Theoretische Informatik und neue Anwendungen

Traitement des la Langue Naturelle

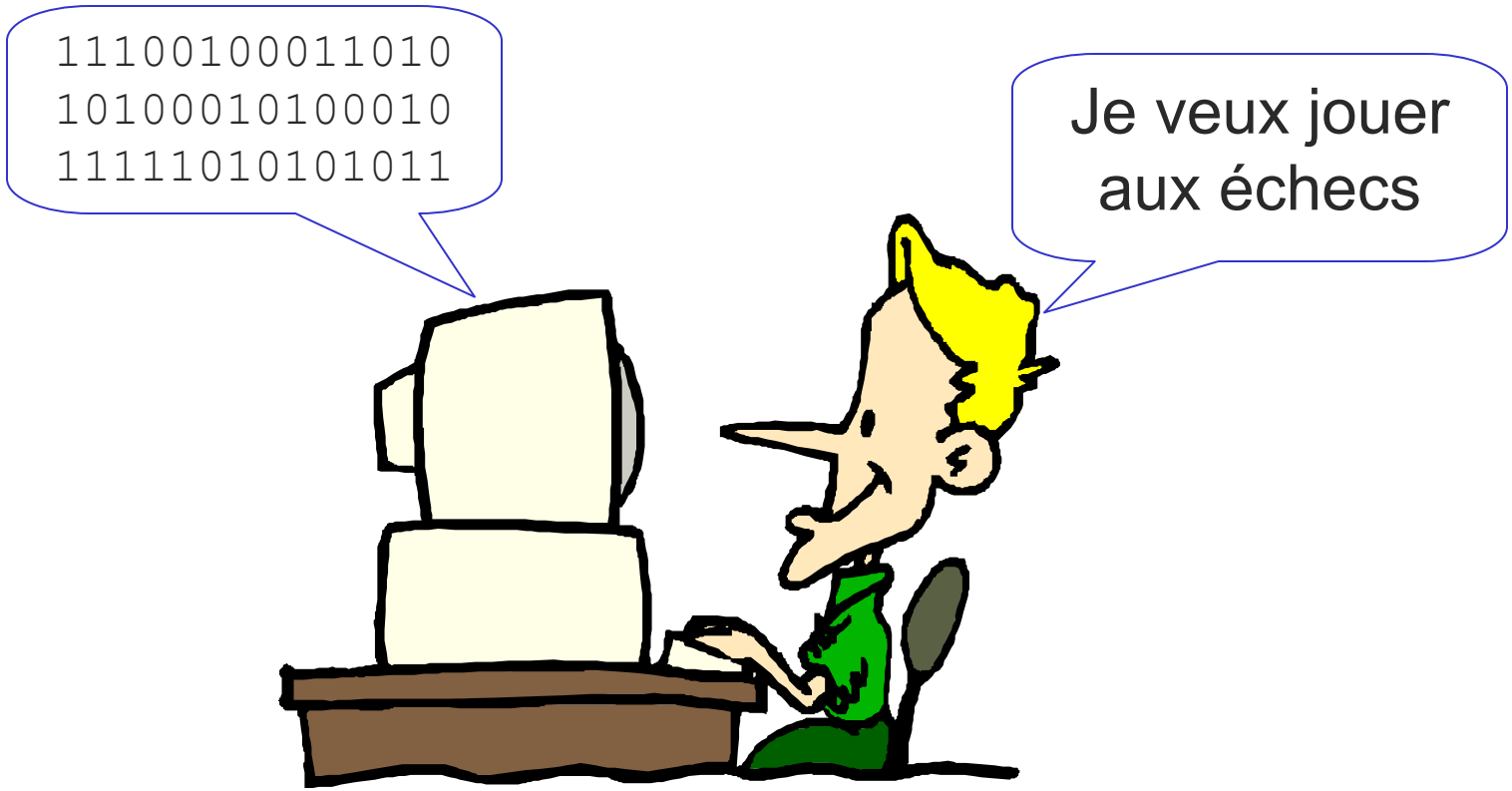
Universität Trier

Serge Linckels

Université du Luxembourg, FSTC, 6 janvier 2005

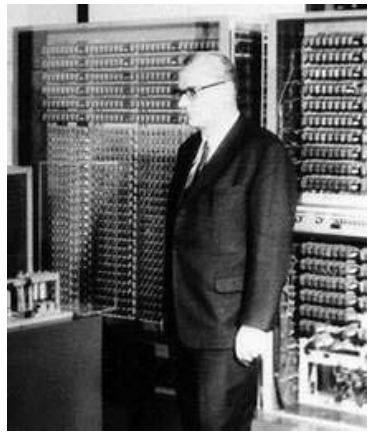
L'Interaction Homme-Machine

Universität Trier



L'Interaction Homme-Machine

Universität Trier



- 1940 – 1980 : l'Homme s'adapte à la machine
  - Les ordinateurs sont réservés aux experts
  - Les experts communiquent dans un langage très techniques (ex. Assembleur, C, VMS…)

- Années 1980 : la machine doit comprendre la langue de l'Homme
  - La machine "parle" comme un humain
  - L'utilisateur "parle" avec la machine

- L'interaction Homme-Machine est aujourd'hui un domaine en pleine évolution

Introduction

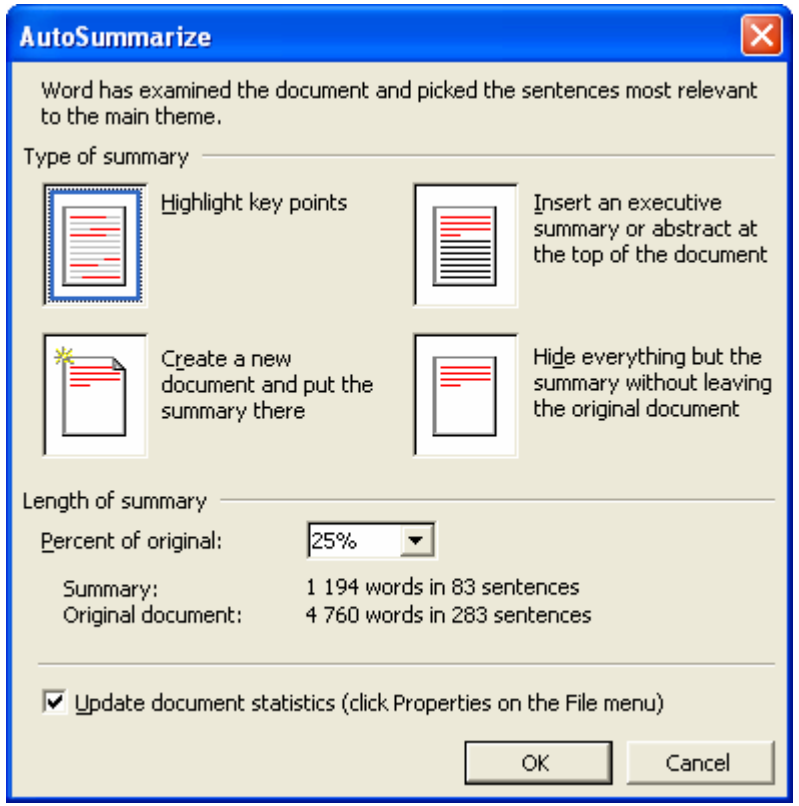
- Traitement de la langue naturelle (anglais : *Natural Language Processing*, NLP)
- Exemples d'applications :
  - OCR (*Optical Character Recognition*)
  - Reconnaissance de la voix parlée
  - Traduction automatique de textes
  - Résumer un texte
  - …
- Nouveau domaine de recherche : *Computer Linguisitic*

Réalité sur la NLP

+ OCR fiable (ex : tri P&T)

- Traducteurs très limités
- Outils de reconnaissance vocale très chers ou peu fiables (souvent à base de système d'entraînements)
- Peu d'applications fiables qui résument des textes de façon convenable

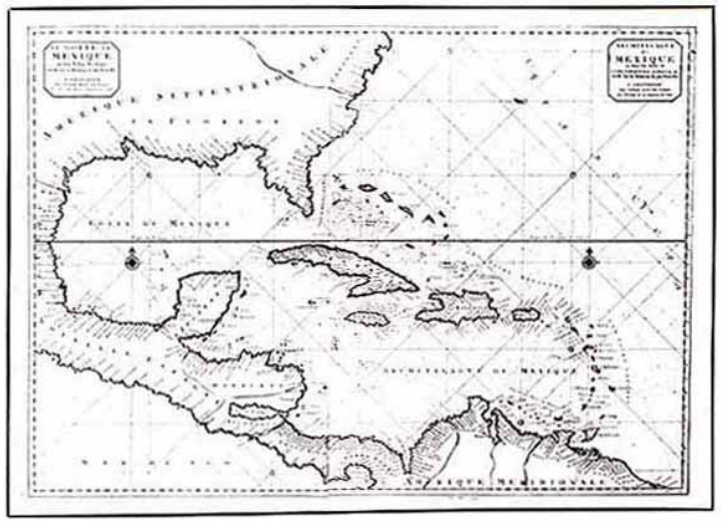
http://www.altavista.com



Problèmes de la NLP - Exemples

- Prononciation différente suivant l'utilisateur
- Nous vivons dans un monde avec des informations incomplètes et incertaines.

  Exemple : mots multimodales



Première Approche

- Conversation humaine :
  - La conversation est mise dans un contexte spécifique
  - Toutes les phrases sont interprétées dans ce même contexte
  - On se réfère intuitivement à des sources d'informations

  → Demander à la machine de faire pareil !

- Notre approche :
  - Utilisation d'une ontologie de domaine (≠ ontologie mondiale)
  - Utilisation d'un lexique (dictionnaire) pour cette ontologie
  - Effectuer une analyse sémantique (examiner les relations entre mots)

Notre Projet : CHESt

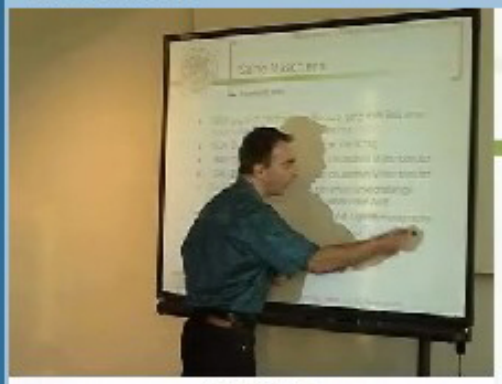
- CHESt = *Computer History Expert System*
- Caractéristiques :
  - Permettre à l'utilisateur de retrouver une ressource dans une base de connaissance → moteur de recherche
  - Permettre à l'utilisateur de formuler une requête en langue naturelle → NLP
  - Trouver seulement les résultats pertinents → recherche sémantique

http://www.linckels.lu/chest

CHEST

Computer History Expert System

zuse.smil



Theoretische Informatik und neue Anwendungen

Seine Maschinen

Universität Trier

- 1936, gibt sich nach seinem Studium ganz dem Bau einer relaisgesteuerten Rechenmaschine hin
- 1938, Z1 ist fertig, funktionierte aber nie richtig
- 1940, Z2 ist fertig und wurde vom deutschen Militär benutzt
- 1941, **Z3** ist fertig und wurde vom deutschen Militär benutzt
- Die Z3 gilt möglicherweise als der erste funktionsfähige, programmgesteuerter, digitaler Rechner der Welt
- 1944, erfindet das **Plankalkül**, eine Art Algorithmensprache

Frage: Was weisst du über den deutschen Computerpionier Konrad Zuse?

Konrad Zuse 1910 - 1995 (Erfinder)

Metadata

[Zeitlinie]

[Spezial]

Laden | Notizen | Notizen anzeigen

Recherche par Mots-Clés

- Recherche systématique
  sans tenir compte de la sémantique

- Améliorations :
  - Ranking : les documents qui contiennent plus de mots clés sont plus pertinents
  - Ne pas considérer les "Stop Words" (ex. un, une, de, à)
  - Stemming : analyse morphologique (ex. réduire les conjugaisons à un tronc commun
  - Loi de Zipf : tenir compte de l'importance des mots
  - autres

Lois de Zipf (1949)

- George Kingsley Zipf (1902-1950), prof. à Harvard
- Améliorer par Benoit B. Mandelbrot (1924-) en 1954
- Idée :
  - Classement des mots suivant leur fréquence
  - Il existe une constante k tel que $f \cdot r = k$
  - Cela veut dire que p.ex. le mot à la 50e place devrait apparaître 3 fois plus souvent que le mot à la 150e place

mot	freq (f)	rang (r)	$f \cdot r = k$
the	3332	1	3332
and	2972	2	5944
a	1775	3	5235
…			
he	877	10	8770
but	410	20	8400
be	294	30	8820
there	222	40	8880
one	172	50	8600

Modèle Vecteur (1968)

- Permet de calculer le degré de similitude entre deux documents
- Base théorique :
  - Ensemble des documents : $D = \{d_1,\ldots,d_k\}$
  - Vecteur des poids des mots : $\vec{d_j} = (w_1,\ldots w_m)$
  - Vecteur requête : $\vec{q} = (w_1,\ldots w_n)$
  - Similitude entre document et requête :

$$sim(d_j,q) = \frac{\vec{d_j} \bullet \vec{q}}{\left|\vec{d_j}\right| \times \left|\vec{q}\right|}$$

$$= \frac{\sum_{i=1}^{t} w_{i,j} \times w_{i,q}}{\sqrt{\sum_{i=1}^{t} w_{i,j}^2} \times \sqrt{\sum_{i=1}^{t} w_{i,q}^2}}$$

Recherche Sémantique

- Ne pas considérer la requête comme un flux de mots mais comme une entité
- Détecter la relation entre les mots
  → Analyse structurelle
- Comprendre la relation entre les mots
  → Analyse contextuelle

Analyse Structurelle

- Structure d'une phrase : sujet, verbe, adjectif
  - sujet : personne, animal, concept, chose
  - verbe : exprime l'action dans une phrase
  - adjectif : décrit des propriétés du sujet
- Problèmes : certains mots peuvent être membre dans plusieurs classes
  - Exemple : Children eat sweet candy
- Pour l'informaticien : Utilisation d'outils linguistiques (ex. analyseur lexical)

Analyse Contextuelle

- Objectif : Trouver la signification pertinente d'un mot dans le contexte de l'ontologie

- Une source sémantique (lexique) est nécessaire pour le NLP, p.ex. *WordNet*

  http://www.cogsci.princeton.edu/~wn/

- Malheureusement souvent un manque :
  - Production est très coûteuse
  - Saisie d'un dictionnaire classique (livre) est très difficile
  - Machine n'a pas d'accès aux informations contextuelles

Vocabulaire Sémantique

- Hyperonyme : mot avec un sens plus général

  Exemple : animal est un hyperonyme de chat

- Hyponyme : mot avec un sens plus spécifique

  Exemple : chat est un hyponyme d'animal

- Antonymes : mots avec un sens opposé

  Exemple : chaud et froid sont antonymes

- Synonymes : mots avec même sens

  Exemple : voiture et automobile sont synonymes

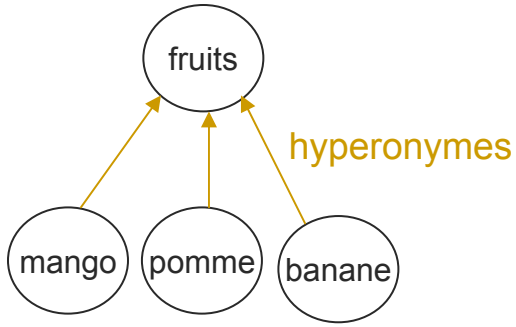
- Homonymes : mots qui sont écris de la même façon mais ayant des sens différents

  Exemple : Pascal = {prénom, unité de mesure, langage de programmation}

Similarité Sémantique

- Impossible de représenter un sens de façon opérationnelle pour une machine
- On se contente souvent de calculer la similarité sémantique entre mots, phrases ou documents
- But : quantifier cette similitude
- Exemple : « Alice aime les mangos »

  Si le système connais pas les mangos mais sait
  - que mango est hyponyme de « fruits », et
  - que fruits et hyperonyme de « banane, pomme »,
  - alors le système peut conclure que mango est sémantiquement similaire à banane et pomme



Idée de Base pour CHESt

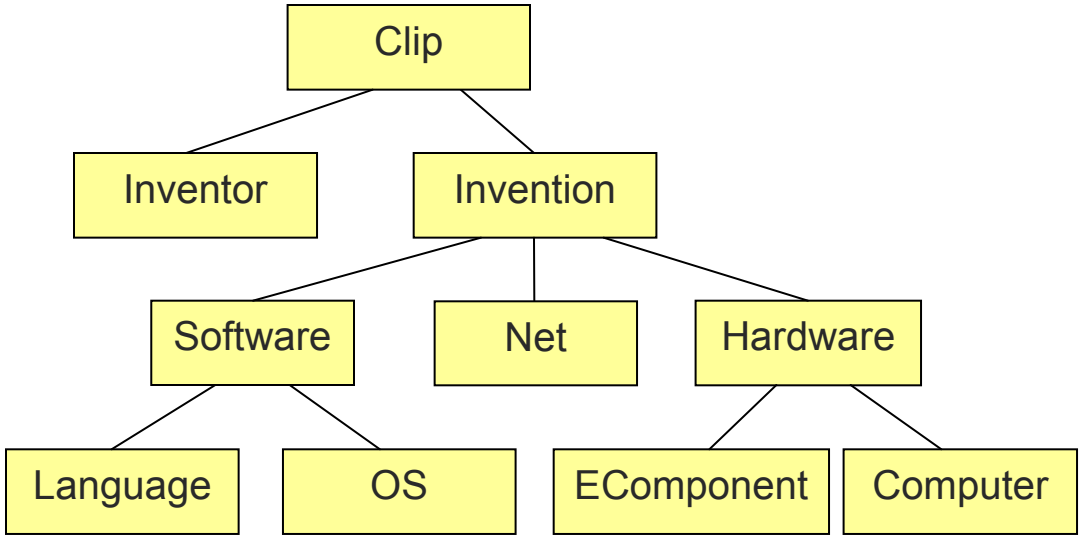
- "Système Bibliothécaire Intelligent" pour des leçons en ligne
- Propriétés :
  - Le système ne donne pas la réponse à la question de l'utilisateur, mais lui retourne un document (une ressource) dans lequel l'utilisateur trouvera la réponse
  - Le système doit comprendre la question de l'utilisateur
  - Le système doit maîtriser l'organisation interne de la base de connaissances

Motivations Pédagogiques

- Forme « attractive » de l'information
- Réponses courtes
- Grand potentiel d'informations
- Base de connaissances sécurisée et fermée
- Garantie pour la qualité de l'information
- Simple à administrer (add/remove)
- Interaction humaine et simple
- Recherche d'informations facile
- Accès facile

Ontologie de Domaine

- Ontologie de domaine = { taxonomie, langage }
- Taxonomie : classification de concepts dans une hiérarchie
- Exemple pour l'histoire des ordinateurs :



Interprétation d'une Question

- Interprétation d'une question: $\mathcal{I}\left[q\right]_g^H = R$
  - g fonction de *mapping*
  - H taxonomie
  - R *Ensemble de documents pertinents*

q = “What did Konrad Zuse invent?”

↓ **Sentence interpretation**

Φ = {(dc:creator;“invent"),(chest:Inventor;“Zuse")}

↓ **Assertion mapping**

a_q = An invention was invented by one or more inventor(s)

↓ **Query generation**

R = SELECT WHERE (?x;dc:creator;“Zuse")

Interprétation d'une Question (1)

- Tous les mots du dictionnaire (L_H) sont classés dans la taxonomie (H)

Concept: **Operating System**
s = chest:OS *T* = { Windows, Linux, MsDOS, VMS } type = object

- Tous les mots de la question $q = (w_1, ..., w_n)$ sont attachés, si possible, à une interprétation dans la taxonomie avec la fonction $wi()$

w	*wi(w)*	**#interprétations**
the	{ }	0
Windows	{ (v_α,"Windows",0) }	1
Ada	{ (v_β,"Ada",0), (v_χ,"Ada",0) }	2

Distance de Levenshtein (1965)

- Quantifier la similarité entre deux mots
- Exemple : ABCDE et ABXDFE

A	B	C	D	E	F	
A	B	X	D	X	E	F

DL = 2

1 modification

1 suppression

http://www.merriampark.com/ld.htm

Interprétation d'une Question (2)

- Effectuer une analyse sémantique (ne garder que les mots pertinents) avec la fonction $p()$

> q = "Who invented the very first transistor?"
> $p(q)$ = "who invented the very first transistor"

- Affecter toute la question à une assertion générale avec la fonction $g()$

> q_1 = "Who invented the very first transistor?“
> $g(q_1)$ = An invention was invented by an inventor
>
> q_2 = “Did Konrad Zuse once met the American H. Aiken?“
> $g(q_2)$ = Somebody is related to somebody

Interprétation d'une Question (3)

- Enrichir l'assertion générale avec les mots pertinents de la question et générer une requête

q_1 = "Who invented the very first transistor?"
$g(q_1)$ = An invention was invented by an inventor

Invention = "transistor"
Inventor = ?x

SELECT Inventor WHERE Invention = "transistor"

Présentation

Universität Trier

DÉMO

…enfin