Chap. 04: Natural Language Processing for Information Retrieval

Jean-Pierre Chevallet

LIG-MRIM

2018

Outline

1. Indexing with textual descriptors
   - Indexing pipeline
   - Morphology for text indexing
   - Stemming
   - Filtering and counting
2. Descriptor coordination
   - Descriptor coordination ressources
3. Morpho-Syntax
   - Part Of Speech
   - Phrase and Terminology
4. Syntax, Semantics and Concepts

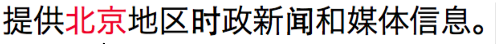
Table of Contents

Computing the Matching

Matching

Does the information inside the document enough to compute the matching ?

北京


提供北京地区时政新闻和媒体信息。

No external resources

An IRS without resources as its matching limited to "identity"

Text Indexing Elements

- **Descriptor**: element of the index, can be atomic or structured
- **Annotation**: select descriptors, relevant to document content

- **Controlled** Indexing: a fixed list of descriptors
- **Free** Indexing: no 'a-priori' list is given
- **User oriented** indexing: pre-select descriptors a user may use
- **Document oriented** indexing: only descriptors extracted from document content.

Indexing

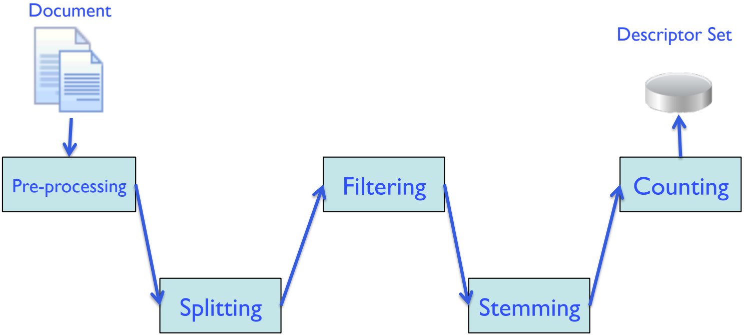
Put descriptors from annotation step, into an efficient structure (ex: inverted file).

Indexing Parameters

To be chosen :

- Descriptors (also called "index terms")
- IR Model
  - Document representation
  - Matching Function
  - Query language and representation
- Automatic text indexing: method to extract descriptors from text

Automatic text indexing: Indexing pipeline



Pipe line Elements

Pre-processing (lexical)

Digits, hyphens, punctuation marks, case of letters

Splitting (Morphological)

Into descriptors

Filtering (Grammatical)

Elimination useless descriptors (ex: tools word)

Stemming (Normalisation)

Approximation (ex: porter), or grammatically correct with POS (lemmatisation)

Splitting: Language Analysis Levels

- Morphology: words, stemmed words
- Morpho-Syntax: terms, nouns, named entities
- Syntax: phrase, sentence (link between words)
- Semantics : acception, concepts

Morphology

Morphology

"Morph" = shape

"logy" = study

So : Morphology is the study of the shape of words.

Why it can be useful to IR ?

Because is can help to build or select de text descriptors.

Language: Speaking and Writing units

Phoneme: atomic audio (Phonetic) unit of a language. Role: to unify different sound, accent.

Grapheme: atomic visual (Graphic) unit of a language. Role: to unify different graphical variant, ex: 'a', 'A', 'α'

- Pictograms: concrete representation of an objet (explicit drawing)
- Ideogram: abstract representation of object (no direct link with object shape).
- Phonogram: no representation of object, only abstraction of the sound (Phoneme), i.e. Letters !

Language: Speaking and Writing units

Phoneme: no meaning, because only related to "sounds" of a language.

Morpheme: the smallest unit in a language that have a meaning. Free Morpheme: can exists alone, as bound morphemes can't stand alone as words.

Example of Morpheme:

- "pollutio": means "dirty" in Latine, hence the morpheme "pollu-"
- "-tion": means the result of an action, hence "pollu-tion"

Language: what is a "word" ?

What is a "word" ?

A free morpheme, or a composition of morpheme.

Hence, a "word" is the smallest linguistic form, that have its own autonomy, and so having a meaning.

Caution

- A word can include space: ex: "hot dog", "White House", in Vietnamese most words have 2 Ideograms.
- Several words can be agglutinated: "naturwissenschaft" = "nature wissen shaft" (German)

Words as index

Definition of a word: depend of the language.
Select words more prone to be used as index:

- Depending on frequency : *idf* .
- Depending on word themselves: stop words.
- Depending on Part Of Speech (POS).

Unify words (plural forms, etc.)
reducing inflected or derived words to their stem, base or root form.
With/without POS analysis

"Stem", "Root" and "Term"

Root: a morpheme that expresses the basic meaning of a word and cannot be further divided into smaller morphemes ex: "struct"

Stem: the root plus another morpheme to which other morphemes can be added ex: "construct", "structural"

Term: a sequence of words that have a unique meaning in a given domain

One of the longest German term: "Rindfleischetikettierungsüberwachungsaufgabenübertragungsgesetz"

Descriptor Choice

To define what will be a **IR descriptor**:

- Grapheme: possible but weak (set of grapheme is small).
- Phoneme: possible but not related to meaning, but sound. Good if indexing also speech as sound.
- Morpheme: possible good choice, but need NLP for splitting words.
  - Lexical Morpheme (stem) : have a meaning, stable form, good choice for IR
  - Grammatical Morpheme: almost useless for indexing.
  - n-gram: sort-of artificial (i.e. incorrect) morpheme, but still gives interesting results.
- Word: easy choice in some language (not german !), but usually derivation, specially grammatical is a problem.

Descriptor extraction from text

- Split text into linguistic units: paragraphs, sentence, phrase, words, morpheme, stem, ...
- Need at least basic NLP treatments depending of the language

Problems

Overlap between syntax and semantics: punctuation can delimit sentences but also abbreviations.

The **pipe line process** is not a general correct solution .... but very often used in IR.

Stemming

Example:

if the word ends in 'ed', remove the 'ed'
If the word ends in 'ing', remove the 'ing'
if the word ends in 'ly', remove the 'ly?

Depends on language

Affix, suffix
Agglutination (German, Swedish,?)

"Naturwissenschaftlichen Fakultate"
Natur+wissen+schaft+lichen
Natur+wissenchaft+lichen

Stemming Algorithms

Lemmatisation algorithm

Based of normalization rules that depend of word Part of Speech.

- + controlled small set of rules
- + correct computation
- + POS tagger can infer the POS of new or incorrect words
- - need a POS tagger
- - problem when incorrect POS tagging
- - strong effort to set up the rules

Possible good solution for IR, if indexing results are better than the Suffix-stripping algorithm. Depend on the langage complexity.

Stemming Algorithms

Stochastic algorithms

Learning approach of rules based on a set of example.

- + easy to include a new language
- - need a good set of example
- - no guaranty for correct stemming

Possible also good solution for IR specially to set up quickly a stemmer for new language.

Stemming Algorithms: Porter algorithm

Porter algorithm

A well known example of an Affix-stripping algorithm for English

1) Set a category for grapheme based on phonology.

- A consonant C is a letter other than A, E, I, O or U, and other than Y preceded by a consonant.
- If a letter is not a consonant it is a vowel V.

Ex: TOY $\rightarrow$ CVC

Stemming Algorithms: Porter algorithm

2) Compute the signature of the word

- Repetition of C or V are reduced to one.
- Any word can be represented by $[C](VC)^m[V]$

Example

"Organisation" → $VCCVCVCVCVVC \rightarrow VCVCVCVCVC \rightarrow$ $[.](VC)^5[.] \rightarrow m = 5$

m is called the measure of a word.

- m=0: TREE, BY
- m=1: CAR, PLAY, LOOK,
- m=2: RULES
- m=3: VISIT

Stemming Algorithms: Porter algorithm

3) Set up a rules

- Rule of the form: (condition) S1 → S2.

If a word ends with the suffix S1, and the stem before S1 satisfies the given condition, S1 is replaced by S2

Conditions (examples):

- on m: $m > 1$
- $*s$: the stem ends with "s"
- $*V*$: the stem contains a vowel.
- $*D$: the stem ends with a double consonant. Ex: "off"
- $*O$: the stem ends CVC, where the second C is not W, X or Y. Ex. "hop"
- any logical composition (AND,OR,NOT)

Set of rules written beneath each other, only one is obeyed. The one with the longest matching.

Stemming Algorithms: Porter algorithm

4) The algorithm (part of ..)
Step 1a

- "sses" → "ss"
- "ies" → "i"
- "ss" → "ss"
- "s" → ""

Eg.

- caresses → caress
- ponies → poni
- caress → caress
- cats → cat

Stemming Algorithms: Porter algorithm

4) The algorithm (part of ..)
Step 1b

- $(m > 0)$ "eed" → "ee"
- $(*V*)$ "ed" → ""
- $(*V*)$ "ing" → ""

Eg.

- agreed → agree
- mastered → master
- motoring → motor

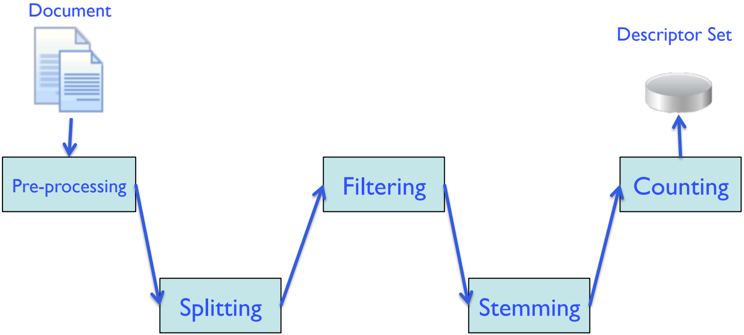
Then step 1c, step 2, 3, 4 and 5 (a,b).
See: http://snowball.tartarus.org/algorithms/porter/stemmer.html

Stemming Algorithms: Porter algorithm

- revival → reviv
- allowance → allow
- replacement ... → replac
- communism → commun
- electricity → electriciti → electric → electr
- electricaly ... → electr
- hopefully ... → hop
- hoping ... → hop

Porter Algorithm is good for recall, and reduces descriptor set.

Automatic text indexing: Indexing pipeline



Filtering

Filtering the descriptors

- A stop list: a explicit list of descriptor to remove (Eg. "is", "the", "a", ...)
- Some characteristic of the words to remove: eg. Part Of Speech.
- Some word distributions: ex. if appears in the all collection (cf. *IDF*).

Filtering reduce descriptor set but could harm document recall.

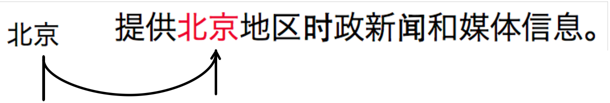
Counting

- Exact: frequency of exact occurrences
- Structured: more complex count for structured descriptors
- Elision: count also references: complexe, need NLP

Ex: "Richard Nixon was the 37th President of the United States who served from 1969 to 1974, when he became the only ...

Table of Contents

Link between descriptors



提供北京地区时政新闻和媒体信息。

How to escape this "simple" matching ?

- Coordination: "pre" and "post"
- Using link between descriptors: using external resource, thesaurus, meta-thesaurus, ontologies
- Taking into account variation.
  - Terminological variation: Pollution, pollute, pollutant : pollu
  - Semantic variation: petrol, petroleum, fuel oil, oil, gasoline, hydrocarbon, fuel
  - Contextual synonym: "Black gold" can be "oil", "Coal", "Black pepper", ...

Descriptor Coordination

Coordination

To compose simple descriptors, to build more complex expressions

Pre-Coordination

A Coordination at Indexing Time
Ex: "services and repairs for motor vehicles" "garage"
Increase recall, or precision (if term extension)

Post-Coordination

A Coordination at Querying Time
Ex: coordinate simple terms into a query
"repair" $\wedge$ "car"

Ressource for Coordination

Static ressources

That are independent from documents.
Already exists.
Ex: Thesaurus, Dictionary, Terminology, ontology

Dynamic Ressources

Software that takes documents as input.
Ex: Stemmer, "rooter", Part of Speech Phrase Analyzer, Surface Phrase,or Sentence Analyzer

Endogenous vs. Exogenous Ressource

From and external collation of data, or extracted from document themselves.

Static Ressources

Thesaurus : to link descriptors. Different type of links.

Dictionary : to describe words meaning and usage. To record what words have meant to authors in the distant or immediate past.

Lexical DB : to link lemma based on meaning (ex: WordNet).

Terminology : to describe all terms from a given domain, for translation (ex: "Le grand dictionnaire terminologique (GDT)")

Ontology : to describe concepts (even to related to words or terms), using formal description (logic). Ex: CyC. Caution : a "Lexical Ontology" is a Terminology

Lexicon versus n-gram

Indexing a Vietnamese Corpus. Words are composed of 2 graphemes in Chinese, and transformed in Latin characters, then 2 Morpheme. Comparison with a lexicon.

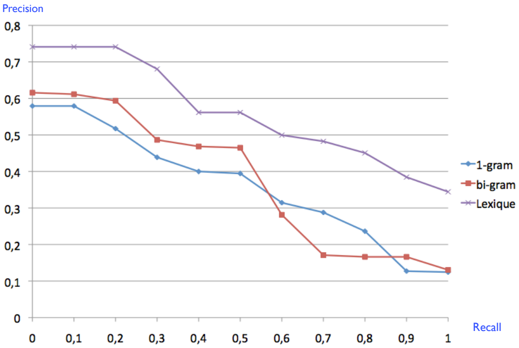


Table of Contents

1 Indexing with textual descriptors
- Indexing pipeline
- Morphology for text indexing
- Stemming
- Filtering and counting

2 Descriptor coordination
- Descriptor coordination ressources

3 Morpho-Syntax
- Part Of Speech
- Phrase and Terminology

4 Syntax, Semantics and Concepts

Part Of Speech

What ?

- A closed list of tags: Noun, Adjective, Verb, etc. How long is this list ?
- A POS tagger: associate a tag to each word

Type of algorithms

- Lookup Algorithms: ok for closed list of terms (tool words)
- Rule Based Algorithms: need to manually set up rules.
- Stochastic Algorithms: simples rules are learns from example (eg. TreeTagger).

Characteristic for IR

- Robust to new terms or misspelling
- Use of morphology to guess POS.
  Eg. The Smurfs "We're going smurfing on the River Smurf today"

Named Entities

What is it ?

- Is an unique reference of an element in the real world.
- Organization, locations, people, measures (weight, money, ...)
- Important in IR because strong semantics, and discriminating

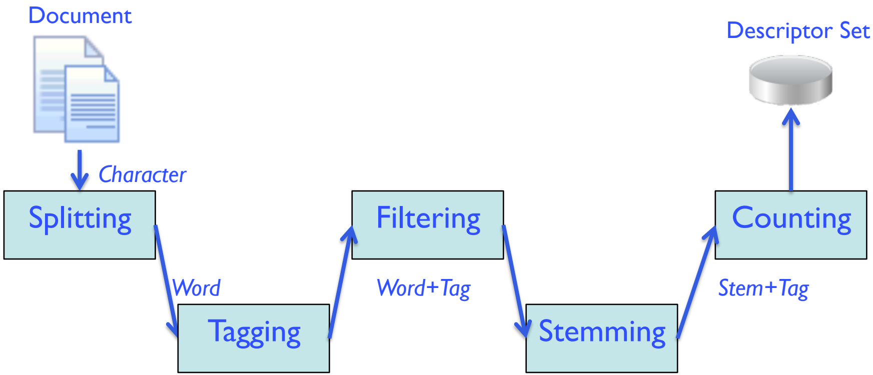
How to detect ?

- Lookup Algorithms: a list.
- Syntactic Pattern: need Part Of Speech, or Surface analyzer.

How difficult ?

- Intersection with other common names (bill + gate, white + house)
- Extension or abbreviation: AIDS, international business machines
- Idiomatic form : "big blue"

Text Indexing Path with NLP Part Of Speech



Phrase

After POS tagging, one can analyze a sentence into "phrase".

Phrase

A Phrase is a sub part of a sentence with a grammatical structure that play a role into the global structure of the sentence.

- Noun Phrase: role of subject, or object.
- Verb Phrase: action

Phrase

”There have been many assassination attempts and plots on Presidents of the United States”

- ”have been”
- ”many assassination attempts and plots on Presidents of the United States”
- ”many assassination attempts”
- ”assassination attempts”
- ”many assassination plots”
- ”Presidents of the United States”
- ”the United States”

Term and Terminology

Term

A Term is a Noun Phrase with a unique and clear semantics attached to a knowledge Domain.

Terminology

A Terminology is the study and list construction of all Terms of a Domain.

Terminology

Computer Terminology

- "computer"
- "keyboard"
- "mouse"
- "bug"
- "virus"
- ...

Terminology

Medical Terminology

- "medication"
- "physician"
- "mouse"
- "bug"
- "virus"
- ...

Terminology

How to build a Terminology ?

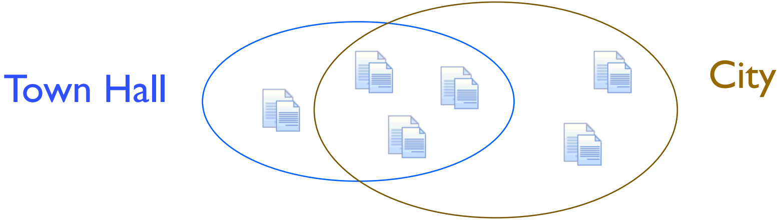
- Manually: high quality, but high cost (ex: Library of Congress Subject Headings, UMLS)
- Automatically: analyze lots of texts from the same domain.

One can also compute automatically links between terms, and create an *Association Thesaurus*

- Using Documentary context
- Lexical context

Documentary context

The link strength is relative to the amount to shared context as amount of **text part** where these two terms appears.

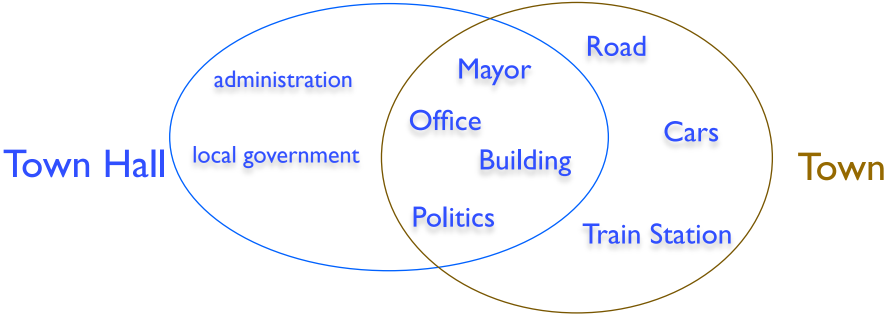


Parameters

- Scope: size of text part, from a sentence to a whole document/
- Filter: the part that are used. Eg; only title, only noun phrase

Lexical context

The link strength is relative to the amount to shared context as amount of **shared word**.



Context computation

Symmetric measures:

- $cos(X,Y) = \frac{|X \cap Y|}{\sqrt{|X| \times |Y|}}$
- $dice(X,Y) = \frac{2 \times |X \cap Y|}{|X| + |Y|}$
- $jaccard(X,Y) = \frac{|X \cap Y|}{|X \cup Y|}$
- $tanimoto(X,Y) = \frac{|X \cap Y|}{[X| + |Y| - |X \cap Y|}$
- $overlap(X,Y) = \frac{|X \cap Y|}{min(|X|,|Y|)}$
- ...

Context computation

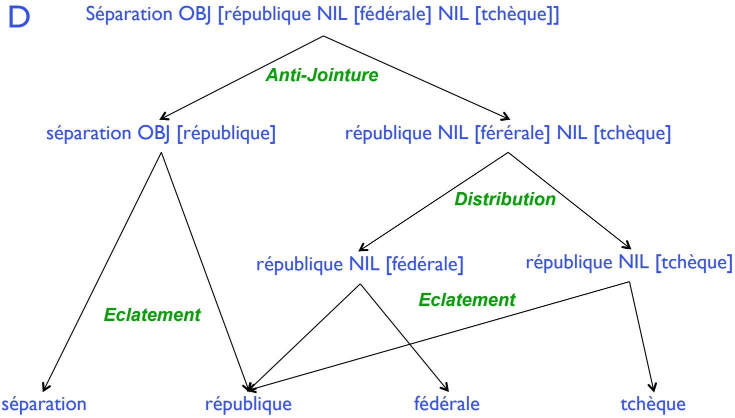
Non Symmetric measures, $X \subset E$, $Y \subset E$:

- $support(X,Y) = \frac{|X \cup Y|}{|E|}$
- $confidence(X,Y) = \frac{|X \cap Y|}{|X|}$

Comes from association rule learning for data mining:

- $support(X,Y) = P(X \cup Y)$
- $confidence(X,Y) = P(X \rightarrow Y) = P(Y|X)$

Term matching



D

Séparation OBJ [république NIL [fédérale] NIL [tchèque]]

Anti-Jointure

séparation OBJ [république]

république NIL [férérale] NIL [tchèque]

Distribution

république NIL [fédérale]

république NIL [tchèque]

Eclatement

Eclatement

séparation

république

fédérale

tchèque

Word Latent Spaces

- Continuous representation for word meaning.
- Based on documentary context: Latent Semantic Indexing (Deerwester, 1988)
- Based on the lexical context (cooccurrence) : Word embedding, Conceptual Vectors
- Capture finer grain information than graph output from text mining

Origin in 2000 with "A Neural Probabilistic Language Model" , Yoshua Bengio in Montreal.
But also other origines like Didier Schwab and Mathieu Lafourcade 2002 in France Montpellier : "Antonymy and Conceptual Vectors", work influenced by Jacques Chauché (1990)

Conceptual Vectors

The French experiments:

- Exploiting existing ressources like dictionnary
- Build a space by modification of points in space
- Model not clearly formalized
- No usage in IR
- Does not spread among other reserchers

Word embedding

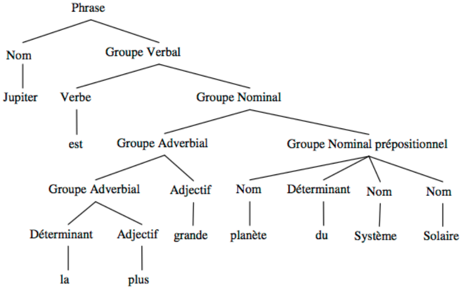
The "deep leaning" popularity effect and Google experimentations:

- Now view as a learning problem
- Reduction of computation complexity: Tomas Mikolov in 2013 (word2vec)
- Enhancement of the quality of the vectors: $\vec{king} - \vec{woman} \simeq \vec{queen}$
- Start to be used in IR: bag-of- embedded-words (BoEW)

Table of Contents

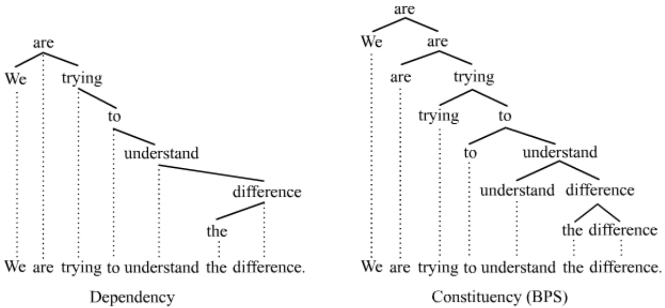
Grammar Types

- Symbolic Grammar (vs. Statistical)
- Lexical functional grammar (constituency): view language as being made up of multiple dimensions of structure represented as a distinct structure with its own rules. Composition of structures.



Grammar Types

- Dependency grammar: based on dependency relation that views the verb phrase as the structural center of all clause structure.



Wikipedia

Grammar Usage

- Shallow parsing: identifies the constituents (noun groups, verbs, verb groups, etc.), but does not specify their internal structure, nor their role in the main sentence.
- Partial parsing: only some phrases.

Grammar Usage

- Shallow parsing: identifies the constituents (noun groups, verbs, verb groups, etc.), but does not specify their internal structure, nor their role in the main sentence.
- Partial parsing: only some phrases.

In IR:

- Robust: should always produce a result.
- Useful at least for query parsing.
- Help in solving some linguistic phenomenon like **Anaphora**
- Must define tree matching !

Concept vs. Acception

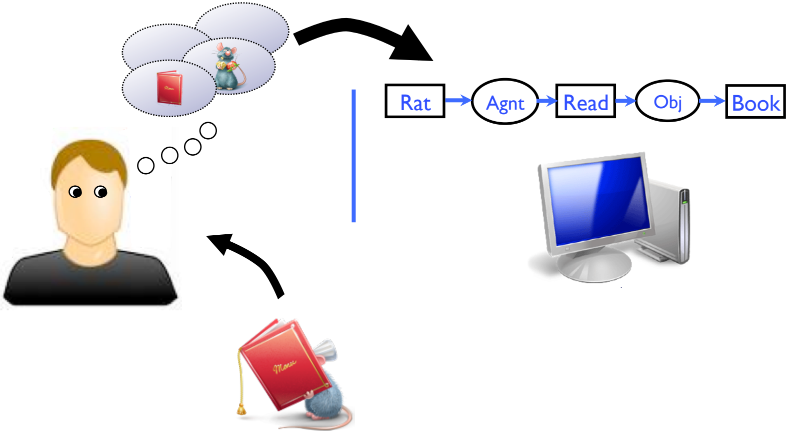
Acception

Meaning of a word based on its usage in the language.
Eg. Synset of Wordnet.

Concept

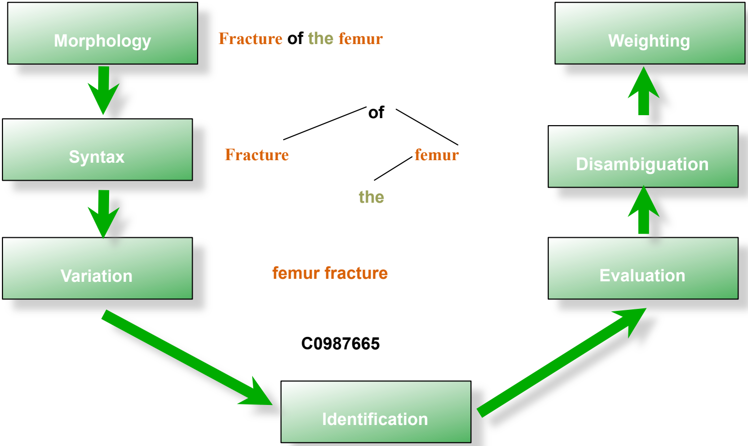
Abstract entity, that unify a set of concrete or mental object by performing an abstraction of common relevant attributes.
Eg. Concept of CyC.

Concept construction

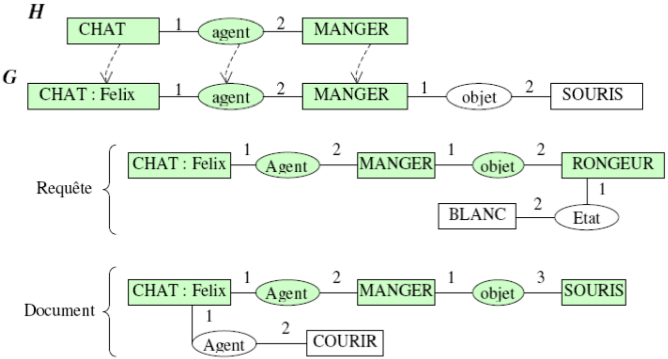


Rat → Agnt → Read → Obj → Book

Concept construction



Concept Matching



Conclusion

Common step for indexing:

- Define descriptor set
- Automatize descriptor extraction from documents
- Select a model for index representation and weighting
- Define a matching process and an associated ranking

NLP in IR: developments

Research in IR using NLP had not often show a strong positive effect.

- Useful in small domain like Medical
- Depend on resource quality
- Very costly for a very small positive impact

The use of semantic ressources, split in several directions:

- Semantic Web: explicit ressources, use of Logic, deduction, but outside of IR search engine
- Construction of lexical / terminological ressources: to expend query or documents, for small query / documents
- Word Embedding: exploit very large textual ressources, possible now, only statistical