Learning Verbal Relations in Text Maps

Isabelle Debourges, Sylvie Guilloré-Billot, Christel Vrain

LIFO – Rue Léonard de Vinci – 45067 Orléans cedex 2 – France – email: {debourge, billot, christel.vrain}@lifo.univ-orleans.fr – tel : 33 (0)2 38 41 72 98

Abstract

We are interested in Text Mapping that we have defined as learning a partial representation of a domain by means of maps, given a corpus and a user request. In this paper, we mainly focus on learning relations linking the concepts occurring in maps. We concentrate on relations expressed in the corpus by means of verbs: we first learn the verbs frequently occurring between two concepts, and then we aim at refining this information by orientating the relation and/or annotating it with adverbs or prepositions. Let us precise that we define two constraints for developing our method: the first one was to avoid sophisticated tools for preprocessing the texts they are often time consuming and the second one was efficiency. After presenting the algorithms we give results we have obtained on five English and French corpora.

Résumé

Nous nous intéressons à la cartographie de textes que nous avons définie comme l'apprentissage d'une représentation partielle, à l'aide de cartes, d'un domaine à partir d'un corpus et d'une requête utilisateur. Dans ce papier, nous nous intéressons plus particulièrement à l'apprentissage de relations verbales : nous apprenons les verbes apparaissant fréquemment entre deux concepts et nous cherchons à raffiner cette information en l'orientant et/ou en l'annotant par des adverbes ou des prépositions. Ce papier présente les algorithmes que nous proposons et leurs fondements. Il présente également les résultats obtenus sur cinq corpus de langues anglaise et française.

Keywords: Text Mapping, semantic and lexical relations, filtering information, acquisition of domain knowledge.

1. Introduction

Many texts are now available in their electronic version and they provide a very large knowledge source that can be now quite easily built by using Information Retrieval tools. Nevertheless, exploiting such a knowledge source is very difficult because of the number of texts, their diversity and their large sizes. We aim at developing a tool that given a request expressed by a user and a corpus linked to that request, helps the user to build a representation of the information contained in the corpus and linked to his request.

To reach that goal, we propose a knowledge acquisition system that provides a partial model of the domain. It filters the information available in the texts. The result is given as a text map (or conceptual map) that depends on the initial set of keywords and on the request evolution.

A conceptual map is a view of the content of the text at a meta-level: it does not represent the content of each sentence, but the main concepts occurring in the texts in the context of the user request, and their links. Conceptual maps can be connected to semantic networks (Quillian, 1968) and conceptual graphs (Sowa, 1984) because of their graphical representation.

Other works have been done in that direction in the last few years and that aims at representing the main information contained in a text into maps. Let us mention for instance WordMapper (www.wordmapper.com), Leximancer (www.leximancer.com) which are recent commercial products. They are based on statistical algorithms and only give the concepts related to the one given by the user.

But as far as we know they do not propose semantic relations between the concepts. Moreover, when the user gets the concepts related to his initial request, he may be interested to study more deeply parts of the texts containing them to determine whether some new concepts emerge in that restricted context. These two points are partly addressed in our system: the process for learning concepts is iterated taking into account the answers of previous iterations, and as explained in Section 3, links between concepts are, when possible, labeled by verbs.

This paper is organized as follows. Section 2 is devoted to the presentation of conceptual maps, and the method we propose for building such maps is briefly presented. In Section 3, we focus on learning relations in the maps in order to give a label to the edges, to orientate the edges when possible, to refine the labels. Results obtained on five corpora (four in English and one in French) are given in Section 4. Perspectives for that work will be proposed in Section 5.

2. Conceptual Maps

2.1. Definition

Given a keyword and a corpus, a *conceptual map* presents a set of concepts strongly linked to that keyword, in that corpus. It is defined as a graph. One of the node is labelled by the keyword and the other ones are labelled by words representing concepts. Moreover, there exists edges between the node representing the keyword and each other node. The edges can be labelled by relations. This generic representation is illustrated in Figure 1:

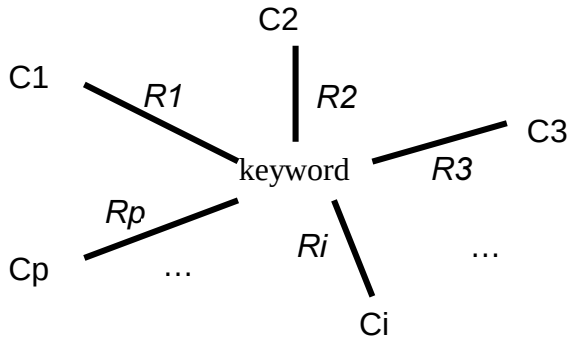


Figure 1: A generic representation of a conceptual map

The system that we have developed aims at building such maps, from a corpus and an initial request (expressed by one or several keywords). The idea underlying such maps aims at giving a representation of the corpus focused on the request of the user. He can ask the system to refine a map on a concept appearing in it. A new map is then built centered on that new concept, but considering only a part of the corpus taking into account the previous request. So, the previous requests are influent on the map obtained.

Our text maps extraction system is composed of three steps: text preprocessing (the text is cleaned-up, syntactically tagged and lemmatized), concepts extraction, and relations

extraction. The first step is processed once, the two other ones are processed for each map. The concepts linked to the keyword are first extracted. Then for each couple (keyword, concept_i) relations are searched for.

2.2. The Concepts Extraction

The concepts extraction algorithm we propose is based on an iterative algorithm. At the first iteration it selects the set S_1 of non empty words appearing the more frequently in the same sentences as the keyword. In the following iterations, it selects the set S_i of non empty words appearing the most frequently in the same sentences as words of S_{i-1}. This process stops when a fixed point is reached (Debourges and al, 2001): the set of words extracted is the set of emerged concepts.

This algorithm is original because of its iterations. The propagation of the selection allows to attain words that are semantically close to the keyword although not directly connected in sentences.

Here are some maps (restricted to the concepts) processed on corpora presented in Section 4.

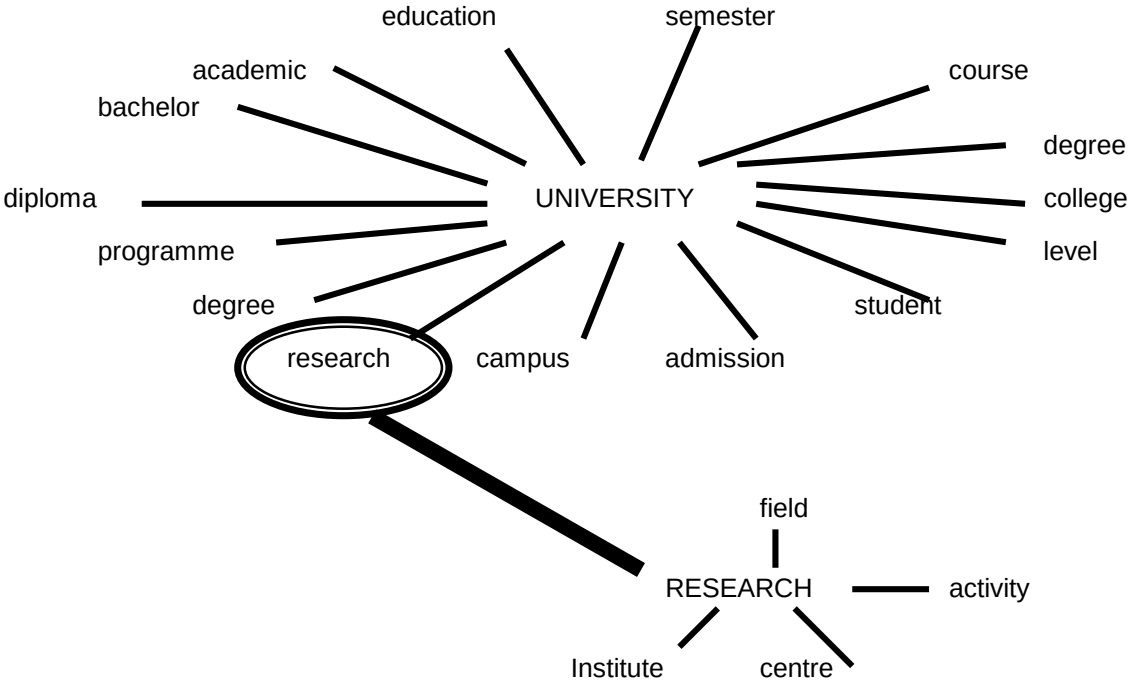


Figure 2. Maps obtained on the universities corpus with the initial keyword universit*y and a refinement on* research

The maps presented in Figure 2 have been obtained by a request on the keyword *university* and a refinement on the emerged concept *research*. On the other hand, the map presented in Figure 3 has been obtained with the initial keyword *research*. Those maps show that the emerged concepts depend on the evolution of the request.

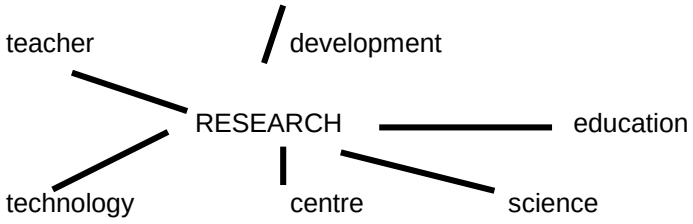


Figure 3. A map obtained on the universities corpus with the keyword research

The map presented in Figure 4 has been processed on the ***Little Prince*** corpus. It presents the main "actors" of the novel.

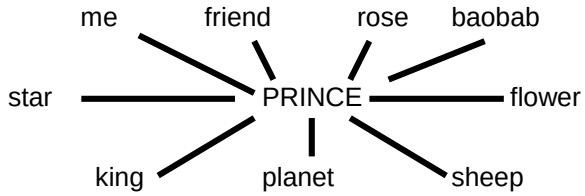


Figure 4. A map obtained on the Little Prince corpus with the keyword prince

3. Extracting the relations

As already mentioned, the third step of the algorithm consists in learning labels on the link between the keyword and a related concept, so that to capture partially the semantic relations between them.

The extraction of relations is a difficult task (Morin, 1999; Le Priol, 2000). Most of the works in that field are related to the detection of hyponymy / hyperonymy relations between concepts. The relations we are looking for must express the way concepts are related in the corpus: it may be hyponymy / hyperonymy relations, but it may be any information obtained in the corpus and connecting the words. The relation we search for is a semantic one, but we use linguistic indices to learn it. For the time being, we focus on verbs. Actually, when the extraction of concepts is processed on a corpus whose sentences have the classical form

"*subject + verb + complement* ", we have noticed that most of the extracted concepts are nouns, and therefore if we wish to attribute a label to the relation existing between an emerged concept and the keyword (classically chosen as a noun by the user), it looks relevant to extract verbs. Moreover, in such cases the syntactic functions give more precise information about the relation. This is why this task is divided into three steps: the first one learns verbs, the two other one refine the relation by studying syntactic functions to determine the direction and eventually to add prepositions or adverbs.

3.1. Verb Relations

The algorithm we propose to extract verb relations looks for the verbs which co-occur the most frequently with the keyword and the concept.

The preprocessing step labels words using TreeTagger. Verbs are then identified by the label they receive (verb + active/passive voice). Their lemmatization allows to count altogether all

the derivations of the same seed. The verbs which appear the most frequently in the sentences containing simultaneously the concept and the keyword are extracted as potential verb relations. For example, we can extract the relations:

student —— attend ——— university

category —— organize —— synonyms

prince ——— eat ———— sheep

prince ——— manger ——— mouton

The information given by the verb alone is often not expressive enough. We would like to learn the direction of the action. Is it

prince ——— eat ——→ sheep ?

(*The little prince eats the sheep)*

or sheep ←—— eat ———— prince ?

(*The sheep eats the little prince)*

This explains why next step aims at indicating the direction of the verb relation, in order to detect when possible the object of the action.

3.2. Orientation of the verb relation

The method that we have developed is based on a heuristic, stating that usually the actor of a verbal relation is located before the verb in the active voice and after it in the passive voice. Therefore, we count the number of cases these situations occur for both the keyword and the concept. The algorithm is thus composed of two main steps:

- sentence segments extraction
- study of the sentence segments taking into account two criteria: the active or the passive voice, the relative positions of the verb and the concept/keyword in order to define the predominant role of each of them.

3.2.1. Sentence segments extraction

In the following, C1 and C2 are respectively the concept and the keyword, and V is the verb of the relation. The segments extraction algorithm is as follows:

For each sentence S containing simultaneously C1, C2 and V do

> For each instance of V in S do
>
> Point at the nearest Ci on the left of V and extract the corresponding segment [Ci … V]
>
> Point at the nearest Ci on the right of V and extract the corresponding segment [V … Ci]

Example:

Let us suppose that the user gave the keyword *fonction*. One of the concepts he obtained is *valeur*. If we are looking for details on the relation *rendre* between *fonction* and *valeur*, then the following original sentence is selected:

Si la dernière instruction exécutée par une fonction n'est pas une

instruction return, la valeur rendue par la fonction est indéterminée.

The two extracted segments are:

valeur rendue

rendue par la fonction

Let us notice that a sentence S is selected if and only if the verb V occurs at least once in it. In the worst case (only one occurrence of V in S and C_i C_j V or V C_i C_j), only one sentence segment will be extracted from S. The number of segments is therefore at least equal to the number of sentences that simultaneously contain V, C1 and C2.

3.2.2. Discrimination

The sentences segments that are obtained are divided into four groups:

C1_actor	=	[C1 $V_{(\text{active voice})}$]	∪	[V $C1_{(\text{passive voice})}$]
C1_object	=	[C1 $V_{(\text{passive voice})}$]	∪	[V $C1_{(\text{active voice})}$]
C2_actor	=	[C2 $V_{(\text{active voice})}$]	∪	[V $C2_{(\text{passive voice})}$]
C2_object	=	[C2 $V_{(\text{passive voice})}$]	∪	[V $C2_{(\text{active voice})}$]

Then afterwards, we introduce a coefficient n to discriminate the roles and to insure the predominance of one of the roles. The value of n we currently use satisfies:$1 < n \leq 1.5$

If Ci_actor > n* Ci_object then Ci is more likely to be the actor

Else if Ci_object > n* Ci_actor then Ci is more likely to be the object

Else Ci has a fuzzy role.

Detection of conflict: if C1 and C2 have the same role, the result is a conflicting one.

If the roles of one of the concept is fuzzy, then its role is set to the complementary of the role of the other concept (only if this does not lead to a contradiction to the most important number of cases listed).

If the role of each concept is clearly defined, then the relation will be represented as:

actor ——— verb ——→ object

or object ←——— verb ——— actor.

Example:

Let us consider for instance the verb *rendre* and the two concepts *fonction* and *valeur* : 43 segments are extracted from the corpus; they are split up into the 4 categories:

fonction_actor: 22 segments	valeur_actor : 1 segment
fonction_object: 0 segments	valeur_object : 20 segments

Then, with the verb *rendre*, *fonction* is more likely to be the actor and *valeur* is more likely to be the object. The result is represented by:

fonction ——— rendre ———→ valeur

The heuristic we propose for the extraction of segments is simple but this allows to have a quite efficient method, and this was one of our initial requirements. The results can be disrupted by noise. The system will be less sensitive to noise when the number of segments

extracted and used in our process is high enough. Nevertheless, this is not always satisfied. For example, the relation

prince ⟵ love ——— sheep

emerges only once, in the English version of ***The Little Prince.*** The original sentence is:

For you who also love the little prince, and for me, nothing in the universe

can be the same if somewhere, we do not know where, a sheep

that we never saw has – yes or no? – eaten a rose…

Then the extracted segment is: *love the little prince*

As the verb *love* is at the active voice, this leads to Prince_object : 1 segment. As this is the single sentence containing simultaneously prince, love and sheep in the whole corpus, the conclusion should be that : *prince is the object, and sheep is the actor*. But that is not correct, and in fact, the sheep does not express any feeling in the book. To avoid this problem, we provide a minimum threshold on the number of segments for inducing the relevance of a value.

3.3. Empty words information

The relations can also be annotated using prepositions and/or adverbs. This allows the user to know the way verbs are used in the sentences: he can either learn the way that verb is used in that specific context, or get a finer information on the content of the text.

Those prepositions and adverbs are selected from a subset of the extracted segments: the ones in which the concept holds the role that was attributed to it in section 3.2.

For example, on the *Langage C* corpus, we can get:

paramètre —EN→ passer —PAR→ valeur

(passer une valeur EN paramètre)

(passer un paramètre PAR valeur)

This relation is quite complete. It contains three types of information: the label *(passer)*, the direction and the prepositions (*en, par*).

4. Results on five corpora

The results we are presenting here have been processed by our system (which implements all the proposed algorithms), on five corpora:

▪ The English version of ***The Little Prince*** and its original version ***Le petit Prince*** (Antoine de Saint Exupéry). This novel contains 17, 285 words in English and 13, 614 words in French.

▪ The ***Universities Corpus*** describes the conditions of admission in about 950 world-wide universities. Each text is one to two pages long. This domain specific corpus contains 385, 250 words.

▪ The ***Five Papers*** describing the WordNet ontology. It is a technical corpus that contains 40, 233 words.

▪ ***Introduction au Language C*** is a book written by Bernard Cassagne working at the Strasbourg University (France). It is free, available at http://www-clips.imag.fr/commun/bernard.cassagne/Introduction_ANSI_C.html , and contains 34, 005 words.

4.1. "The Little Prince" and "Le Petit Prince"

Let us show some relations extracted from the English and French corpora. Most of them are identical but there exists some differences.

prince ——— raise ⟶ rose *(English version)*

prince ——— cultiver ⟶ rose *(French version)*

Those relations (*raise* and *cultiver*) have the same meaning and the same direction in the two corpora, and they emerge with the same score.

prince ——— have ⟶ sheep *(English version)*

prince ——— eat ⟶ sheep *(English version)*

prince ⟵ love ——— sheep *(English version)*

The relations *have (avoir)* and *eat (manger)* between prince and sheep *(mouton)* are well oriented.

prince ——— avoir ⟶ mouton *(French version)*

prince ——— manger ⟶ mouton *(French version)*

prince ——— vouloir ⟶ mouton *(French version)*

The relations appearing in the French corpus look better, and well oriented. Two of them are the direct translation of *have* and *eat* we obtained in the English corpus, but the third (*love* and *vouloir*) do not have the same meaning, even if they have exactly the same score.

4.2. The Universities

Here are relations extracted on the Universities corpus, between the keyword *university* and some concepts emerging with that keyword. Those relations where chosen because of their high quality: their directions were decided on a large number of segments and there was no ambiguity on the roles of each element.

university ——— follow ⟶ semester

university ——— offer ⟶ level

university ——— provide ⟶ level

university ⟵ establish ——— level

university ——— offer ⟶ diploma

university ——— provide ⟶ diploma

university ——— promote ⟶ research

university ——— include ——→ research
university ——— undertake ——→ research
university ——— provide ——→ research
university ——— develop ——→ research

university —IN— register ——→ student
(a student is registered in a university)

Those relations and their directions are pertinent and give extra information to the user and allow him to know more about the domain.

But very few prepositions are extracted in that corpus. Perhaps because English language natives use those little words less often ?

This corpus contains some examples of conflicts:

university ——— offer ——— campus
university ——— provide ——— campus

The orientation of those relations (*provide* and *offer*) are fuzzy. The scores are:

	Offer	*provide*
university is actor	**15** segments	**5** segments
university is object	**12** segments	**5** segments
campus is actor	**11** segments	**3** segments
campus is object	**9** segments	**5** segments
	Σ = 47 segments	*Σ = 18 segments*

The ratios between the scores are too small to allow to take a decision on the roles of university and campus. But is it absolutely necessary to give a direction to them? Are not those relations pertinent in the two directions?

There is a similar problem with the three relations

research ←——— include ——→ development

and

research ←——— learn ——→ science
research ←——— include ——→ science

whose scores are

	include
research is actor	4 segments
research is object	7 segments
development is actor	1 segments

development is object	4 Segments
	Σ = 16 segments

Most of the ratios between the scores are higher than in the previous example, but another problem appears: the two concepts are susceptible to be alternately actor (*research* and *development* with *include*) or susceptible to be alternately object (*research* and *science* with *learn* and *include*).

	Learn	*include*
research is actor	1 segments	1 segments
research is object	1 segments	3 segments
science is actor	1 segments	1 segments
science is object	2 segments	5 segments
	Σ = 5 segments	*Σ = 10 segments*

An expertise of the relations can conclude on the pertinence in the two directions.

We shall mention that in those cases, the number of segments extracted is not very important in spite of the size of the corpus. Moreover, the more the number of segments is large, the more the results are pertinent.

4.3. Five Papers on WordNet

The corpus composed of the Five Papers describing WordNet allows us to extract relations as:

category ——— arrange ⟶ synonym

category ——— organize ⟶ synonym

synset ——— contain ⟶ noun

set ——— organize ⟶ synonym

Those relations are relevant and very representative of the domain it is processed on. They show that text maps applied to specific domain give a partial representation of the domain knowledge.

The extraction of some hyponymy / hyperonymy relations is illustrated by:

synset ——— be ⟶ category

antonymy ——— be ⟶ relation

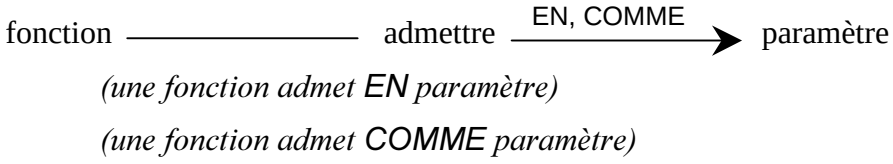
Of course the extraction of such relations depends on the fact that the corpus must explicitly contain the definition with the verb *be*. It could not emerge if that relation was not expressed with such a verb.

4.4. Introduction au Langage C

The maps processed on the *Introduction au Langage C* manual are another example of application of Text Maps to a specific domain. This book was written to help programmers to use the C language. The maps processed on that corpus will help the programmers to have an overview and find the information they are looking for.

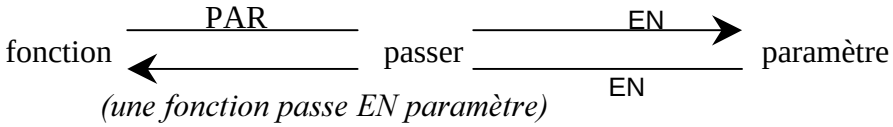
As the corpus is a French corpus, it contains lots of little words as prepositions and adverbs. Furthermore the way the ideas are formulated in that domain is quite particular and specific. So it is very important to show the user the way the sentences are constructed.

Let us have a look at some extracted relations:



(une fonction admet EN paramètre)

(une fonction admet COMME paramètre)



(une fonction passe EN paramètre)

(une fonction est passée EN paramètre)

(on passe PAR paramètre à une fonction)

In the relation *passer* between *fonction* and *paramètre* , the two roles are fuzzy. As prepositions are extracted and are not exactly the same in the two directions, the two cases are distinguished.

(une valeur est rendue PAR une fonction)

4.5. General remarks

▪ The prepositions were nearly exclusively extracted from the ***Introduction au Language C*** corpus. This corpus is written in French. Nevertheless, ***Le Petit Prince*** is also a French text and no preposition has emerged. Maybe can we suggest that French language uses more prepositions, but the most convincing hypothesis is that it depends on the way the corpus is written, and the nature of the verbs used in it.

▪ In accordance with the initial constraints we had decided, the execution times are short. The extraction of all the relations between university and level takes 2. 021 seconds on the whole corpus; it costs 0.512 seconds on the European texts. The orientation of (relation, be, antonymy) costs 0.129 second for 42 segments; the orientation of (university, offer, level) costs 0.379 second for 144 segments.

5. Perspectives

▪ One of the next evolutions of the system will concern the extraction of relations between concepts that are not necessarily nouns. Indeed we are working on a very specific corpus *"Flore de la Polynésie française"(Jacques Florence, éditions de l'ORSTOM)* whose sentences are not classically formed: most of the time, they are only composed of nouns and adjectives (no verb).

▪ Another evolution will be to increase the number of extracted segments, by selecting the synonyms and hyponyms of the concepts we are searching for the roles.

▪ To make the results more exploitable by the user, we will provide link from the concepts and the relations to the original text.

▪ As our initial objective was to help the users to find information among corpora, we are now beginning an evaluation of the feeling of users in front of such maps. At a first stage, a group of students and teachers will have to give their opinion on a pre-defined set of maps processed on the *Langage C* corpus. This evaluation will measure the precision and the recall of the relations and the concepts in the presented maps. At a second stage, as an interactive interface is being prepared, some users will have to test if Text Mapping allows them to save time.

6. Conclusion

We have presented in this paper a system that extracts concepts related to a user request in a corpus. We focus on learning relations between the user request and the learned concepts. Let us insist on the fact that these relations are learned from linguistic indices. In this work those indices are mainly verbs and the syntactic functions associated to them. Those relations contain the label, the direction, and the associated prepositions.

The system that has been developed in C uses only TreeTagger to lemmatize and to label the words of the corpus. More sophisticated tools could be used , but the system would be less portable and certainly less efficient: executions are immediate on the implemented prototype.

Conceptual Maps are complements to Information Retrieval and Information Extraction. They provide a new approach to large corpora for any user: specialist or novice of the corpus domain can discover the content of the texts according to his request.

Acknowledgement

We would like to thank Bernard Cassagne to have entrusted us the original ascii files of his book ***Introduction au Langage C***.

References

Debourges I. and al. (2001). *Cartographie de Textes, Une nouvelle approche pour l'exploitation sémantique des corpus homogènes de grande dimension*. Research Report Lifo 2001-01.

Kayser D. (1997). *La représentation des connaissances*. Dunod.

Lebart L. and Salem A. (1994). *Statistique Textuelle*. Dunod.

Le Priol F. (2000). *Extraction et capitalisation automatiques de connaissances à partir de documents textuels. SEEK-JAVA: identification et interprétation de relations entre concepts.* PhD Thesis.

Morin E. (1999). *Extraction de liens sémantiques entre termes à partir de corpus de textes techniques*. PhD thesis, Institut de Recherche en Informatique de Nantes.

Quillian M.R. (1968). Semantic Memory, in *Semantic Information Processing*. MIT Press.

Sowa J.F. (1991). Principles of semantic networks. *Exploration in the representation of knowledge*. Morgan Kaufmann.

Viprey J-M.(2000). Hypertexte de corpus littéraire: cartographie et statistique multidimensionnelle. *Proc. of JADT 2000 (5èmes Journées Internationales d'analyse statistique des données textuelles)*.