Parsing sémantique à la FrameNet

William Léchelle

RALI — UdeM

19 Mars 2015

1 Frame Semantics

2 FrameNet

3 SEMAFOR

1 Frame Semantics

2 FrameNet

3 SEMAFOR

Frames

- Structures sémantiques prédicat / arguments
  - "Schémas de compréhension"
  - "Schémas d'action"
  - "Cadres évènementiels"

Frames

- Structures sémantiques prédicat / arguments
  - "Schémas de compréhension"
  - "Schémas d'action"
  - "Cadres évènementiels"
- Exemple : `Attack`

  An Assailant physically attacks a Victim (which is usually but not always sentient), causing or intending to cause the Victim physical damage. A Weapon used by the Assailant may also be mentioned, in addition to the usual Place, Time, Purpose, Reason, etc.

- Revenge

  An Avenger performs a Punishment on a Offender as a consequence of an earlier action by the Offender, the Injury.

- `Revenge`

  An Avenger performs a Punishment on a Offender as a consequence of an earlier action by the Offender, the Injury.

- `Commerce_sell`

  The words vary individually in the patterns of frame element realization they allow. Typical patterns : Seller sells Goods to Buyer for Money.

Prédicats

- Les prédicats sont des unités lexicales – mot + POS + sens
  - aka "cibles" (*targets*)

Prédicats

- Les prédicats sont des unités lexicales – mot + POS + sens
  - aka "cibles" (*targets*)
- `Attack` : airstrike.n, ambush.n, bomb.v, charge.v, infiltration.n, raid.n, strike.v, . . .

Prédicats

- Les prédicats sont des unités lexicales – mot + POS + sens
  - aka "cibles" (*targets*)
- `Attack` : airstrike.n, ambush.n, bomb.v, charge.v, infiltration.n, raid.n, strike.v, ...
- `Revenge` : avenge.v, `get_back_(at).v`, `get_even.v`, retaliation.n, revenge.v, revengeful.a, vengeance.n, vengeful.a, ...

Rôles = *Frame elements*

- Les arguments (*frame elements*) remplissent différents rôles dans le schéma.

Rôles = *Frame elements*

- Les arguments (*frame elements*) remplissent différents rôles dans le schéma.
- Ex : `Attack`
  - `Assaillant`, `Victime`

Rôles = *Frame elements*

- Les arguments (*frame elements*) remplissent différents rôles dans le schéma.
- Ex : `Attack`
  - `Assaillant`, `Victime`
  - `Circonstances`, `Durée`, `Fréquence`

Rôles = *Frame elements*

- Les arguments (*frame elements*) remplissent différents rôles dans le schéma.
- Ex : `Attack`
  - `Assaillant`, `Victime`
  - `Circonstances`, `Durée`, `Fréquence`
  - `Manière`, `Chemin`, `Raison`

Rôles = *Frame elements*

- Les arguments (*frame elements*) remplissent différents rôles dans le schéma.
- Ex : `Attack`
  - `Assaillant`, `Victime`
  - `Circonstances`, `Durée`, `Fréquence`
  - `Manière`, `Chemin`, `Raison`
  - `Arme`

Rôles = *Frame elements*

- Les arguments (*frame elements*) remplissent différents rôles dans le schéma.
- Ex : `Attack`
  - `Assaillant`, `Victime`
  - `Circonstances`, `Durée`, `Fréquence`
  - `Manière`, `Chemin`, `Raison`
  - `Arme`
  - ...

Frame elements

- Actants / *Core frame elements* (2-5)
- Circonstants / *Non-core frame elements* (5-20)

Frame elements

- Actants / *Core frame elements* (2-5)
- Circonstants / *Non-core frame elements* (5-20)
  - *Peripheral* (compléments circonstanciels)

Frame elements

- Actants / *Core frame elements* (2-5)
- Circonstants / *Non-core frame elements* (5-20)
  - *Peripheral* (compléments circonstanciels)
  - *Extra-thematic* (par rapport à autre chose, e.g. "dans le cadre de")

Relations entre élements

- Exclus, Requiert, . . .

Relations entre élements

- Exclus, Requiert, ...
- Exemple : `Motion_noise` (*The door **banged** open.*)

  This frame pertains to noise verbs used to characterize motion. `Motion_noise` verbs take largely the same Source, Path and Goal expressions as other types of `Motion` verbs.

Relations entre élements

- Exclus, Requiert, ...
- Exemple : `Motion_noise` (*The door **banged** open.*)

  This frame pertains to noise verbs used to characterize motion. `Motion_noise` verbs take largely the same Source, Path and Goal expressions as other types of `Motion` verbs.

  - `Area` *This FE identifies the general Area in which motion takes place, used particularly if the motion is understood as following a complex or non-linear path.*

Relations entre élements

- Exclus, Requiert, ...
- Exemple : `Motion_noise` (*The door **banged** open.*)

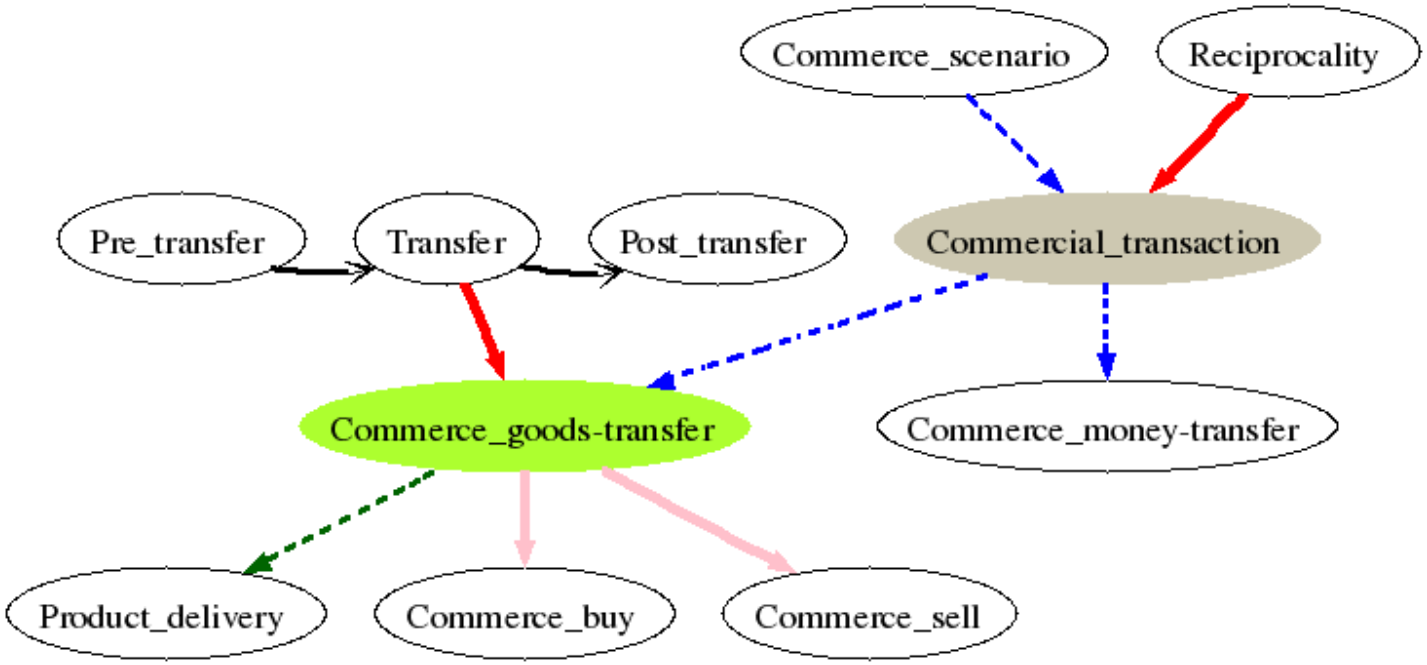
  This frame pertains to noise verbs used to characterize motion. `Motion_noise` verbs take largely the same Source, Path and Goal expressions as other types of `Motion` verbs.

  - `Area` *This FE identifies the general Area in which motion takes place, used particularly if the motion is understood as following a complex or non-linear path.*
  - `Path` *The Path is any description of a trajectory of motion which is neither a Source nor a Goal.*

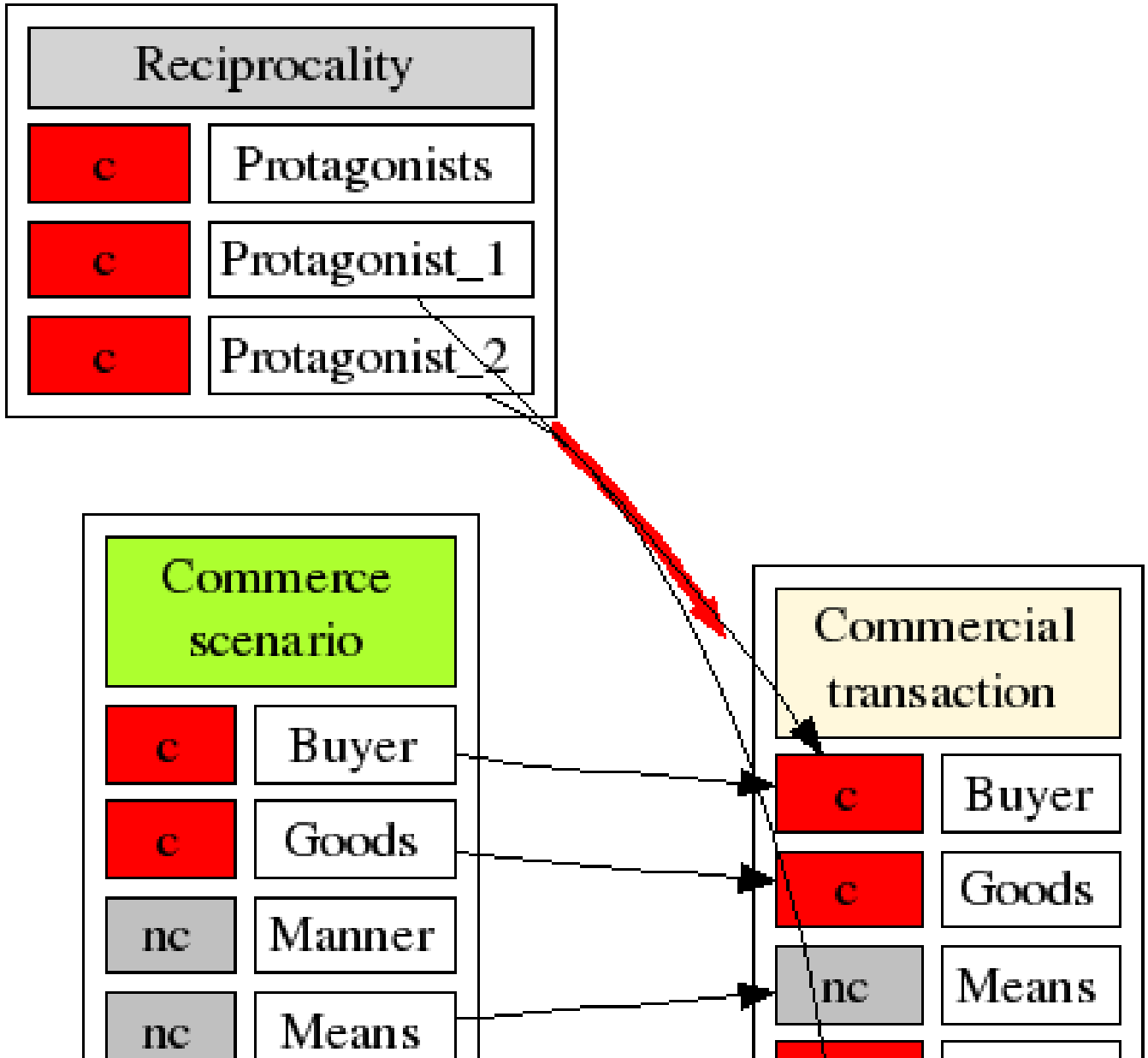
    Excludes: Area

Hiérarchie entre frames

- Héritage, Sous-cas, Point du vue sur, Utilise, Précède...



Correspondance des éléments



Ce que ça regroupe

- Critère majeur : paraphrase

Ce que ça regroupe

- Critère majeur : paraphrase
- Passif, constructions de temps

Ce que ça regroupe

- Critère majeur : paraphrase
- Passif, constructions de temps
- Degrés d'un même phénomène

Ce que ça regroupe

- Critère majeur : paraphrase
- Passif, constructions de temps
- Degrés d'un même phénomène
- Mêmes dépendances syntaxiques

Ce que ça distingue

- Différences de point de vue (acheteur / vendeur)
  - Typiquement, avec une *frame* de plus haut niveau qui les regroupe

Ce que ça distingue

- Différences de point de vue (acheteur / vendeur)
  - Typiquement, avec une *frame* de plus haut niveau qui les regroupe
- Stades de l'action

Ce que ça distingue

- Différences de point de vue (acheteur / vendeur)
  - Typiquement, avec une *frame* de plus haut niveau qui les regroupe
- Stades de l'action
- Nombre / type d'arguments différents

1 Frame Semantics

2 FrameNet

3 SEMAFOR

Concrètement

Base de données lexicale de *frames*

Concrètement

Base de données lexicale de *frames*

- Version 1.5 sortie en 2010
  - 900 *frames*, 10k roles
  - 9k unités lexicales

Commercial_transaction

Definition:

These are words that describe basic commercial transactions involving a Buyer and a Seller who exchange Money and Goods. The individual words vary in the fra
For example, the typical patterns for the verbs buy and sell are: BUYER buys GOODS from the SELLER for MONEY. SELLER sells GOODS to the BUYER f

FEs:

Core:

Buyer [Byr]	The Buyer wants the Goods and offers Money to a Seller in exchange for them.
Goods [Gds]	The FE Goods is anything (including labor or time, for example) which is exchanged for Money in a transaction.
Money [Mny]	Money is the thing given in exchange for Goods in a transaction.
Seller [Slr]	The Seller has possession of the Goods and exchanges them for Money from a Buyer.

Non-Core:

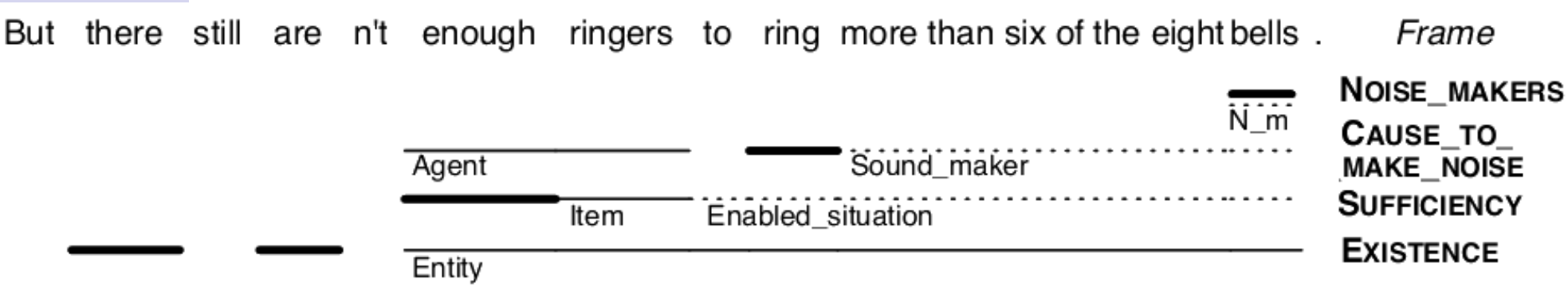
Means [Mns] **Semantic Type:** State_of_affairs	The means by which a commercial transaction occurs.
Rate [Rate]	Price or payment per unit of Goods.
Unit [Unit]	The Unit of measure of the Goods according to which the exchange value of the Goods (or services) is set. Generally, it occur

Frame-frame Relations:

Inherits from: Reciprocality
Is Inherited by:
Uses:
Is Used by:
Subframe of: Commerce_scenario
Has Subframe(s): Commerce_goods-transfer, Commerce_money-transfer
[...]

Lexical Units: *transaction.n*

Annotations



Données

- 150k exemples pris dans le British National Corpus
- 78 documents exhaustivement annotés
  - 5k phrases
  - 24k cibles (50% noms, 30% verbes)

1 Frame Semantics

2 FrameNet

3 SEMAFOR

En gros

- Annoter sémantiquement selon ces structures.
  - Trouver les schémas d'action
  - Trouver comment sont remplis les rôles

En gros

- Annoter sémantiquement selon ces structures.
  - Trouver les schémas d'action
  - Trouver comment sont remplis les rôles
- Utilise un parseur syntaxique

Pipeline

- Trouver les cibles

Pipeline

- Trouver les cibles
- Déterminer quelles frames correspondent

Pipeline

- Trouver les cibles
- Déterminer quelles frames correspondent
- Identifier les arguments

Trouver les cibles

- Difficulté : beaucoup de cibles potentielles n'en sont pas (verbes-supports, compléments...)

Trouver les cibles

- Difficulté : beaucoup de cibles potentielles n'en sont pas (verbes-supports, compléments...)
- Cibles vues à l'entrainement, à morphologie près.
  - dans un arbre syntaxique élagué

Trouver les cibles

- Difficulté : beaucoup de cibles potentielles n'en sont pas (verbes-supports, compléments...)
- Cibles vues à l'entrainement, à morphologie près.
  - dans un arbre syntaxique élagué
- Précision : 90%, Rappel : 70%
  - Cibles discontiguës laissées de côté (rappel : 95%)

Désambiguiser les frames

- 50% des cibles du jeu de test sont ambiguës.

Désambiguiser les frames

- 50% des cibles du jeu de test sont ambiguës.
- Modèle à maximum d'entropie, selon le lemme de la cible.
  - 600k *features* binaires

Désambiguiser les frames

- 50% des cibles du jeu de test sont ambiguës.
- Modèle à maximum d'entropie, selon le lemme de la cible.
  - 600k *features* binaires
- On peut trouver une *frame* reliée (réussite partielle)

Désambiguiser les frames

- 50% des cibles du jeu de test sont ambiguës.
- Modèle à maximum d'entropie, selon le lemme de la cible.
  - 600k *features* binaires
- On peut trouver une *frame* reliée (réussite partielle)
- Précision : 70%, Rappel : 60% (avec les cibles trouvées)

Identifier les arguments

- Typiquement :
  - Identifier les arguments
  - Assigner les rôles (ou $\emptyset$)
- SEMAFOR :
  - Pour chaque rôle, quel segment correspond le mieux ?

- Pour chaque rôle, quel segment correspond le mieux ?
  - Un mot ou un nœud (r : 81%)
    - ou $\emptyset$
  - Un seul nœud pour chaque rôle (r : 93%)
  - (après élaguage)

- Pour chaque rôle, quel segment correspond le mieux ?
  - Un mot ou un nœud (r : 81%)
    - ou $\emptyset$
  - Un seul nœud pour chaque rôle (r : 93%)
  - (après élaguage)
- Modèle à maximum d'entropie
  - 1M *features* binaires pour chaque candidat

Features with both null and non-null variants: These features come in two flavors: if the argument is null, then one version fires; if it is overt (non-null), then another version fires.

- ● some word in t has lemma λ
- ◐ some word in t has lemma λ, and the sentence uses PASSIVE voice
- ◐ the head of t has subcategorization sequence $\boldsymbol{\tau} = \langle \tau_1, \tau_2, \ldots \rangle$
- ● the head of t has c syntactic dependents
- ● some word in t has POS π
- ◐ some word in t has lemma λ, and the sentence uses ACTIVE voice
- ◐ some syntactic dependent of the head of t has dependency type τ
- ● bias feature (always fires)

Span content features: apply to overt argument candidates.

- ○ POS tag π occurs for some word in s
- ○ the head word of s has POS π
- ○ the first word of s has POS π, provided $|s| > 0$
- ○ the last word of s has POS π, provided $|s| > 0$
- ● the first word of s: w_{s_1}, and its POS tag π_{s_1}, provided that π_{s_1} is a closed-class POS[16]
- ● w_{s_2} and its closed-class POS tag π_{s_2}, provided that $|s| \geq 2$
- ● the last word of s: $w_{s_{|s|}}$, and its closed-class POS tag $\pi_{s_{|s|}}$, provided that $|s| \geq 3$
- ◐ lemma λ is realized in some word in s
- ◐ lemma λ is realized in some word in s, the voice denoted in the span (ACTIVE or PASSIVE)
- ● $|s|$, the number of words in the candidate argument[15]
- ○ the head word of s has syntactic dependency type τ
- ● the syntactic dependency type τ_{s_1} of the first word with respect to its head
- ● τ_{s_2}, provided that $|s| \geq 2$
- ● $\tau_{s_{|s|}}$, provided that $|s| \geq 3$
- ○ the first word of s has lemma λ, provided $|s| > 0$
- ○ the head word of s has lemma λ
- ○ the last word of s has lemma λ, provided $|s| > 0$
- ◐ lemma λ is realized in some word in s, the voice denoted in the span, s's position with respect to t (BEFORE, AFTER, or OVERLAPPING)

Syntactic features: apply to overt argument candidates.

- ○ dependency path: sequence of labeled, directed edges from the head word of s to the head word of t
- ○ length of the dependency path[15]

Span context POS features: for overt candidates, up to 6 of these features will be active.

- ○ a word with POS π occurs up to 3 words before the first word of s
- ○ a word with POS π occurs up to 3 words after the last word of s

Ordering features: apply to overt argument candidates.

- ● the position of s with respect to to the span of t:
- ○ target-argument crossing:

Figure: *Features*

Résultats

- Après le pipeline : P=58%, R=39%, F_1=46%

Résultats

- Après le pipeline : P=58%, R=39%, F_1=46%
- Connaissant cibles et *frames* : P=79%, R=61%, F_1=68%

Résultats

- Après le pipeline : P=58%, R=39%, F_1=46%
- Connaissant cibles et *frames* : P=79%, R=61%, F_1=68%
- Connaissant cibles, *frames*, et *spans* des arguments : P=88%, R=75%, F_1=81%

Résultats

- Après le pipeline : P=58%, R=39%, F_1=46%
- Connaissant cibles et *frames* : P=79%, R=61%, F_1=68%
- Connaissant cibles, *frames*, et *spans* des arguments : P=88%, R=75%, F_1=81%
  - 19% des arguments ne sont pas des nœuds de l'arbre (ou erreurs de parsing).

Résultats

- Après le pipeline : P=58%, R=39%, F_1=46%
- Connaissant cibles et *frames* : P=79%, R=61%, F_1=68%
- Connaissant cibles, *frames*, et *spans* des arguments : P=88%, R=75%, F_1=81%
  - 19% des arguments ne sont pas des nœuds de l'arbre (ou erreurs de parsing).
  - 10% des prédictions sont décalées de 1 ou 2 mots.

Différents parseurs

- `Shalmaneser` (2005)
  - Une chaine d'outils en SALSA/TIGER XML, Ruby & MySQL
  - Peut utiliser plusieurs parseurs
  - Désambiguise avec des sacs de mots et un classifieur bayésien

Différents parseurs

- `Shalmaneser` (2005)
  - Une chaine d'outils en SALSA/TIGER XML, Ruby & MySQL
  - Peut utiliser plusieurs parseurs
  - Désambiguise avec des sacs de mots et un classifieur bayésien
- `LTH`, baseline de Semafor (2007)
  - Moults SVM (un par *frame*).
  - Néglige les cibles de plusieurs mots (e.g. *ballistic missile*.N)

Différents parseurs

- `Shalmaneser` (2005)
  - Une chaine d'outils en SALSA/TIGER XML, Ruby & MySQL
  - Peut utiliser plusieurs parseurs
  - Désambiguise avec des sacs de mots et un classifieur bayésien
- `LTH`, baseline de Semafor (2007)
  - Moults SVM (un par *frame*).
  - Néglige les cibles de plusieurs mots (e.g. *ballistic missile*.N)
- Semafor (2010-2012)

Annotation sémantique

On a pas parlé de

- PropBank (plus syntaxique)

Annotation sémantique

On a pas parlé de

- PropBank (plus syntaxique)
- Entity Linking (plus superficiel)

Annotation sémantique

On a pas parlé de

- PropBank (plus syntaxique)
- Entity Linking (plus superficiel)
- Sémantique formelle (plus logique)

Bibliographie

- *Frame semantics and the nature of language*, Fillmore Charles, 1976.
- *The Berkeley FrameNet Project*, Baker, Fillmore, and Lowe, 1998
- *FrameNet II: Extended Theory and Practice*, Ruppenhofer et al. 2005