

Niveaux d'analyse linguistique en Traitement Automatique des Langues

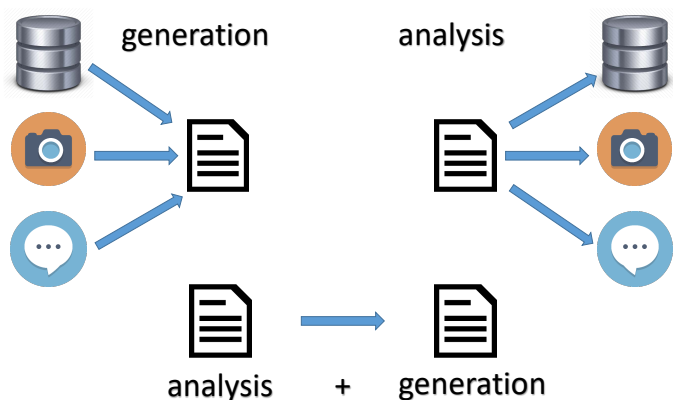
Frederic Bechet

Aix Marseille Université - Laboratoire d'Informatique et Systèmes - LIS-CNRS UMR 7020



Master Info

Traitement Automatique des Langues



Traitement Automatique des Langues

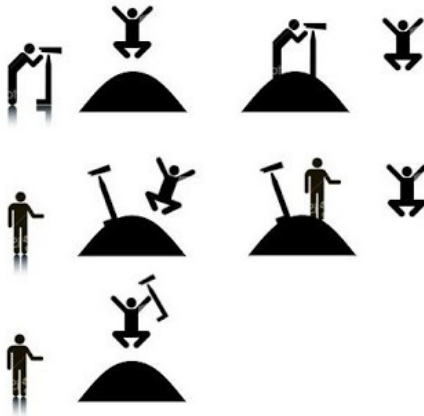
- Les langages naturels sont implicites et ambigus
 - ▶ *Remove the stones from the cherries and put them in the pie*
 - ▶ *I saw the man on the hill with a telescope*
- **Compréhension du langage ?**
 - ▶ rendre le langage explicite
 - ▶ enlever les ambiguïtés
- OK, mais comment ?
 - ▶ *Compréhension* à travers une tâche
 - ▶ *Compréhension* à travers un modèle formel

Compréhension

- à travers une tâche
 - ▶ *traduction*
 - ▶ *résumé automatique*
 - ▶ *systèmes de dialogue*
- à travers un modèle formel
 - ▶ basé sur une **theorie** du langage
 - ▶ but : découvrir les structures sous-jacentes de la langue
 - ▶ produire des analyses non-ambigues

Analyse linguistique sur plusieurs niveaux

I saw the man on the hill with a telescope



Beaucoup de théories linguistiques !!

- Accord sur les principaux niveaux de traitement auqnd on analyse une *phrase*
 - ❶ d'abord, qu'est-ce qu'une phrase ?
 - ❷ niveau morpho-lexical → quels sont les *mots*, les unités de base d'une *phrase* ?
 - ❸ niveau syntaxique → quelles sont les structures qui lient les mots ensembles ?
 - ❹ niveau sémantique → quel est le *sens littéral* d'une séquence de mots ?
 - ❺ niveau pragmatique → quel est le sens en contexte d'une séquence de mots ?
- Cependant de très nombreuses théories ont été proposées pour chaque niveaux !!
- et tous les niveaux sont inter-dépendants
 - ▶ *I saw [the man on the hill] with [a telescope]*
 - ▶ *I saw [the man] on [the hill with a telescope]*

Niveau Morphologique

- Part-Of-Speech tagging
 - ▶ étiquetage morpho-syntaxique



Niveau syntaxique

- A quoi sert la syntaxe ?



6/29/16, 1:48 PM

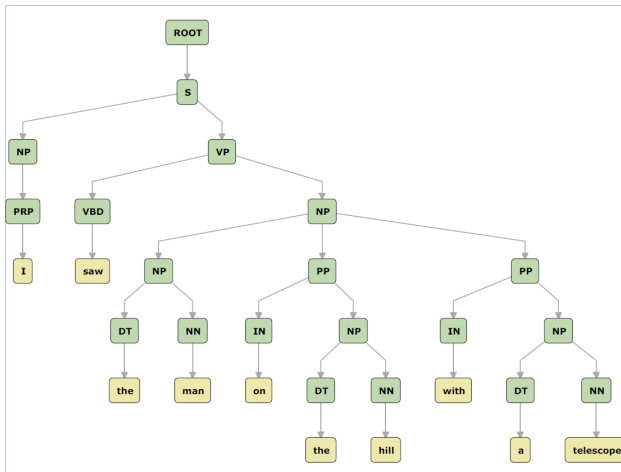
Niveau syntaxique

- Beaucoup de théories différentes !!
- Structure plate : *chunking*
 - ▶ *[I] [saw] [the man] [on the hill] [with a telescope]*
- Structure de graphe
 - ▶ Constituency parsing (analyse en constitutants)
 - ▶ Dependency parsing (analyse en dépendance)

Constituency parsing

- Principe

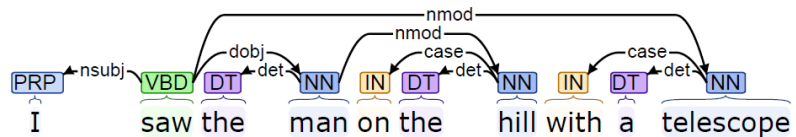
- ▶ La structure d'une phrase est faite de mots enchassés en groupes appelés constituants
- ▶ Context-Free-Grammar / Noam Chomsky



Dependency parsing

- Principe

- ▶ La structure syntaxique d'une phrase consiste à relier les mots avec des relations binaires, typées, appelées *dépendances*
- ▶ Dependency Grammar / Lucien Tesnière



Niveau Semantique

Encore plus de théories !!

- Globale
 - ▶ thème
- Locale
 - ▶ Named-entity recognition (reconnaissance en *entités nommées*)
 - ▶ Etiquetage concept/valeur
- Structures sémantiques
 - ▶ coreference resolution (résolution des coréférences)
 - ▶ analyse sémantique de surface en cadre sémantique (semantic frames)
 - ▶ predicat/valeur
 - ▶ graphes sémantiques
 - ▶ sémantique formelle → forme logique
 - ▶ ...

Niveau Semantique

- Semantic Role Labelling

	I	saw	the	man	on	the	hill	with	a	telescope	.
SRL		V: see.01	thing viewed [A1]	location [AM-LOC]							
Nom											
Prep			Governor					Instrument (with)		Object	
Prep			Governor	PhysicalSupport (on)		Object					

Niveau Semantique

- Berkeley Framenet

Taking

Definition:

An **Agent** removes a **Theme** from a **Source** so that the it is in the **Agent**'s possession.
Milton **TOOK** the can of beer out of the refrigerator.

FEs:

Core:

Agent []

Semantic Type: Senticnt

The person who takes possession of the **Theme**.

Milton **TOOK** the can of beer out of the refrigerator.

Source []

Semantic Type: Source

The location of the **Theme** prior to the taking.

Milton **TOOK** the can of beer out of the refrigerator.

Theme []

Semantic Type: Physical_object

The **Agent** takes possession of the **Theme**.

Milton **TOOK** the can of beer out of the refrigerator.

Niveau Semantique

- Abstract Meaning Representation

AMR – Abstract Meaning Representation

- Nodes are **variables** labelled by **concepts**

- Entities, events, states, properties
- **d** / **dog**: **d** is an instance of **dog**

- Edges are semantic **relations**

- E.g. “The dog is eating bones.”

(**e** / **eat-01**
:ARG0 (**d** / **dog**)
:ARG1 (**b** / **bone**))



eat.01: consume (VN-class: **eat-39.1**, FN-frame: **Ingestion**)

ARG0-PAG: consumer, eater (VN-role: **agent**)

ARG1-PPT: meal (VN-role: **patient**)

Analyse linguistique

Beaucoup de tâches, beaucoup de modèles, mais quels processus sous-jacents ?

- 3 opérations de base
 - ① segmentation
 - ② étiquetage
 - ③ mise en relation (linking)

- Architecture naïve

- ▶ Etant donné une entrée X

- 1 énumération de toutes les solutions possibles $Y = Y_1, Y_2, \dots, Y_N$
- 2 évaluer et donner un score à chaque solution Y_i as $p(Y_i)$
- 3 choix de la *meilleure* solution $\hat{Y} = \operatorname{argmax}_{Y_i \in Y} p(Y_i)$

Segmentation

$X =$	a	b	c	d
	a	b	c	d
	a	b	c	d
$\mathcal{Y} =$	a	b	c	d
	a	b	c	d
	...			
	a	b	c	d

- 2^{n-1} segmentations possible

Labelling

$X =$	a	b	c	d
	1	1	1	1
	1	1	1	2
$\mathcal{Y} =$	1	1	2	1
	...			
	k	k	k	k

- k^n séquences de labels possibles

Linking

$X =$	a	b	c	d
<hr/>				
	a	a	a	a
	a	a	a	b
$\mathcal{Y} =$	a	a	b	a
	\dots			
	d	d	d	d

- n^n relations possibles dans le graphe

Quels algorithmes pour toutes ces opérations ?

- Problème 1 → nombre énorme de solutions possibles pour chaque opération
 - ▶ impossible d'énumérer toutes les solutions
 - ▶ propositions
 - ★ couper chaque solution Y en *parties* : $F(Y) = y_1, y_2, \dots$, partagées par des solutions multiples
 - ★ évaluer une solution revient à trouver une fonction combinant les évaluations de chacune des parties :
$$p(Y) = \sum_{y \in F(Y)} p(y)$$
 - ★ ajouter des contraintes pour réduire le nombre de solutions
 - ▶ implémentation
 - ★ heuristiques, règles expertes
 - ★ (très large) choix de méthodes d'apprentissage automatique

Quels algorithmes pour toutes ces opérations ?

- Problème 2 → interdependence entre les operations
 - ▶ impossible de garder tous les espaces de recherche de chaque opération
 - ▶ propositions
 - ★ ajouter plus de contraintes !!
 - ★ **pipeline architecture**

NLP pipeline architecture

- Principes

- ▶ chaque operation est un module M
- ▶ l'entrée du module M_i est la sortie du module M_{i-1}
- ▶ chaque module M_i produit uniquement la meilleure solution \hat{Y}^i d'après la fonction d'évaluation



NLP pipeline architecture

- Avantages

- ▶ simplicité → l'analyse linguistique est la concaténation de modules indépendants
- ▶ flexibilité → chaque module peut avoir une architecture différente, du moment qu'il respecte les entrées/sorties attendues
- ▶ efficacité → l'espace de recherche est réduit de manière radical entre chaque module (1-best)

- Inconvénients

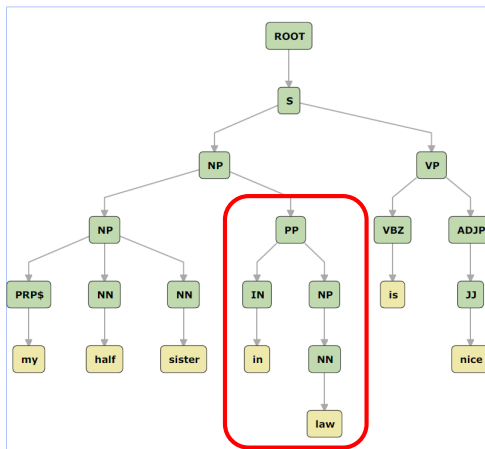
- ▶ sous-optimal : on brise l'hypothèse d'interdependance entre les niveaux d'analyse
- ▶ effet *boule de neige* possible : une erreur au module M_i peut provoquer plusieurs erreurs au niveau M_{i+1}

Exemple de pipeline linguistique

- 1 Sentence segmentation
- 2 Word tokenization
- 3 Part-Of-Speech tagging
- 4 Chunking
- 5 Syntactic parsing
- 6 Named Entity Recognition
- 7 Co-reference resolution
- 8 Semantic parsing

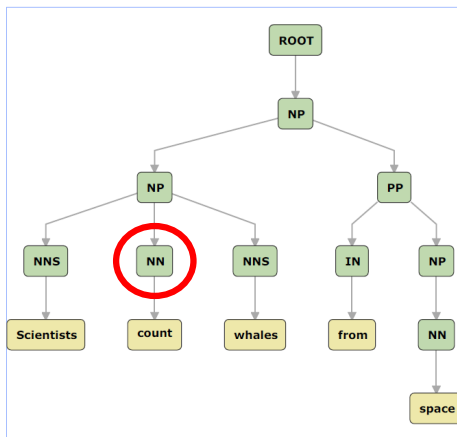
Est-ce qu'on peut faire mieux que le pipeline ?

- Prendre des décisions trop tôt peut poser problème ..



Est-ce qu'on peut faire mieux que le pipeline ?

- Effet “boule de neige” des erreurs

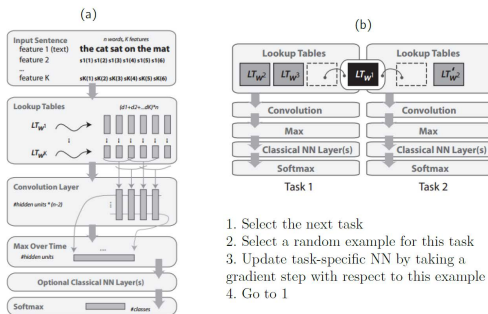


Comment briser ce pipeline ?

- Garder plusieurs hypothèses entre chaque module : espace de recherche
 - ▶ listes de n-meilleures (n-best) solutions
 - ▶ treillis contenant des solutions multiples
 - ▶ avantages : simplicité, chaque module reste indépendant
 - ▶ inconvénients : les n-meilleures solutions sont souvent très similaires, et garder un treillis pose des problèmes combinatoires

Comment briser ce pipeline ?

- Utiliser une approche “multi-tâche”
 - ▶ Unification des représentations du texte quelle que soit la tâche
 - ▶ sorties multiples ou on non interdépendance entre les niveaux d'analyse
 - ▶ implémentation avec des *réseaux de neurones profonds*



- *Natural Language Processing (Almost) from Scratch* (Collobert et al.)

Multi-task with Deep Neural Network

- Using a multitask approach
 - ▶ need to unify all processes
 - ▶ with a unified representation for all tasks

Can be implemented thanks to *deep neural networks*!!!!

Unified representation with DNN

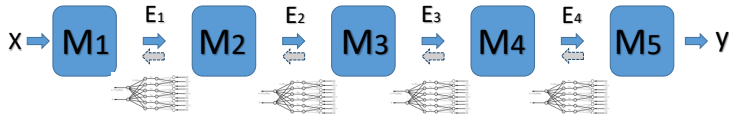
- Continuous word meaning representation
 - ▶ Nothing new!! **Distributional Semantic**
 - ★ *You shall know a word by the company it keeps* (J.R. Firth, 1957)
- From *Frequency-based models* to *Prediction-based models*
 - ▶ *sucess story of word embeddings*

Unified representation with DNN

- *Prediction-based* models (or probabilistic models)
 - ▶ Model training on a (very) large text corpus for :
 - ★ Predicting a word according to its context
 - ★ Predicting a context from a word
- (Deep) Neural Network prediction models
 - ▶ The last layer in the NN before the prediction output is used to represent each word
 - ▶ This layer is the word vector on which similarities will be estimated
 - ★ Mikolov 2013 : Word2Vec
 - ★ *Don't count, predict!* [Baroni et al., 2014]
- Dynamic contextual representation (Transformer models)
 - ▶ *BERT* the *swiss army knife* of NLP

Breaking the pipeline with Deep Neural Networks

- Deep Neural Networks + Representation Learning (*embeddings*)
 - ▶ open a lot of possibilities for multi-task learnings
- A lot of architecture can be defined !!
 - ▶ unified network with multi-task loss
 - ▶ different networks with common representation updating one task at a time
 - ▶ learning representation for a task + fine-tuning on another one → *embeddings pipeline*



Le règne de BERT

- **BERT** le *couteau-suisse* des modèles de langue ?
 - ▶ modèle de représentation *contextuelle* des *mots*
 - ▶ appris sur plusieurs tâches en mode auto-supervisé
 - ▶ sur d'énormes quantités de données
 - ▶ sur plusieurs langues *en même temps*
- General Language Understanding Evaluation (GLUE) (et SuperGLUE)
 - ▶ 9 tâches de “*compréhension*” du langage
 - ★ question/réponse
 - ★ inférence
 - ★ analyse de sentiment
 - ★ ...
- Machine 1 - Humain 0 ?
 - ▶ <https://gluebenchmark.com/leaderboard>