Automatic extraction of characteristic properties of a concept

Etienne Picard

research environment

PhD Contract with France Telecom (09/2005 to 09/2008)

Industrial supervising : France Telecom

- Knowledge Sciences pôle, Knowledge Structuring axis
- ADN team (Natural Dialogue Agent)

supervisor : Florence Duclaye

Academic supervising : Joseph Fourier university (Grenoble)

- Laboratoire d'Informatique de Grenoble (LIG)
- HADAS team

supervisor : Marie-Christine Rousset

NII International Internship Program from 02/26 to 05/18

supervisor : Akiko Aizawa

research environment

summary

1 Research project
- Context
- Goal
- Approach

2 System developed
- Instance extraction
- Instance clustering

Conclusion

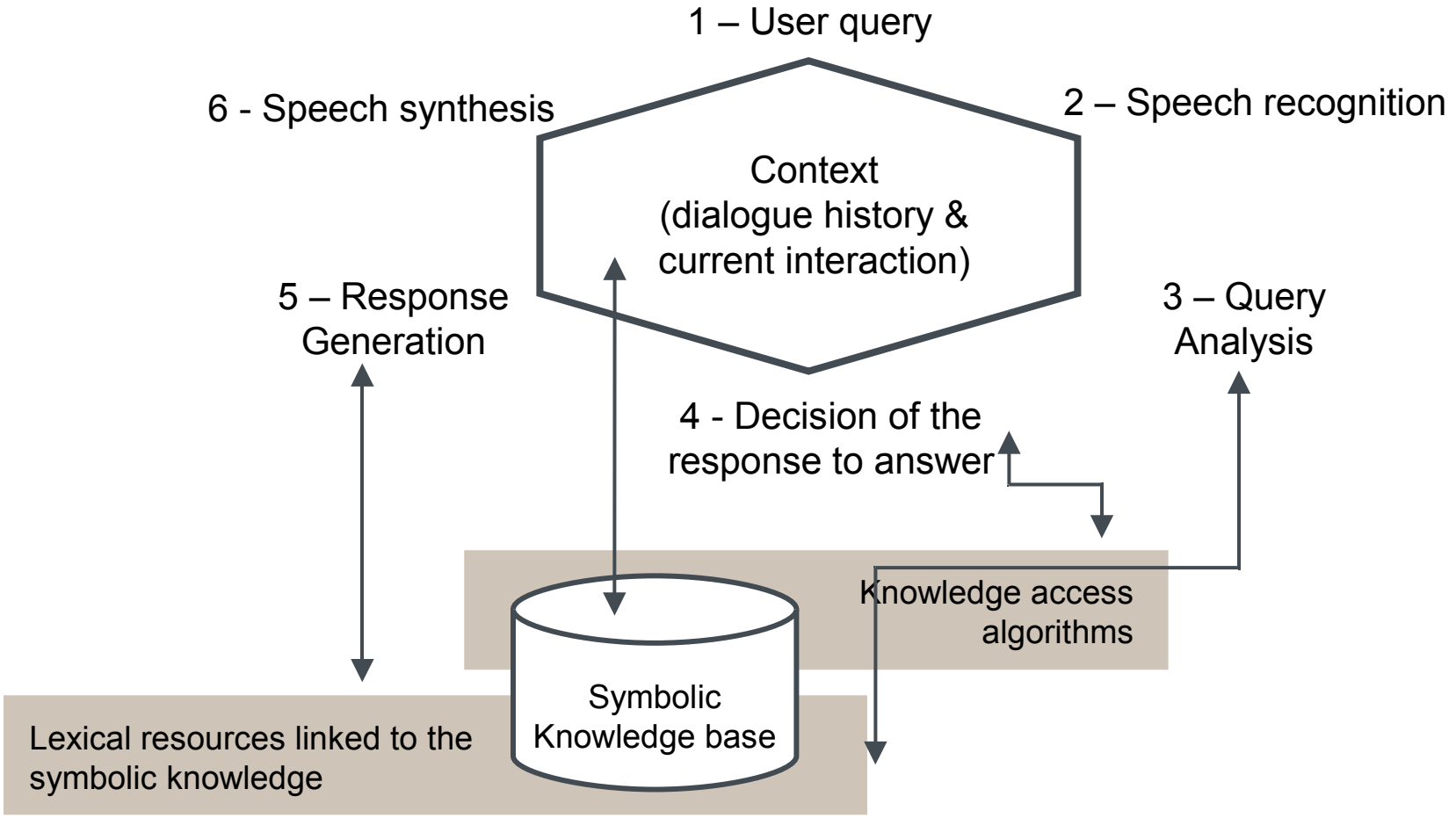
1 Research project

Context

Goal

Approach

information retrieval dialoguing agent



problem to solve

Current situation

- The manual creation of the knowledge bases is long and expensive
- In the case of dialogue agents, the knowledge bases are designed specifically for each application (hardly reusable because non consistent).

Goal

- Automate the creation of reusable semantic resources and store them into a library
- Use the web as a data source to create such resources

proposition

We introduce the concept of a "star" of characteristic properties :

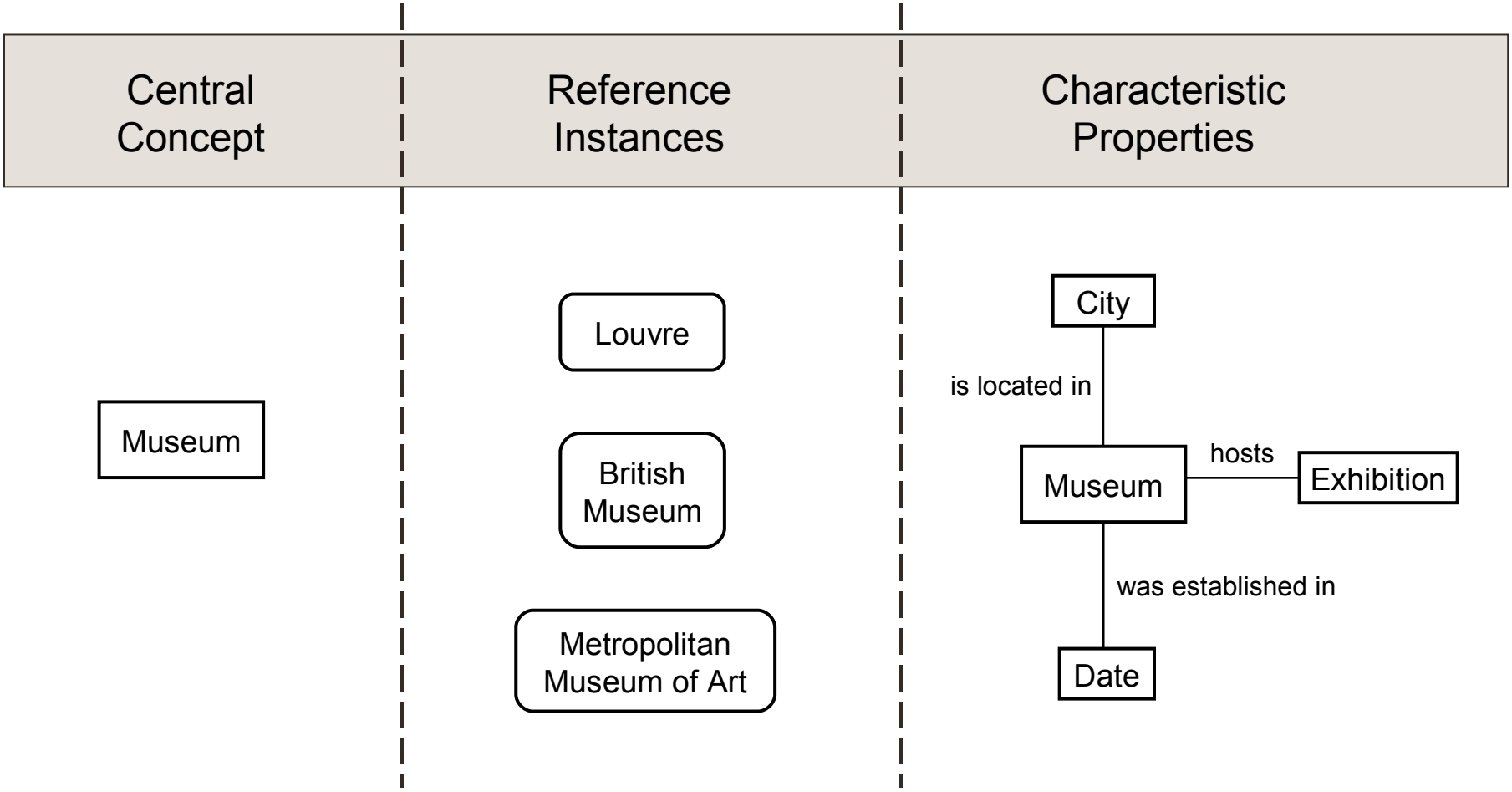
→ The Star Description of a Concept

From a set of instances of a concept, learn both the structure and the content of the star

We choose instances among famous named entities (of the type person, place or organization) for which many data are available on the web

- Ex : Singer, Actor, Museum, International Organization…

example


Central Concept
Reference Instances
Characteristic Properties
Museum
Louvre
British Museum
Metropolitan Museum of Art
City
is located in
Museum
hosts
Exhibition
was established in
Date

definition* : characteristic properties

The characteristic properties of a class are defined as properties which are used to state restrictions on this class

- The property is defined for all the instances of the class (i.e. the domain of the property is the class)
- The property doesn't have the same value for all the instances (i.e. the range of the property is composed of several instances)

* : RDFS notations are used in this definition

characteristic properties : example

The properties *is located in* and *hosts* are both characteristic properties :

- they are defined for any *Museum*
- their ranges (City and Exhibition) contain more than one instance

Ex : *Louvre – is located in – Paris*

British Museum – is located in – London

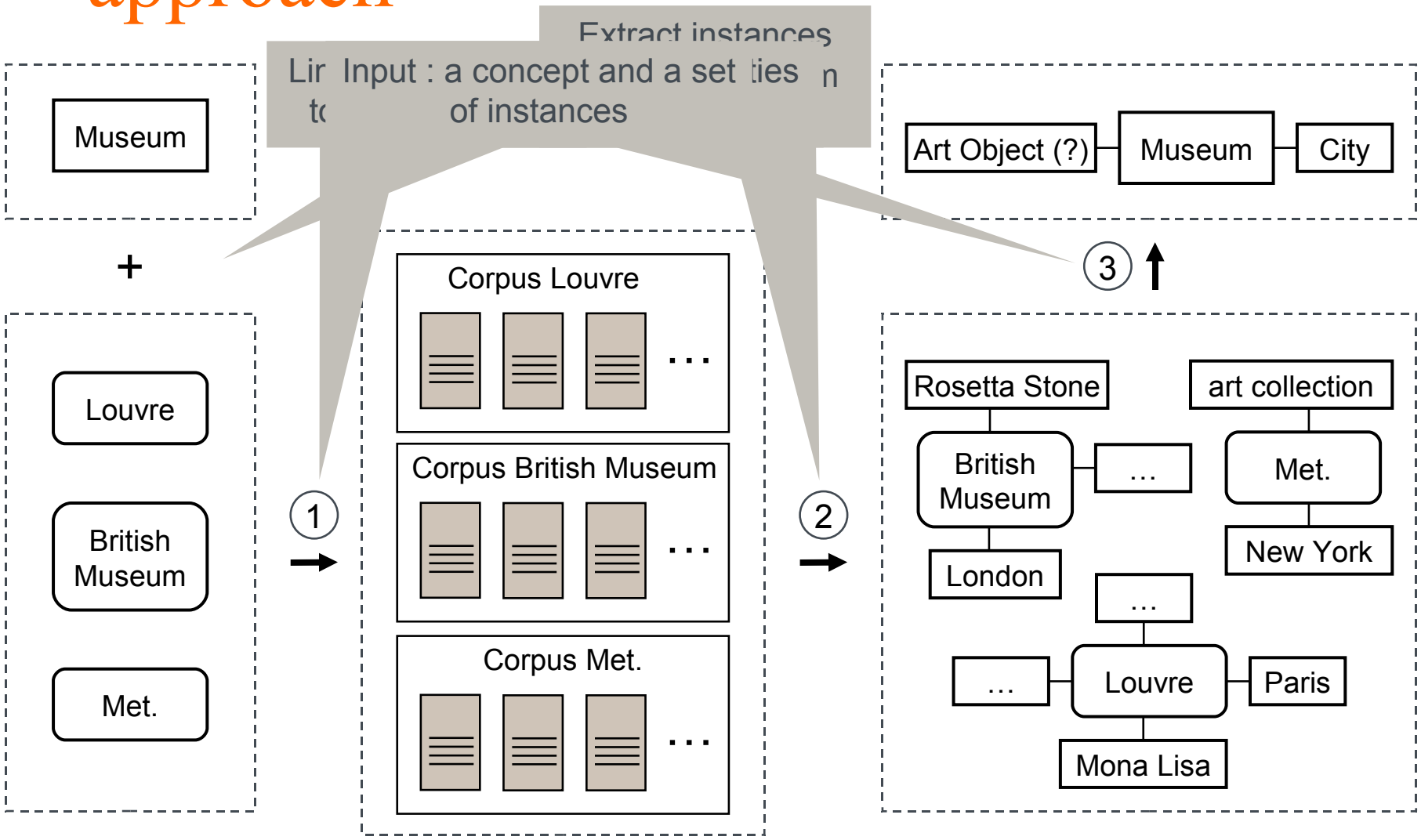
Centre Pompidou – is located in – Paris

Louvre – hosts – the Mona Lisa

British Museum – hosts – the Rosetta Stone

Louvre – hosts – the Venus de Milo

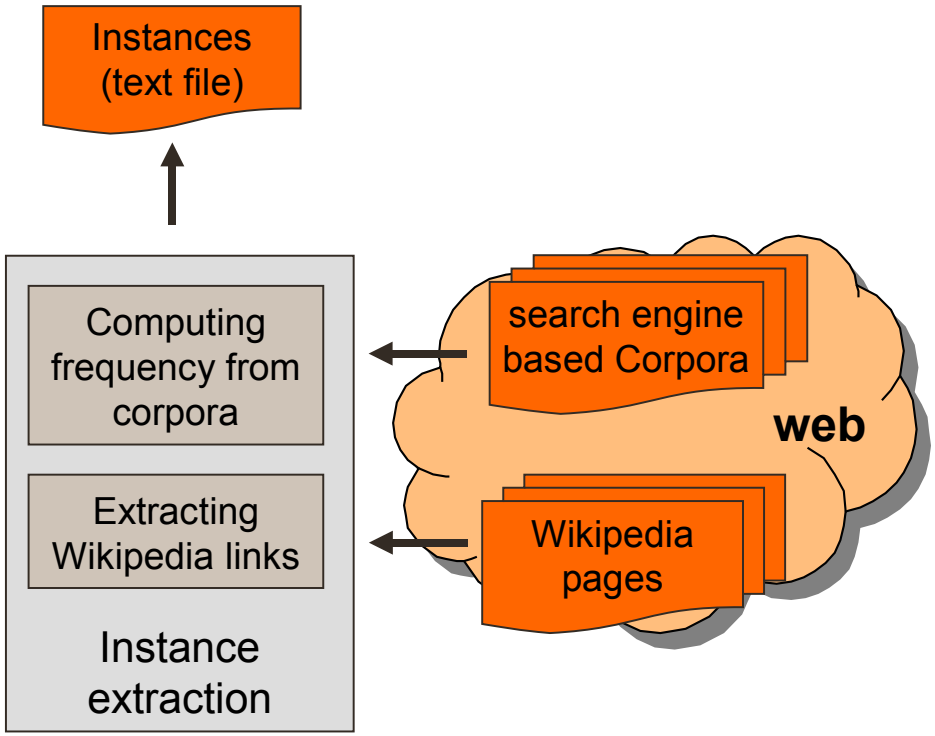
approach


Extract instances
Input : a concept and a set of instances
Museum
+
Louvre
British Museum
Met.
1
Corpus Louvre
Corpus British Museum
Corpus Met.
2
Art Object (?)
Museum
City
3
Rosetta Stone
art collection
British Museum
…
Met.
London
New York
…
…
Louvre
Paris
Mona Lisa

2 System developed

- Instance extraction
- Instance clustering

system overview



search engine based corpora

We build our corpora by sampling the Web using search engines :

- for a concept and an instance, we send a query of the form
  <concept> + *<instance>*
  Ex. : query = Museum + Louvre, Museum+ Prado…etc.

- for each result given by the search engine we extract the text in natural language of the corresponding web page.

Experimentation parameters :

- Search Engine used : Yahoo
- Number of pages extracted : 200
- Concepts addressed : Museum (Louvre, British Museum, Metropolitan Museum of Art, Prado), Singer (Bob Dylan, Madonna, Michael Jackson)

entity extraction

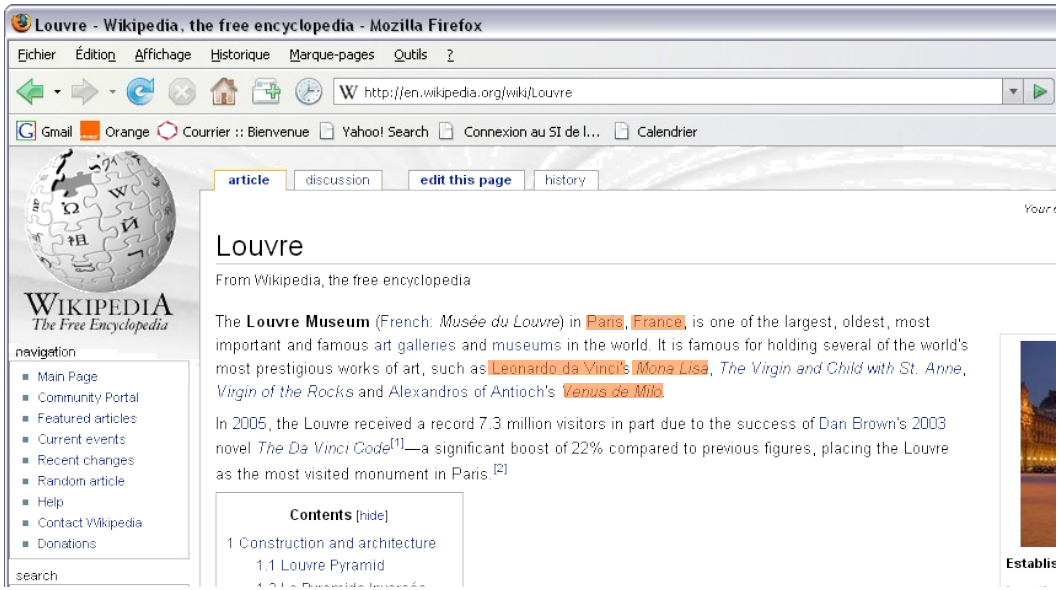
We use Wikipedia to find words likely to be entities :

- In the Wikipedia page related to our reference instance (ex. the Wikipedia page for the Louvre), we collect all words which are a link to other Wikipedia pages
- We calculate in the corpus related to this instance the frequency of each of these words
- The most frequent words are considered as words likely to be names of instances

entity extraction

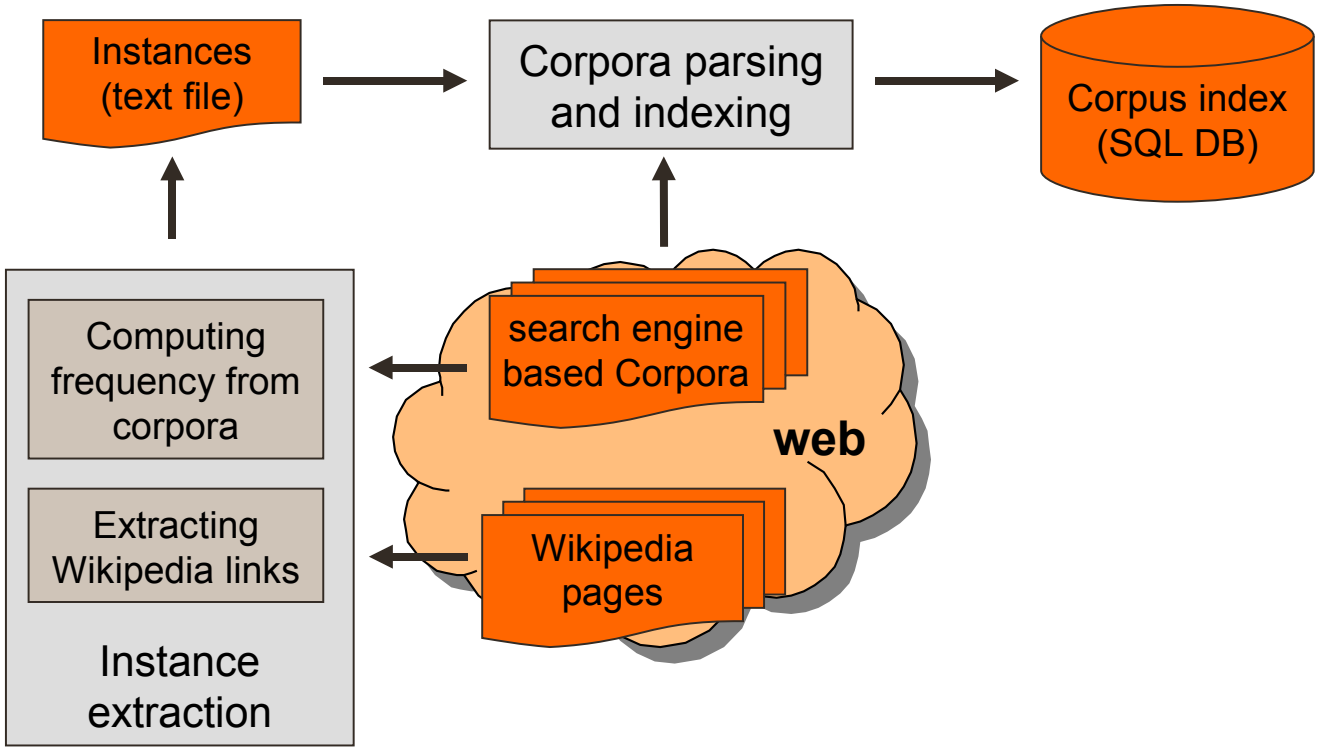
Extraction of wiki links

The frequencies are calculated in a corpus related to the Louvre museum built with the 200 first pages returned by Yahoo.



Terms	Frequency
Paris	419
France	201
Napoléon	127
mona lisa	126
new york	55
leonardo da vinci	44
louis XIV	34
venus de milo	33
napoleon III	33
Europe	33

system overview



corpora parsing and indexing

Using Stanford Parser*

Possibility of collapsing parsed dependencies

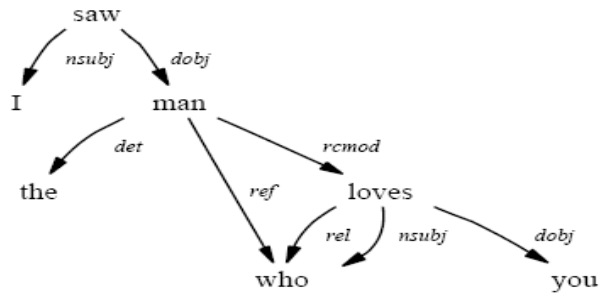


Figure 3: An example of a typed dependency parse for the sentence "I saw the man who loves you".

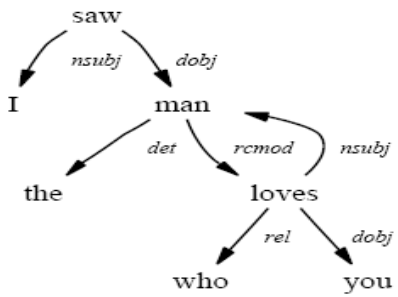
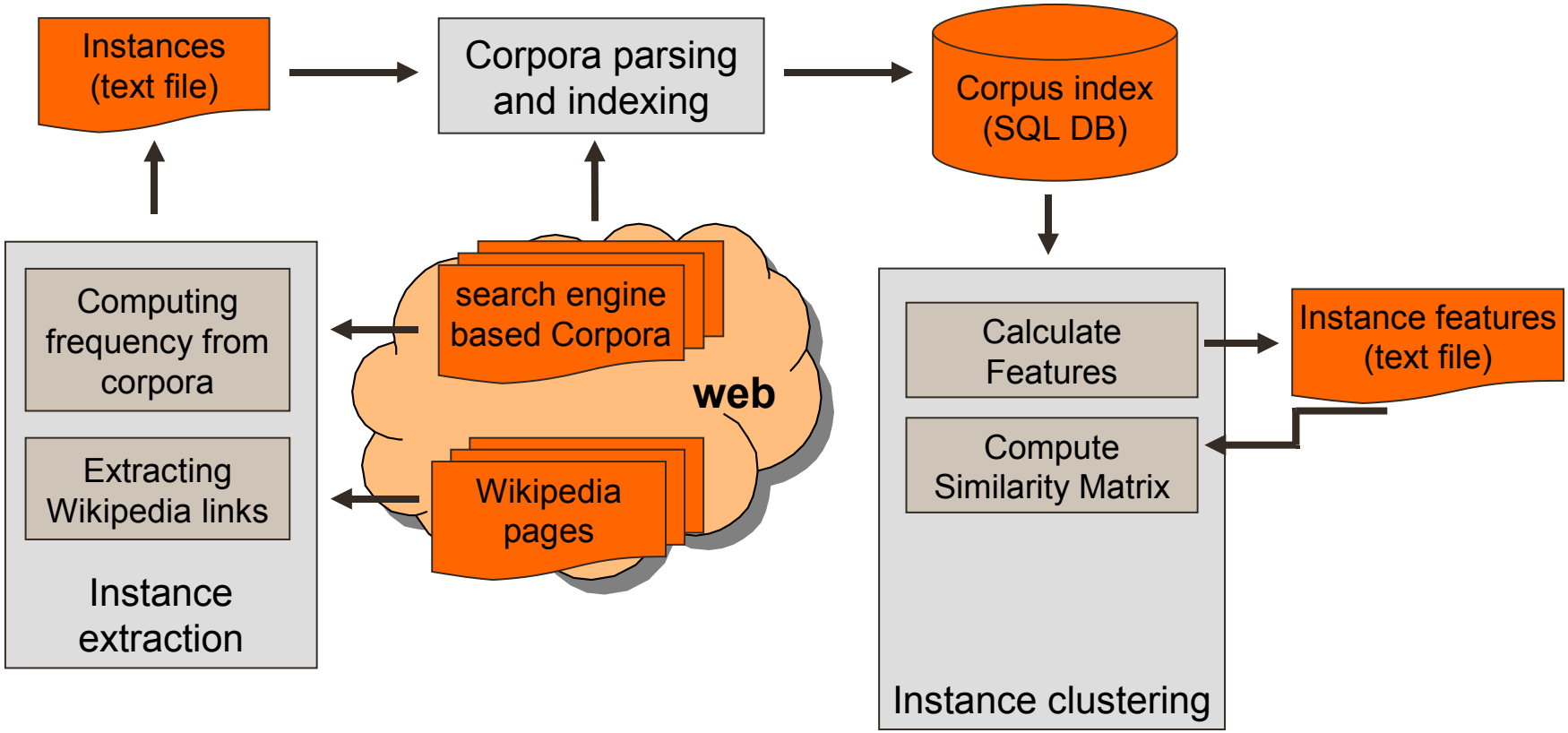


Figure 4: An example of a typed dependency parse for the sentence "I saw the man who loves you", with "collapsing" turned on.

Storing results in MySQL database

* : http://nlp.stanford.edu/software/lex-parser.shtml

system overview



calculate entity features

- We represent each word by a feature vector
- Each feature corresponds to a context in which the word occurs
- The value of the feature is the pointwise mutual information between the feature and the word.

mutual information : $$mi_{w,c} = \log \frac{\frac{F_c(w)}{N}}{\frac{\sum_i F_i(w)}{N} \times \frac{\sum_j F_c(j)}{N}}$$

w is a word and c is a context

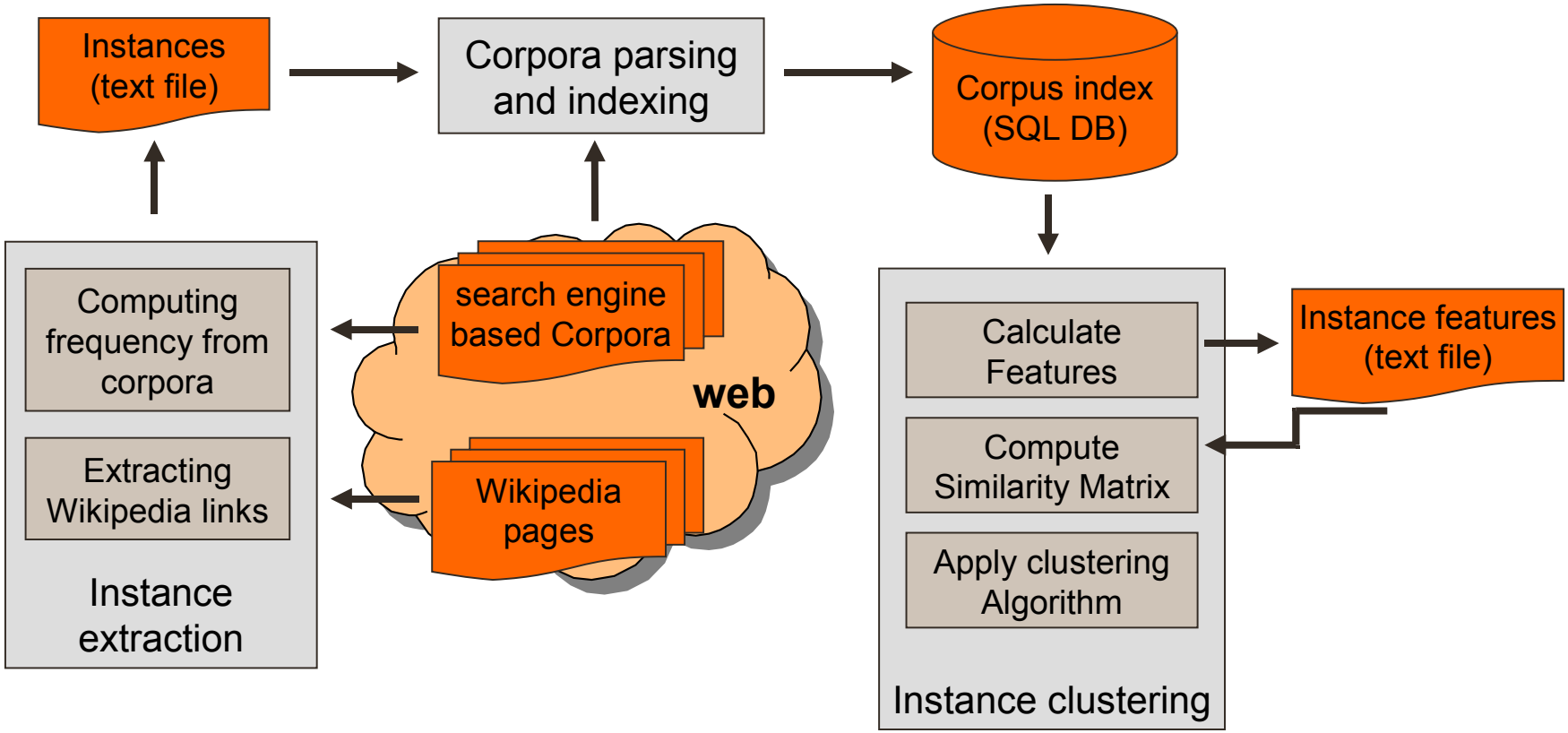
N is the total frequency count of all words and their context ($\sum_i \sum_j F_i(j)$)

compute similarity matrix

We compute the similarity matrix, by calculating the cosine similarity between the features of each pairs of instances.

Cosine similarity : $$sim(w_i, w_j) = \frac{\sum_c mi_{w_i c} \times mi_{w_j c}}{\sqrt{\sum_c mi_{w_i c}^2 \times \sum_c mi_{w_j c}^2}}$$

system overview



clustering algorithm

For each entity

- Choose top 10 most similar entities
- Perform hierarchical clustering
- Store best scoring cluster

For all stored cluster

- Compute cluster overlap, for each pair of stored corpus
- Identify similar clusters (cluster overlap + threshold value)
- Discard lowest scoring cluster

(cluster score : $score(c) = |c| \times avgsim(c)$, where $avgsim(c)$ is the average pairwise similarity between elements in c.)

Reference

P. Pantel & D. Lin. Discovering Word Senses from Text. In (KDD-02).

clustering algorithm

Cluster	Score
European, American, Italian, French, Chinese, Islamic, Asian, Greek, Egyptian, Roman, Persian	1.1519537
Titian, Rembrandt, Raphael, Botticelli, Goya, el-greco	0.825086
parthenon-marbles, elgin-marbles, rosetta-stone, Guernica, temple-of-dendur, reading-room, mona-lisa, winged-victory-of-samothrace, venus-de-milo, las-meninas	0.48822415
central-park, Manhattan, France, Paris, Spain, Madrid, London, new-york, united-states, Europe	0.43967333
charles-III, louis-XIV	0.36391425
Greece, Egypt, Rome	0.27002823
metropolitan-museum, british-museum, louvre-museum, prado-museum	0.26234153
act-of-parliament, hans-sloane	0.14359762

clustering algorithm

Work in progress…

Results

- Good results with the "museum corpus", by using as features Stanford Parser collapsed relations
- No convincing results with the "singer corpus"

Next step

- Try to find a subset a features (syntactic dependencies) that gives good results for any corpora…

conclusion

Goal

- We are trying to build Star Descriptions of Concepts by mining different types of resources from the web.

Approach

- Implement a set a techniques for extracting and linking words
- Filter the results by crossing the results obtained with different techniques

conclusion

Next Steps

- Find new techniques for entity (words likely to be instances) extraction
- Find techniques for relation (statements) extraction
- Conduct experiments on more concepts and instances

Thank you very much for attending this presentation…

Appendix

using a web search engine as a statistic resource

In order to filter the extracted words (both names of concepts and instances), we try to find the hyponymy relations between words likely to be names of classes and words likely to be names of instances.

The technique we use relies on Hearst patterns and Search engine counts* :

- Instantiate for each concept-instance pair, a list of lexico-syntactic patterns (Hearst patterns)
- Submit each instantiated pattern as a request to a web search engine and collect the number pages found by the engine
- For each concept-instance pair calculate a score equal to the sum of the number of pages found, for this pair, with each pattern
- For each instance keep as a hypernym the concept with the best score

** ref. : P. Cimiano, S. Staab, Learning by Googling, 2004, SIGKDD Explorations*

using a web search engine as a statistic resource

Experiment realized with a list of concepts containing the 20 most frequent words in the previously presented corpora :

part, year, gallery, information, world, site, collection, city, time, work, history, day, room, artist, building, century, painting, exhibition, art, sculpture

Terms	Concept	Score
Paris	city	105457
France	part	18369
Napoléon	day	179
mona lisa	painting	593
new york	city	448864
leonardo da vinci	artist	1676
louis XIV	time	21
venus de milo	sculpture	23
napoleon III	collection	2
Europe	part	135662