

Introduction à l'Intelligence Artificielle (L2 Portail Sciences et Technologies)

Andrea G. B. Tettamanzi
Laboratoire I3S – Équipe SPARKS
`andrea.tettamanzi@univ-cotedazur.fr`



univ-cotedazur.fr

Séance 4

Traitement de la langue, agents conversationnels

Plan pour cette séance

- Introduction
- Langages formels et langages réels
- Modèles de langage
- Recherche d'information
- Analyse syntaxique
- Sémantique
- Agents conversationnels

Introduction

- La faculté du langage est un des traits caractéristiques de l'intelligence

Langages formels : intuition de fond

- L'intuition sur laquelle se fonde la théorie des langages est que l'on peut décrire un langage par l'ensemble (éventuellement infini) des énoncés (appelés « mots » ou « phrases ») qui sont acceptables (bien formés) dans ce langage.
- Cette définition est plus simple et fondamentale que celle familière basée sur des règles, des spécifications ou, comme on les appelle, des grammaires.
- Par exemple, le français peut être défini comme l'ensemble de toutes les chaînes de caractères qui représentent des phrases syntaxiquement valides

Langage formel

Σ Alphabet : ensemble fini de symboles (lettres, lexèmes, ...)

Σ^* Ensemble de toutes les chaînes d'éléments de Σ

$L \subseteq \Sigma^*$

het is altijd de burger
asxj kxopq kdo
capté cru j'aurais t'as pas que
suih uihlkj shj qowdinmx ji jpo
yüz maskesi üretim süreci
en catchana baby tu dead ça
scientists hunt for the truth
t'as cru que j'aurais pas capté
une des priorités est de rétablir les communications
je vous en prie
oui
aaa
ecco le formazioni ufficiali

Spécification d'un langage

Un langage formel peut être spécifié par différents moyens. Ce qui est recherché, c'est une méthode finie et explicite (donc un algorithme) qui permet de produire ou d'analyser un langage en général infini. Parmi ces méthodes :

- Les grammaires formelles : les mots sont produits par des règles, en nombre fini, qui s'appliquent dans des conditions précises
- Les automates : ce sont des machines mathématiques qui reconnaissent une certaine catégorie de mots ; parmi eux, il y a les machines de Turing ou les automates finis
- L'ensemble des instances d'un problème de décision dont la réponse est OUI
- divers systèmes logiques de description à l'aide de formules logiques
- les expressions rationnelles (ou régulières, RegExp)

Expressions Rationnelles

Les expressions régulières sur l'alphabet Σ sont des expressions obtenues à partir des constantes

- \emptyset (le langage vide)
- ε (le mot vide)
- les symboles $\sigma \in \Sigma$

par les opérations suivantes, dites rationnelles :

- l'opération $+$ (parfois notée $|$) d'union de langages ;
- l'opération \cdot de concaténation ;
- l'opération $*$ de fermeture de Kleene : X^* est le plus petit langage qui contient ε , le langage X et qui est clos pour l'opération de concaténation ou, en d'autres termes, l'ensemble de toutes les concaténations de tous les mots de X .

Classe des langages réguliers ; reconnus par des automates finis

Expressions Régulières

- Caractère d'échappement : \
- Classes de caractères : [aeiouy], [A-Z], [^0-9]
- \w : caractère d'un mot ; \W : le contraire
- \d : chiffre ; \D : tout sauf un chiffre
- \s : caractère blanc ; \S : non-blanc
- . : caractère jocker
- Alternative : |
- Répétitions : *, +, {n}, {m,n}
- ^ : début de ligne ; \$: fin de ligne ; \b : limite de mot

Grammaires



- Noam Chomsky (1956)
 - G. transformationnelles-génératives
 - Idée de grammaire universelle
- Règles de production

$$\alpha \rightarrow \beta \qquad \alpha, \beta \in (\Sigma \cup N)^*$$

N Symboles non-terminaux, métasymboles, variables

Exemple

$P \rightarrow SN\ SV$

$SN \rightarrow A\ N$

$SV \rightarrow Vi$

$SV \rightarrow Vt\ SN$

$A \rightarrow la$

$A \rightarrow une$

$N \rightarrow fille$

$N \rightarrow pomme$

$Vi \rightarrow tombe$

$Vt \rightarrow mange$

$Vt \rightarrow voit$

$P \rightarrow SN\ SV$

$\rightarrow A\ N\ SV$

$\rightarrow A\ N\ Vt\ SN$

$\rightarrow A\ N\ Vt\ A\ N$

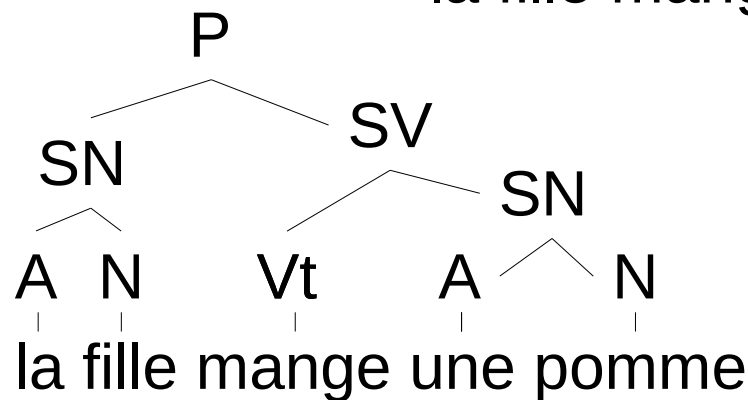
$\rightarrow la\ N\ Vt\ A\ N$

$\rightarrow la\ fille\ Vt\ A\ N$

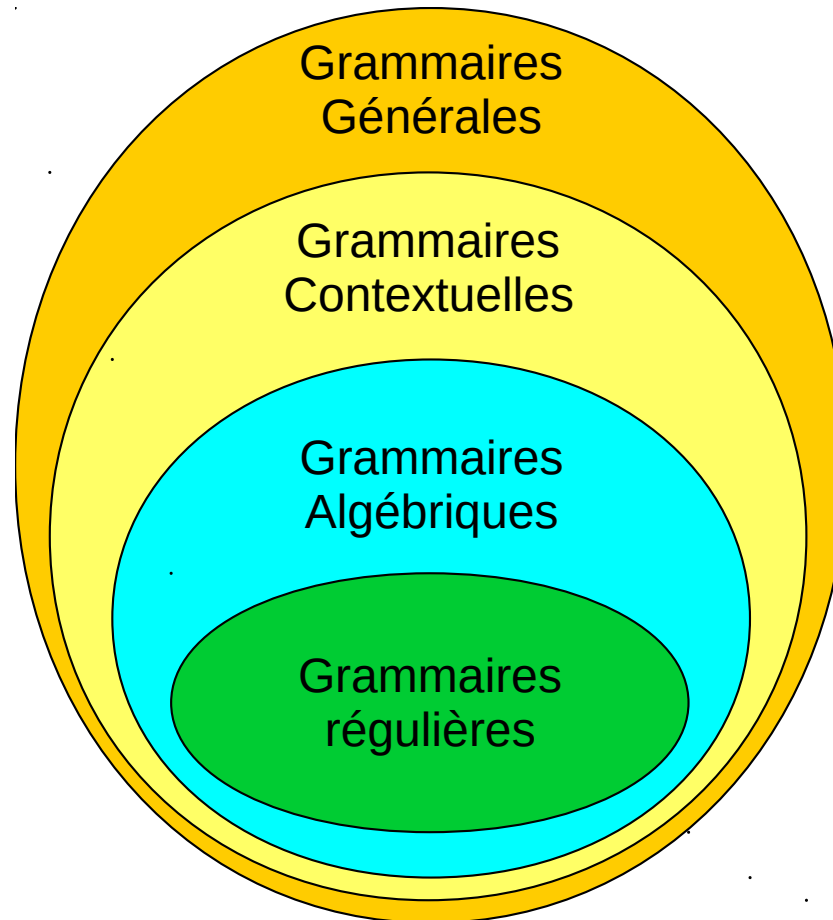
$\rightarrow la\ fille\ mange\ A\ N$

$\rightarrow la\ fille\ mange\ une\ N$

$\rightarrow la\ fille\ mange\ une\ pomme$



Hiérarchie de Chomsky

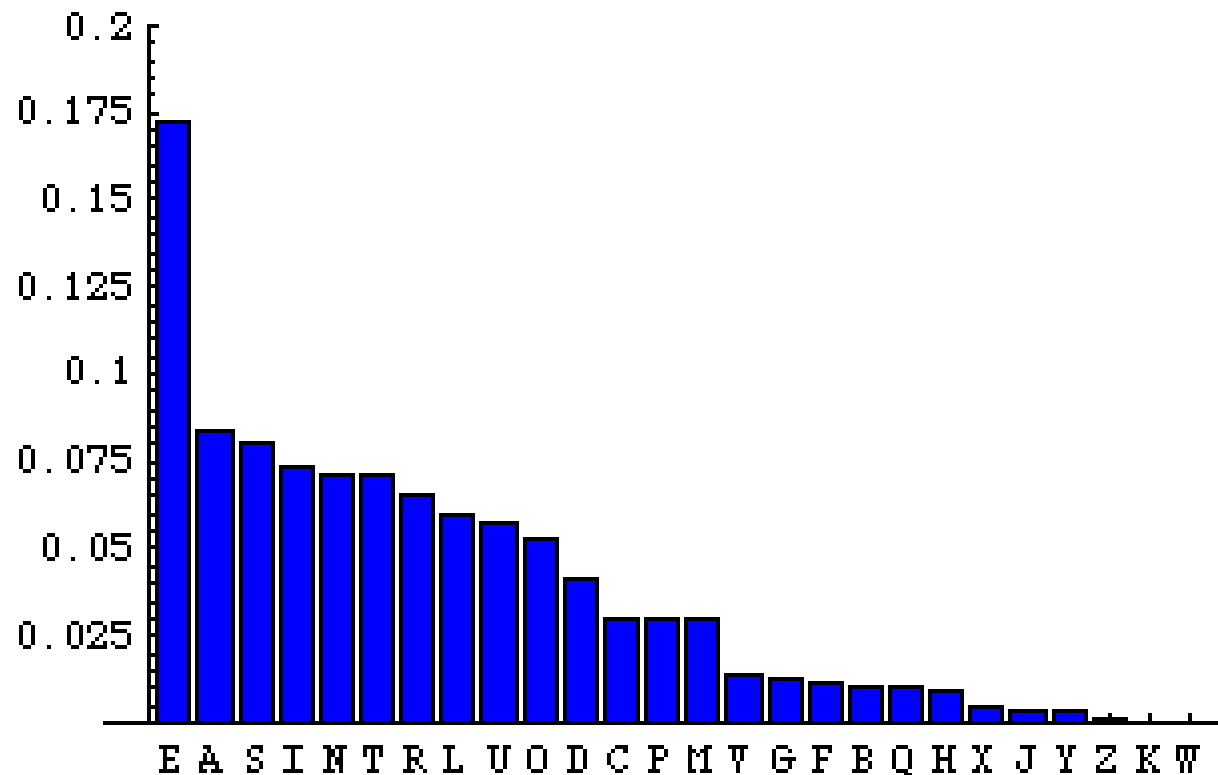


Par Fschwarzentruber — Travail personnel, CC BY-SA 4.0,
<https://commons.wikimedia.org/w/index.php?curid=52121545>

Modèles de langage

- Approximation : modélisation statistique
- Un texte est composé de caractères (ou mots)
- On pourrait décrire un langage par une distribution de probabilité sur les chaînes de caractères (ou mots, lexèmes)
- Ce genre de modèle trouve beaucoup d'applications, parmi lesquelles la reconnaissance du parlé et la traduction automatique

Modèle à unigrams



Fréquence d'apparition des lettres dans la langue française

Fréquence des lexèmes français

le dét.	1050561	de dét.	119106
de prép.	862100	ce pron.	107074
un dét.	419564	le pron.	105873
être verbe	351960	pour prép.	104779
et conj.	362093	pas adv.	103083
à prép.	293083	que pron.	99412
il pron.	270395	vous pron.	89623
avoir verbe	248488	par prép.	82277
ne adv.	186755	sur prép.	80180
je pron.	184186	faire verbe	77608
son dét.	181161	plus adv.	75499
que conj.	176161	dire verbe	72134
se pron.	168684	me pron.	71086
qui pron.	148392	on pron.	70246
ce dét.	141389	mon dét.	70121
dans prép.	139185	lui pron.	65988
en prép.	143565	nous pron.	62554
du dét.	127384	comme conj.	59902
elle pron.	126397	mais conj.	57690
au dét.	123502	pouvoir verbe	55394
		...	

Modèle à N -grams

$$w = s_1, s_2, \dots, s_{i-1}, s_i, \dots \in \Sigma^*$$

$$P(s_i \mid s_1, s_2, \dots, s_{i-1}) = P(s_i \mid s_{i-n}, \dots, s_{i-1})$$

$$P(w) = \prod_{i=1}^m P(s_i \mid s_{i-n}, \dots, s_{i-1})$$

$$\log P(w) = \sum_{i=1}^m \log P(s_i \mid s_{i-n}, \dots, s_{i-1})$$

Pré-traitement d'un texte

- Transformation dans un format utile pour l'analyse
 - tokénisation
- Nettoyage
 - Normalisation
 - Stemming ou lemmatisation
 - POS tagging
 - Correction des fautes de frappe
 - Chunking (reconnaissance d'entités nommées, extraction de termes composés)

Tokénisation

- Division du texte brut en petites unités atomiques pour les traitements qui suivent
- Ces unités peuvent être :
 - Des mots (mais qu'est-ce qu'un mot?)
 - Phrases
 - N-grams
 - D'autres unités définies par des expressions régulières

Normalisation

- Conversion de majuscules en minuscules
- Élimination de la ponctuation
- Traitement des nombres
- Élimination des mots vides (*stop words*)
 - Mots très (trop) fréquents qui ne portent pas d'informations utiles
 - Articles, auxiliaires, conjonctions
 - Pronoms
 - Quelques adverbes
 - Liste de quelques centaines (200 à 700) pour le français

Stemming et lemmatisation

- Stemming : couper les mots pour les réduire à une forme-base :
 - Utilisation d'heuristiques simples
 - Ex. : croire, croit, croyais (y → i) → croi-
- Lemmatisation : reconduire chaque forme (conjuguée et/ou déclinée) à son entrée de dictionnaire (lemma)
 - Utilisation de dictionnaires et analyse morphologique
 - Ex. : croire, croit, croyais, cru → croire

POS Tagging

- POS = *part of speech* = partie du discours
- Affecter à chaque mot du texte sa catégorie grammaticale
- Les catégories exactes dépendent de la théorie grammaticale suivie
 - P.ex. Grammaire traditionnelle (Port-Royal) : nom, pronom, verbe, adjectif, article, adverbe, préposition, conjonction, interjection
 - C'est une tâche de **classification**

Entités nommées

- Des morceaux de texte qui identifient des entités bien définies :
 - « Donald Trump », « Aéroport de Nice Côte d'Azur », « France métropolitaine », « Université Côte d'Azur », « Portail des sciences et technologies »
- Leur reconnaissance est souvent compliquée
 - [Aéroport de [Nice] [Côte d'Azur]] : 3 entités
 - Majuscules, tirets, guillemets sont des indices
- Reconnaissance → Identification (= lier à un concept dans un base de connaissances)

Mots composés

- Similaires aux entités nommées, mais au lieu d'identifier des entités définies, ils sont des lexicalisations de concepts
 - « président de région »
 - « équation aux dérivées partielles »
 - « taux d'inflation »
- Leur reconnaissance mobilise des connaissances extra-linguistique
- Souvent dépendante du domaine

Recherche d'information

- Étant donnés
 - Un corpus de documents
 - Une requête – p.ex., une liste de mots-clefs
- Renvoyer
 - Liste de documents pertinents par rapport à la requête

Modèle booléen

- Chaque document est vu comme un sac de mots
- Il peut être représenté par un vecteur binaire :
 - $d_i = 0$ si le i -ème mot est absent de d
 - $d_i = 1$ si le i -ème mot est présent en d
- La requête aussi peut être représentée de la même manière
- Un document est pertinent si le produit des deux vecteurs est > 0

BM25

Requête : $\mathbf{q} = (q_1, q_2, \dots, q_n)$

$TF(q, d)$ Fréquence du terme q dans le document d

$IDF(q)$ Inverse de la fréquence du terme q dans le corpus

$$BM25(d, \mathbf{q}) = \sum_{i=1}^n IDF(q_i) \frac{TF(q_i, d) \cdot (k + 1)}{TF(q_i, d) + k(1 - b + b \frac{|d|}{L})}$$

Typiquement, $k = 2; b = \frac{3}{4}$

Analyse syntaxique (*parsing*)

- Analyse automatique d'une séquence de mots afin de déterminer ses structures grammaticales sous-jacentes
- Les théories syntaxiques diffèrent quant à la nature des structures à identifier
 - Syntagmes : nœuds de l'arbre de dérivation
 - Dépendances : lier chaque mot à sa « tête »

Algorithme CKY

- CKY = Cocke-Kasami-Younger (1967)
- Complexité en temps $O(n^3)$ et en espace $O(n^2)$, où n est la longueur de la phrase à analyser
- La grammaire doit être en forme normale de Chomsky : règles de la forme
 - $A \rightarrow B C$ où $B, C \in N$
 - $A \rightarrow \sigma$ où $\sigma \in \Sigma$

Algorithme CKY

CKY(G, w)

$$w = (s_1, s_2, \dots, s_n)$$

$T \leftarrow$ tableau $n \times n$ de listes initialement vides

Pour j de 1 à n

 Pour toute règle « $A \rightarrow w[j]$ » en G

 Ajouter A à $T[j - 1, j]$

 Pour i de $j - 2$ à 0

 Pour k de $i + 1$ à $j - 1$

 Pour toute règle $A \rightarrow B C$ en G

 Si B est en $T[i, k]$ et C est en $T[k, j]$ alors

 Ajouter A à $T[i, j]$

Si P est en $T[0, n]$ alors succès, sinon échec

$P \rightarrow SN SV$

$SN \rightarrow A N$

$SV \rightarrow Vi$

$SV \rightarrow Vt SN$

$A \rightarrow \text{la}$

$A \rightarrow \text{une}$

$N \rightarrow \text{fille}$

$N \rightarrow \text{pomme}$

$Vi \rightarrow \text{tombe}$

$Vt \rightarrow \text{mange}$

$Vt \rightarrow \text{voit}$

Exemple

<i>A</i>				
	<i>N</i>			
		<i>Vt</i>		
			<i>A</i>	
				<i>N</i>

la

fille

mange

une

pomme

$P \rightarrow SN SV$

$SN \rightarrow A N$

$SV \rightarrow Vi$

$SV \rightarrow Vt SN$

$A \rightarrow \text{la}$

$A \rightarrow \text{une}$

$N \rightarrow \text{fille}$

$N \rightarrow \text{pomme}$

$Vi \rightarrow \text{tombe}$

$Vt \rightarrow \text{mange}$

$Vt \rightarrow \text{voit}$

Exemple

A	SN			
	N			
		Vt		
			A	SN
				N

la fille mange une pomme

$P \rightarrow SN\ SV$

$SN \rightarrow A\ N$

$SV \rightarrow Vi$

$SV \rightarrow Vt\ SN$

$A \rightarrow \text{la}$

$A \rightarrow \text{une}$

$N \rightarrow \text{fille}$

$N \rightarrow \text{pomme}$

$Vi \rightarrow \text{tombe}$

$Vt \rightarrow \text{mange}$

$Vt \rightarrow \text{voit}$

Exemple

<i>A</i>	<i>SN</i>			
	<i>N</i>			
		<i>Vt</i>		<i>SV</i>
			<i>A</i>	<i>SN</i>
				<i>N</i>

la fille mange une pomme

$P \rightarrow SN SV$

$SN \rightarrow A N$

$SV \rightarrow Vi$

$SV \rightarrow Vt SN$

$A \rightarrow \text{la}$

$A \rightarrow \text{une}$

$N \rightarrow \text{fille}$

$N \rightarrow \text{pomme}$

$Vi \rightarrow \text{tombe}$

$Vt \rightarrow \text{mange}$

$Vt \rightarrow \text{voit}$

Exemple

A	SN			P
	N			
		Vt		SV
			A	SN
				N

la fille mange une pomme

Sémantique

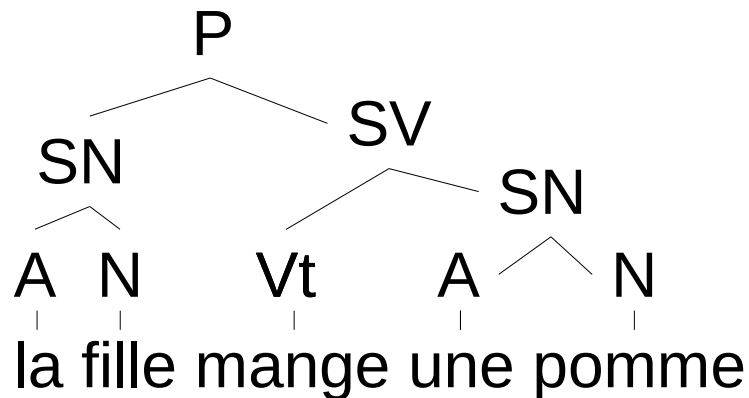
- Le but primaire du langage (naturel) est de faciliter l'échange d'idées entre ses locuteurs (en fait, de **raconter des histoires**)
- Ces idées convergent pour former le « sens » d'un énoncé ou d'un texte sous la forme d'une série de phrases
- La signification d'un texte est appelée sa **sémantique**
- Une sémantique du langage naturel parfaitement adaptée nécessiterait une théorie complète de la façon dont les gens pensent et communiquent les idées
- Comme une telle théorie n'est pas immédiatement disponible, les chercheurs en TALN ont conçu des approches plus modestes et plus pragmatiques de la sémantique des langues naturelles

Deux approches de la sémantique

- Approches basées sur la logique
 - Principe de compositionnalité locale
 - Mécanisme pour combiner les sens des constituants d'un syntagme :
 - Unification
 - λ -calcul
- Approches basées sur un modèle vectoriel
 - Analyse sémantique latente
 - Plongement lexical (*word embedding*)

Approche basée sur la logique

$P \rightarrow SN\ SV$	$[SV]([SN])$	$A \rightarrow \text{toute}$	$\lambda P.\lambda Q.\forall x(P(x) \Rightarrow Q(x))$
$SN \rightarrow A\ N$	$[A]([N])$	$A \rightarrow \text{la}$	$\lambda P.\lambda Q.\exists x(P(x) \wedge Q(x))$
$SV \rightarrow Vi$	$[Vi]$	$A \rightarrow \text{une}$	$\lambda P.\lambda Q.\exists x(P(x) \wedge Q(x))$
$SV \rightarrow Vt\ SN$	$[Vt]([SN])$	$N \rightarrow \text{fille}$	fille
		$N \rightarrow \text{pomme}$	pomme
		$Vi \rightarrow \text{tombe}$	$\lambda x.\text{tombe}(x)$
		$Vt \rightarrow \text{mange}$	$\lambda R.(\lambda y.(R(\lambda x.\text{manger}(x, y))))$
		$Vt \rightarrow \text{voit}$	$\lambda R.(\lambda y.(R(\lambda x.\text{voir}(x, y))))$



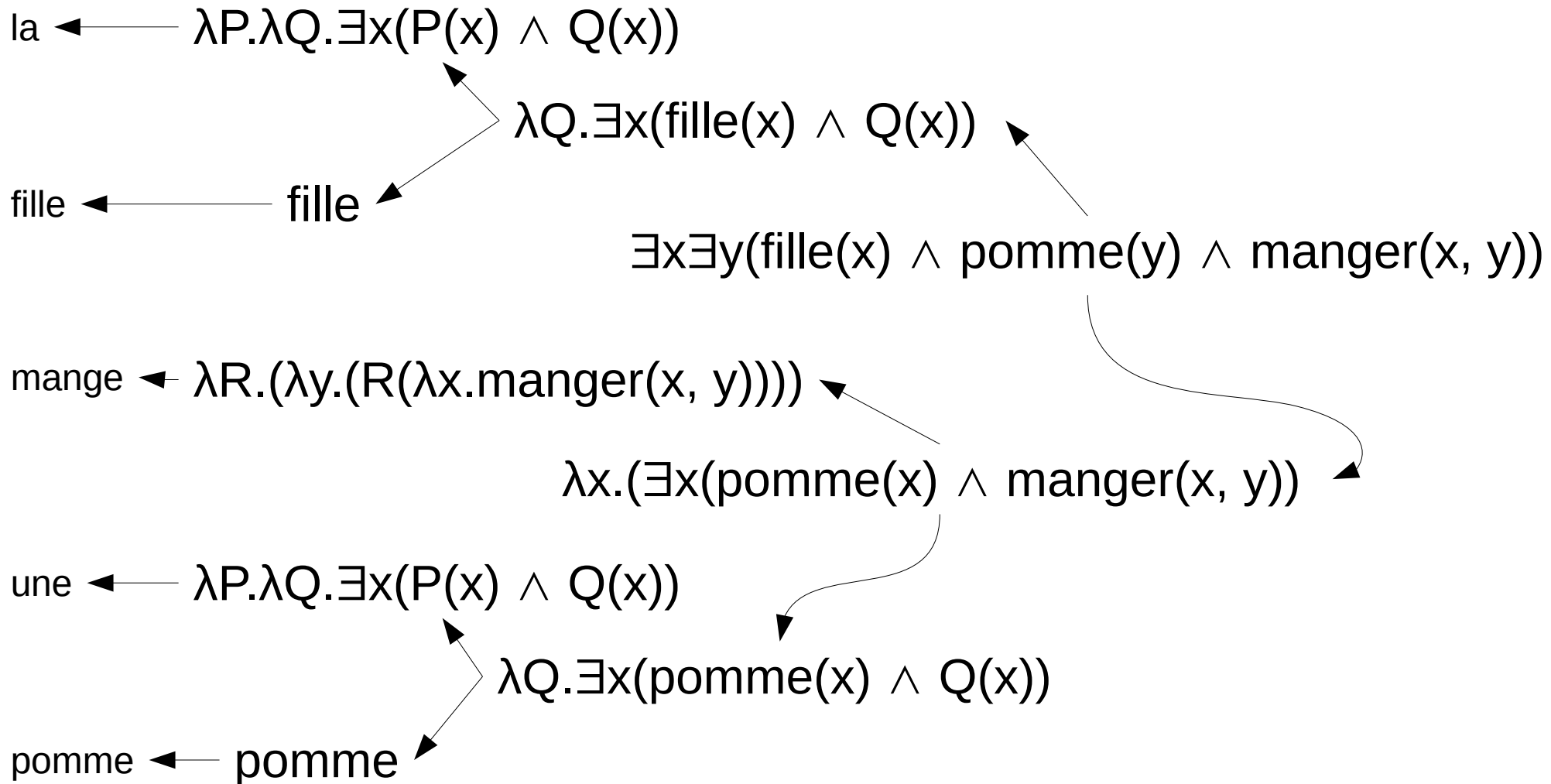
λ -calcul

$$\lambda x.a \longleftrightarrow f : f(x) = a$$

$$\lambda x.\text{tombe}(x) \longleftrightarrow f : f(x) = \text{tombe}(x)$$

$$(\lambda x.\text{tombe}(x)) (\text{pomme}) = \text{tombe}(\text{pomme})$$

Approche basée sur la logique



Agents conversationnels

- Assistants personnels basés sur un téléphone portable :
 - SIRI, Alexa, Cortana, Google Assistant
- Parler à sa voiture
- Communiquer avec des robots
- Utilisation clinique pour la santé mentale
- ... ou juste chatter pour s'amuser

Deux catégories de systèmes

- Chatbots
 - ELIZA (1966)
 - PARRY (1968), le premier à passer le test de Turing !
 - ALICE
 - CLEVER
 - Microsoft Little Bing 小冰
- Agents de dialogue (dirigés par un but)
 - SIRI
 - Interface pour voiture
 - Réservation (de vols, de restaurants, ...)

Architectures des chatbots

- Basées sur des règles
 - Règles patron-action (Eliza)
 - + un modèle mental (Parry)
- Basées sur un (grand) corpus
 - Recherche d'information
 - Réseau de neurones encodeur-décodeur

Eliza: Weizenbaum (1966)

Men are all alike.

IN WHAT WAY

They're always bugging us about something or other.

CAN YOU THINK OF A SPECIFIC EXAMPLE

Well, my boyfriend made me come here.

YOUR BOYFRIEND MADE YOU COME HERE

He says I'm depressed much of the time.

I AM SORRY TO HEAR YOU ARE DEPRESSED

...

WHO ELSE IN YOUR FAMILY TAKES CARE OF YOU?

My father

YOUR FATHER

Eliza: Weizenbaum (1966)

You are like my father in some ways.

WHAT RESEMBLANCE DO YOU SEE

You are not very aggressive but I think you don't want me to notice that.

WHAT MAKES YOU THINK I AM NOT AGGRESSIVE

You don't argue with me.

WHY DO YOU THINK I DON'T ARGUE WITH YOU

You are afraid of me.

DOES IT PLEASE YOU TO BELIEVE I'M AFRAID OF YOU

My father is afraid of everybody.

WHAT ELSE COMES TO MIND WHEN YOU THINK OF YOUR FATHER

Bullies.

DOES THAT HAVE ANYTHING TO DO WITH THE FACT THAT YOUR
BOYFRIEND MADE YOU COME HERE

Eliza: Weizenbaum (1966)

- Comment est-ce que cela marche ?
- Eliza exploite cette astuce bizarre :
 - faire semblant d'être un psychanalyste rogérien
 - Encourager le patient à en dire plus en lui renvoyant ce qu'il dit
 - Rare type de conversation dans laquelle on peut « se donner les airs de ne presque rien savoir du monde réel ».

Psychanalyste rogérien

Patient : « j'ai fait un long voyage en bateau »

Psy: « dites en moi plus sur les bateaux »

- On ne fait pas comme si le patient ne savait pas ce qu'est un bateau
- On fait l'hypothèse que le patient a un but dans la conversation
- Les *chatbots* qui cherchent à passer le test de Turing normalement choisissent de tels domaines

Règles patron-transformation d'Eliza

(* you * me) [patron]

→

(what makes you think I 3 you ?) [transformation]

Le 3, ici, signifie « le 3^{ème} membre du patron »

You hate me →

What makes you think I hate you ?

Règles d'Eliza

	patron	liste ordonnée de transformations				
mot-clé	$(K$	$((D_1)$	$(R_{1, \ 1})$	$(R_{1, \ 2})$	\cdots	$(R_{1, \ m_1}))$
		$((D_2)$	$(R_{2, \ 1})$	$(R_{2, \ 2})$	\cdots	$(R_{2, \ m_2}))$
		\vdots				\vdots
		$((D_n)$	$(R_{n, \ 1})$	$(R_{n, \ 2})$	\cdots	$(R_{n, \ m_n})))$

function ELIZA GENERATOR(*user sentence*) **returns** *response*

Find the word w in *sentence* that has the highest keyword rank

if w exists

Choose the highest ranked rule r for w that matches *sentence*

$response \leftarrow$ Apply the transform in r to *sentence*

if $w = \text{'my'}$

$future \leftarrow$ Apply a transformation from the 'memory' rule list to *sentence*

Push *future* onto memory stack

else (no keyword applies)

either

$response \leftarrow$ Apply the transform for the NONE keyword to *sentence*

or

$response \leftarrow$ Pop the top response from the memory stack

return(*response*)

Mots-clés

- Les mots-clés sont triés de spécifique à général
 - « I » est un mot-clé très général
I: (I ^{*}) → (you say you ²)
YOU SAY YOU KNOW EVERYBODY LAUGHED AT YOU
 - « Everybody » est beaucoup plus intéressant (quelqu'un qui utilise des universels comme everybody/always probablement est en train de parler d'une personne ou un événement particulier)
WHO IN PARTICULAR ARE YOU THINKING OF?
 - Réalisation : mots-clés stockés avec leur rang
Everybody 5 (règles de transformation)
I 0 (règles de transformation)
- Aucun mot-clé trouvé →
PLEASE GO ON / THAT'S VERY INTERESTING / I SEE

Implications

- Certains utilisateurs se sont sentis profondément impliqués au niveau émotionnel en dialoguant avec ELIZA
- Weizenbaum raconte l'histoire de sa secrétaire qui demandait à Weizenbaum de quitter la pièce lorsqu'elle parlait avec ELIZA
- Lorsqu'il a suggéré qu'il pourrait vouloir stocker toutes les conversations d'ELIZA pour une analyse ultérieure, les utilisateurs ont immédiatement souligné les implications en matière de vie privée
- Cela suggère qu'ils avaient des conversations assez privées avec ELIZA
- Anthropomorphisme et l'illusion de Heider-Simmel

