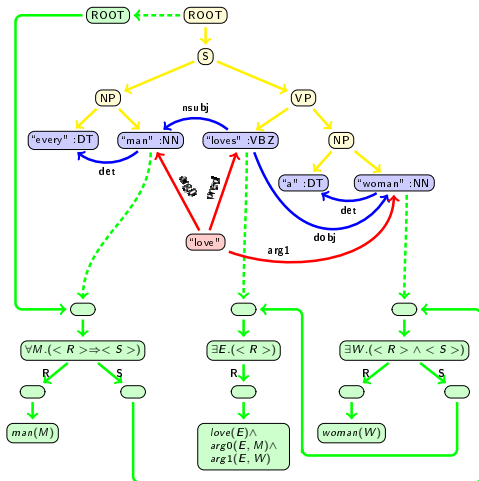
- Texte
- Analyse en constituants
- Dérivation des dépendances
- Étiquetage de rôles sémantiques
- Initialisation de la racine
- Initialisation des têtes
- Création des fragments
- Liaison des fragments
- Remplissage des fragments

« Every man loves a woman »



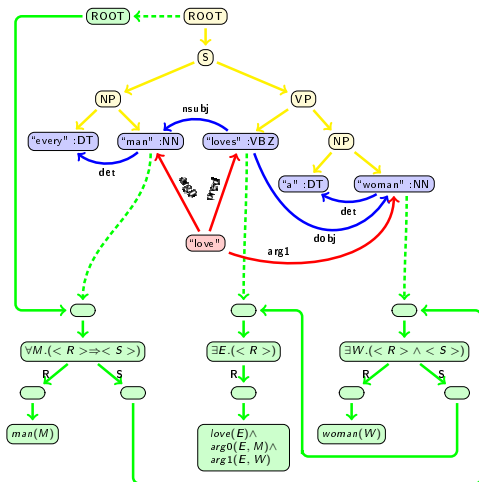
- Texte
- Analyse en constituants
- Dérivation des dépendances
- Étiquetage de rôles sémantiques
- Initialisation de la racine
- Initialisation des têtes
- Création des fragments
- Liaison des fragments
- Remplissage des fragments
- Rattachement à la racine

« Every man loves a woman »



- Texte
- Analyse en constituants
- Dérivation des dépendances
- Étiquetage de rôles sémantiques
- Initialisation de la racine
- Initialisation des têtes
- Création des fragments
- Liaison des fragments
- Remplissage des fragments
- Rattachement à la racine

« Every man loves a woman »



$$\forall M. (man(M) \Rightarrow \exists W. (woman(W) \wedge \exists E. (love(E) \wedge arg0(E, M) \wedge arg1(E, M))))$$

- 1 Introduction
 - Implication Textuelle
 - Approches existantes
 - Réécriture
- 2 Afazio : un système basé sur la réécriture
 - Architecture du système
 - Le calcul sémantique
 - Exemple
- 3 Évaluation sur des suites de tests contrôlées
- 4 Vers une analyse plus approfondie des systèmes
 - Motivation
 - ARTE : un schéma d'annotation fin pour le RTE
 - Fouille d'erreurs
- 5 Conclusion
 - Contributions de la thèse
 - Perspectives

- Pourquoi ?
 - Les problèmes du RTE mélangent **plusieurs** classes de phénomènes
 - Permet de mieux analyser l'**impact** des systèmes de reconnaissance d'implications textuelles sur les différents types de connaissance
- Génération **semi-automatique** de suites de tests **annotés**
 - Représentations sémantiques définies **manuellement**
 - Génération **automatique** de phrases à l'aide d'un réalisateur de surface (Genl [Eric Kow])
 - Test l'**implication** avec **outils de preuve**
- Problèmes **annotés** à l'aide des **informations de constructions** du Texte et l'Hypothèse fournies par **Genl**

- Suite de tests **syntaxiques** (1000 problèmes) :

Système	implications	non-implications	corpus entier
Afazio	68,0%	64,4%	66,2%
Nutcracker	89,2%	1,8%	46,8%
Nutcracker*	27,8%	90,8%	60,9%

- Suite de tests sur les **quantificateurs** (60 problèmes) :

Système	implications	non-implications	corpus entier
Afazio	100,0%	73,3%	86,7%
Nutcracker	23,3%	100,0%	61,7%
Nutcracker*	73,3%	100,0%	86,7%

- Erreurs** : mauvaises analyses de l'analyseur de Stanford

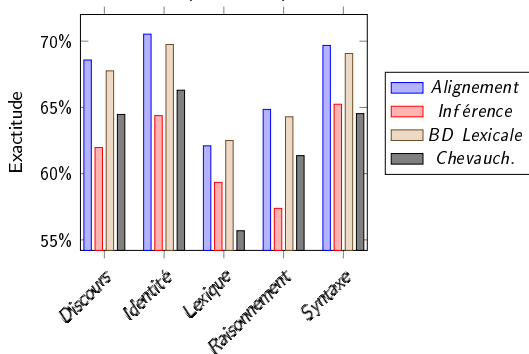
- 1 Introduction
 - Implication Textuelle
 - Approches existantes
 - Réécriture
- 2 Afazio : un système basé sur la réécriture
 - Architecture du système
 - Le calcul sémantique
 - Exemple
- 3 Évaluation sur des suites de tests contrôlées
- 4 Vers une analyse plus approfondie des systèmes
 - Motivation
 - ARTE : un schéma d'annotation fin pour le RTE
 - Fouille d'erreurs
- 5 Conclusion
 - Contributions de la thèse
 - Perspectives

- **Objectif :**
analyser l'impact des différents phénomènes linguistiques et extra-linguistiques sur les performances des différentes approches
- **Pré-requis :**
annotation fine des données (par phénomène)
- **Méthode :**
évaluation visant l'identification des sources d'erreurs plutôt qu'un taux d'exactitude

- Schéma d'annotation pour le RTE [Konstantina Garoufi]
- Annotations pour les implications (400 problèmes) :
 - Identité (365) : Identité
 - Lexique (121) : Acronyme, Hyperonyme, Synonyme, ...
 - Syntaxe (126) : Apposition, Alternation, ...
 - Discours (179) : Contre-factif (0), Implicatif (14), Négation (2), ...
 - Raisonnement (305) : Modifieur, Quantificateur, ...
- Classification des 41 systèmes selon les approches utilisées :
 - BD lexicale (32)
 - Chevauchement de mots (13)
 - Alignement (29)
 - Inférence (5)

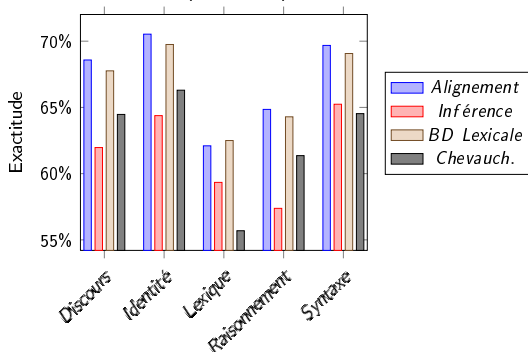
Évaluation basée sur l'exactitude

- Moyennes des exactitudes (ordonnée) des systèmes utilisant une approche (colonne) sur le sous-corpus de problèmes annotés avec une classe de phénomènes (abscisse)



Évaluation basée sur l'exactitude

- Moyennes des exactitudes (ordonnée) des systèmes utilisant une approche (colonne) sur le sous-corpus de problèmes annotés avec une classe de phénomènes (abscisse)



- Résultats **contre-intuitifs**
 - Le lexique est moins bien géré que le raisonnement
 - Les systèmes à base de chevauchement de mots ont de meilleurs résultats que ceux à base d'inférence sur les problèmes de raisonnement

- Le cas du **Lexique** et du **Raisonnement** sur les systèmes à base de chevauchement de mots

	Phénomènes	# Problèmes	Exactitude
1	Lexique sans Raisonnement	29	69%
2	Raisonnement sans Lexique	213	66%
3	Lexique et Raisonnement	92	52%
1 + 3	Lexique	121	56% (-14pts)
2 + 3	Raisonnement	305	61% (-5pts)

- Problème** : distribution **non uniforme** des phénomènes
- Solution** : utilisation d'un algorithme de **fouille d'erreurs**

- Algorithme de **fouille d'erreurs** pour les **analyseurs syntaxiques**
[Sagot De la Clergerie 2008]
- Permet de trouver les **n-grammes** qui posent problème à l'analyseur
- **Hypothèse** : un seul n-gramme est la cause de l'échec
- **Objectif** : Calculer un **taux de suspicion** pour chaque n-gramme

Un algorithme pour la fouille d'erreurs

Algorithme par **point fixe** applicable à un **corpus** de tests

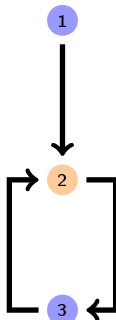
❶ Phase d'initialisation :

- Problème mal analysé :
taux de suspicion partagé **équitablement** par les **occurrences de classe de phénomènes** qui lui sont associées
- Problème bien analysé :
taux de suspicion **nul** pour les **occurrences de classe de phénomènes** qui lui sont associées

❷ Calcul du taux de suspicion **moyen** pour chaque classe de phénomènes

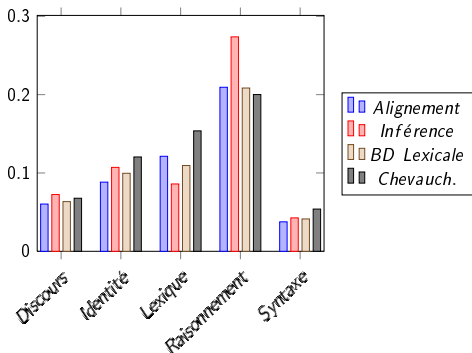
❸ Rééquilibrage des taux de suspicion :

- Problème mal analysé :
$$\frac{\text{taux de suspicion moyen d'une occurrence}}{\text{somme des taux de suspicion moyen de toutes les occurrences associées au problème}}$$
- Problème bien analysé :
rien ne change



Fouille d'erreurs appliquée sur les résultats de chaque système

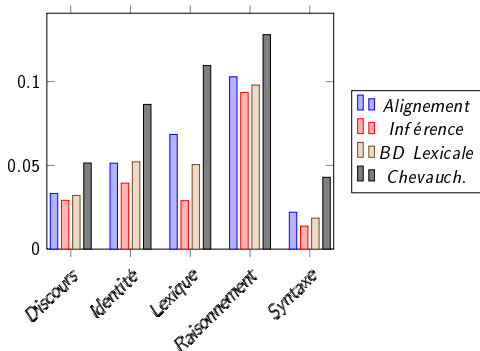
- Moyennes des taux de suspicion (ordonnée) des classes de phénomènes (abscisse) sur les systèmes utilisant une approche (colonne)



- Le raisonnement est moins bien géré que le lexique
- Les systèmes à base de chevauchement de mots ont toujours meilleurs résultats que ceux à base d'inférence sur le raisonnement

Fouille d'erreurs appliquée sur les résultats de tous les systèmes

- Taux de suspicion des couples classe de phénomènes/approche



- Le raisonnement est moins bien géré que le lexique
- Les systèmes à base de chevauchement de mots ont les moins bons résultats

- d'identifier les **points forts** et les **points faibles** des **systèmes**
- d'analyser des systèmes combinant **plusieurs approches** sur des corpus avec une distribution **non uniforme** des différentes classes de phénomènes

- 1 Introduction
 - Implication Textuelle
 - Approches existantes
 - Réécriture
- 2 Afazio : un système basé sur la réécriture
 - Architecture du système
 - Le calcul sémantique
 - Exemple
- 3 Évaluation sur des suites de tests contrôlés
- 4 Vers une analyse plus approfondie des systèmes
 - Motivation
 - ARTE : un schéma d'annotation fin pour le RTE
 - Fouille d'erreurs
- 5 Conclusion
 - Contributions de la thèse
 - Perspectives

- Une analyse critique de la campagne RTE
- Un système de reconnaissance d'implications textuelles basé sur la réécriture
 - Normalisation des structures verbales [TALN'08, IWCS'09]
 - Normalisation des structures nominales [LTC'09]
 - Calcul des formules logiques à partir des structures obtenues [LNAI'11]

- Une méthode pour la génération de suite de tests d'implications textuelles
 - Génération semi-automatique de suites de tests annotés [LREC'10,COLING'10]
 - Algorithme de sélection d'un sous-ensemble d'éléments satisfaisant un ensemble de contraintes de distribution
- Une méthode alternative pour l'évaluation des systèmes de reconnaissance d'implications textuelles [LREC'10,COLING'10]
 - Adaptation d'une méthode de fouille d'erreurs développée par la communauté de l'analyse syntaxique
 - Réévaluation du RTE2 annoté avec le schéma d'annotation ARTE
 - Évaluation comparative d'Afazio et de Nutcracker

- RTE :
 - Analyse des cas où l'implication logique et l'implication textuelle diffèrent
- Afazio :
 - Désambiguïsation des verbes grâce aux types sémantiques de leurs arguments (Stuttgart)
 - Extension du calcul sémantique (couverture, sous-spécification)
 - Applicabilité des logiques de Markov et non-monotones à la détection d'implications textuelles
- Génération de suite de tests :
 - Mise en œuvre d'une méthode d'annotation automatique qui prenne en compte l'alignement du texte et de l'hypothèse
 - Ajout de nouveaux phénomènes