A Lexical Database and an Algorithm to Find Words from Definitions

Dominique Dutoit[1] and **Pierre Nugues**[2]

Abstract. This paper presents a system to find automatically words from a definition or a paraphrase. The system uses a lexical database of French words that is comparable in its size to WordNet and an algorithm that evaluates distances in the semantic graph between hypernyms and hyponyms of the words in the definition. The paper first outlines the structure of the lexical network on which the method is based. It then describes the algorithm. Finally, it concludes with examples of results we have obtained.

1 INTRODUCTION

Le mot juste – the right word – consists in finding a word, and sometimes the only one, that describes the most precisely an object, a concept, an action, a feeling, or an idea. It is one of the most delicate aspects of writing. Generations of students, writers, or apprentice authors have probably experienced this. Unfortunately we must, too often, content ourselves with approximations and circumlocutions.

The right word is also crucial to formulate accurately the elements of a problem and solve it. Naming a broken or defective part in a car or a bicycle is a challenge to any average driver when confronted with a mechanic. The right word is yet essential to find the part number in a database, order it, and have it replaced. This problem is even more acute when no human help is possible as for some e-commerce applications where access to information is completely automated.

When the adequate vocabulary escapes us, a common remedy is to employ a circumlocution, a description made of more general words. Examples of such circumlocutions are dictionary definitions that conform to the Aristotelian tradition as in *une personne qui vend des fleurs* (a person that sells flowers) to designate a *fleuriste* (florist) or *la petite roue dentée au centre d'une roue de vélo* (the small toothed wheel in the center of a bicycle wheel) for *pignon* (sprocket-wheel).

This kind of definitions consists of two parts. A first one relates the object, the idea in question, to a *genus* to which the object, the idea belong, here *personne* (person). Then, the second part specifies it with a *differentia specifica*, a property that makes the object particular, here *qui vend des fleurs* (who sells flowers). A florist can thus be described as a species within the genus *personne*, with the *differentia specifica* "qui vend des fleurs."

The description of florist corresponds closely to its definition in the French *Petit Robert* dictionary: *Personne qui fait le commerce des fleurs* (a person who trades in flowers). In the *Cambridge International Dictionary of English*, the definition is slightly more restrictive: *a person who works in a shop which sells cut flowers and plants for inside the house*. However, the correspondence between somebody's wording of a concept and the word definition in a dictionary is not always as straightforward.

2 THE LEXICAL DATABASE

2.1 The Integral Dictionary

The Integral Dictionary – TID – is a semantic network associated to a lexicon (see [4] and [5] for details). It is available mainly for French and it is currently being adapted to other languages notably English and German. Its size is comparable to that of major lexical networks available in English such as WordNet [7] or MindNet [12].

A subset of the Integral Dictionary forms the core of the French lexicon in the EuroWordNet database [14]. Although the structures of TID and WordNet do not map exactly, it was possible to derive TID data and convert them in a WordNet compatible structure. In addition, the TID structure is used in the European Balkanet project to merge wordnets for Balkan languages (Greek, Turkish, Bulgarian, Romanian, Czech, and Serbian) in a single database [9].

2.2 The Structure of The Integral Dictionary

The Integral Dictionary organizes words into a variety of concepts and uses semantic lexical functions. Concept definitions are based on the componential semantic theory (see [8] and [11]) and the lexical functions are inspired by the Meaning-Text theory (see [10]). Lexical functions and componential semantics can be accessed in the Integral Dictionary using a Java application programming interface (API).

2.2.1 A Graph of Concepts

Ontological concepts are the basic components of the Integral Dictionary where each concept is annotated by a gloss of a few words that describes its content. When a concept is entirely lexicalized, the gloss is reduced to one word. It then corresponds to a particular kind of relation between a concept and a word, which is annotated as *generic*. When a concept is only partially lexicalized, the relation linking a word to this concept is annotated as *specific*, the word does not describe the concept entirely, or *characteristic*, a sort of metonymy.

A starting $\backslash$ denotes a concept as in $\backslash Personne\ humaine$ (human being) or $\backslash Animal\ \grave{a}\ fourrure$ (fur animal). Each concept contains words or other concepts that share a part of a meaning. The graph of concepts forms a structure around which the words are organized.

Concepts are classified into categories. This paper describes only the two main ones: the classes and the themes. Classes form a hierarchy and are annotated with their part-of-speech such as $[\backslash N]$

[1] Memodata, 17, rue Dumont d'Urville, F-14000 Caen and CRISCO, Université de Caen, F-14032 Caen, France, e-mail: dutoit@info.unicaen.fr
[2] Lund University, LTH, Department of Computer science, Box 118, S-221 00 Lund, Sweden, e-mail: Pierre.Nugues@cs.lth.se

or $[\backslash V]$. Themes are concepts that can predicate the classes. They are denoted by a $[T]$. The words appear as terminal nodes in the hierarchical graph of concepts as shown in Figure 1 for the word *fleur* (*flower*). Relations annotate arcs between concepts – themes and classes – and between words and concepts. Major relations are $ToClass$ with the values $Generic$ (hypernymy) and $Specific$ (hyponymy), various forms of synonymy, and $ToTheme$.

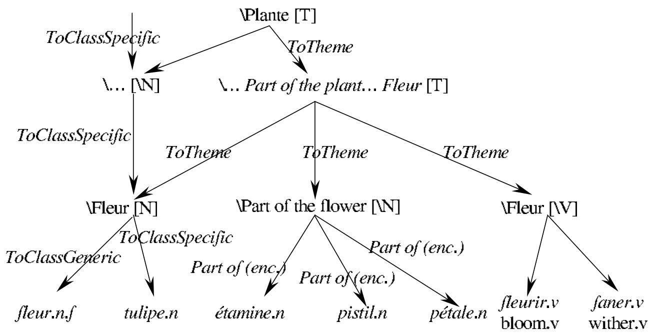

Figure 1. The graph of concepts for the word *fleur*.

The organization of the word and concept network is a crucial difference between TID and WordNet. In the WordNet model [7], concepts are most of the time lexicalized under the form of synonym sets – *synsets*. They are thus tied to the words of a specific language, i.e. English. In TID, Themes and Classes do not depend on the words of a specific language and it is possible to create a concept without any words. This is useful, for example, to build a node in the graph and share a semantic feature that is not entirely lexicalized.

A set of French adjectives shares the semantic feature *qui a cessé quelque chose: d'être, de subir, de devenir*, (that has ceased something: being, suffering, becoming) as the words *mort* (dead), which is no longer living, *démodé* (old-fashioned, outmoded), which is no longer modern. As it doesn't exist any French adjective, which exactly means *no longer*, it wouldn't possible to create a WordNet synset. In TID, there is no such a constraint and there is a class $\backslash Qui\ a\ cessé\ quelque\ chose[\backslash A]$.

2.2.2 The Size of the Integral Dictionary

The Integral Dictionary contains approximately 16,000 themes, 25,000 classes, the equivalent of 12,000 WordNet synsets (with more than one term in the set), and, for French 190,000 words. There is a total of 389,000 arcs in the graph. Table 1 shows the word breakdown by part-of-speech.

Table 1. Size of the lexicon broken down by part-of-speech.

Part-of-speech	Number
Nouns	138,658
Adjectives	20,981
Verbs	21,956
Adverbs	4,287

2.2.3 Componential Semantics

Componential semantics corresponds to the decomposition of the words into a set of smaller units of meaning: the *semes* (see [8] and [11]). The term 'seme' is not very common in English although this concept can prove very effective and instrumental in the construction of a semantic network. English-speaking linguists prefer the phrases *semantic feature* or *semantic component*, which are not exactly equivalent. Following the French semantic tradition, the interpretation of a text is made possible by the semes distributed amongst the words (see [6] and [8]). The repetition of semes in a text ensures its homogeneity and coherence and forms an *isotopy*.

One problem raised by the semic approach is the choice of primitives. Although, there is no consensus on it, a well-shared idea is that the primitives should be a small set of symbolic and atomic terms. This viewpoint may prove too restrictive and misleading in many cases. There are often multiple ways to decompose a word. It corresponds to possible paraphrases and to different contexts as for *fleuriste*:

Semes(fleuriste) = [personne/person] [vendre/sell] [fleur/flower]
Semes(fleuriste) = [vendeur/seller] [fleur/flower]
Semes(fleuriste) = [personne/person] [travailler/work] [magasin/shop] [vendre/sell] [fleur/flower]

The Integral Dictionary adopts a componential viewpoint but the decomposition is not limited to a handful of primitives as suggested by [13], *inter alia*. Any concept is a potential primitive and the possible semes of a word correspond to the whole set of concepts connected to this word. Word semes can easily be retrieved from the graph of themes and classes. This approach gives more flexibility to the decomposition while retaining the possibility to restrain the seme set to specific concepts (Figure 2).

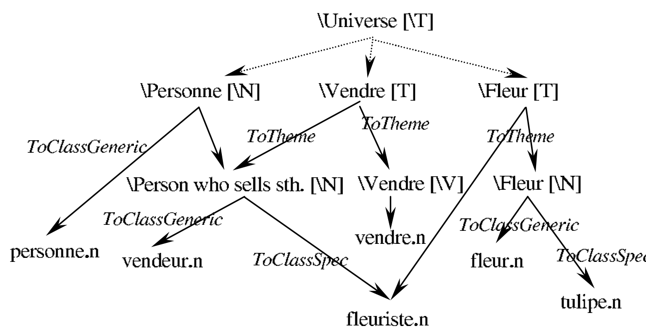

Figure 2. A part of the semantic decomposition of the word *fleuriste*.

2.2.4 Lexical Semantic Functions

Lexical semantic functions generate word senses from another word sense given as an input. Functions are divided into subsets. Amongst the most significant ones, the subset S0, S1, and S2, carries out semantic derivations of the verbs. These functions could be compared to nominalization in derivational morphology but they operate in the semantic domain and they are applied to a specific verb case:

- S0(*acheter*/buy.v) = {*achat*} (morphological nominalization)
- S1(*acheter*) = {*acheteur*}/{buyer} (subject nominalization),
- S2(*acheter*) = {*achat, marchandise, service*}/{purchase, goods, service} (object nominalization).

There are 66 lexical functions available in the Integral Dictionary. It corresponds to 96,000 links between the words. The links between adjectives and nouns are amongst the most productive ones in the French part of the dictionary.

3 AN ALGORITHM TO FIND WORDS FROM DEFINITIONS

The algorithm searches words using two main mechanisms. The first one extracts sets of words from the database that delimit the search space. In the definition *a person who sells daisies*, the algorithm extracts all the sets of persons. The second mechanism computes a semantic distance between each word in the person sets and the definition. This distance is asymmetric. It is based on the structure of the *differentia specifica* and the semantic topology of TID.

As we can imagine, such sets can be very large. The sets corresponding to *person* cover more than 10,000 words in TID. When needed, a third mechanism prunes rapidly the search space (see [2]).

3.1 The Semantic Network

The Integral Dictionary superimposes two graphs. The first one forms an acyclic graph whose terminal nodes are the words and the other nodes are concepts. The second one connects the words using lexical functions. The distance between two words or phrases is derived from the first graph.

Figure 3 shows a simplified picture of this structure. Nodes beginning with a backslash '\' are concepts while $W1$, $W2$, $W3$, etc. are words. The root node of the graph is the ancestor of all the concepts and is called the $\backslash Universe$. It has three children respectively $\backslash A$, $\backslash B$, and $\backslash C$, which can either be classes or themes.

Arc labelled Rn are relations linking the concepts and LFn are lexical functions. In Figure 3, $W3$ has two parents connected by arcs representing two different relations: $R14(\backslash E) = W3$ and $R16(\backslash F) = W3$. $LF1$ is a lexical function linking $W3$ to $W4$: $LF1(W3) = W4$. Inverse relations and lexical functions are implemented so that a parent can be found from its child.

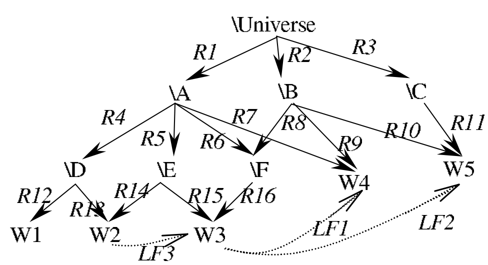

Figure 3. The graph of concepts, words, relations, and lexical functions.

The average number of parents of a word or a concept in TID is 2.1. The average depth of the graph from the root is 15. From these numbers, we can evaluate the average number of concepts a word can be member of: $15^{2.1} = 294$.

3.2 A Semantic Distance

The distance between two words or phrases is derived from the graph topology. It is the sum of two terms that we call respectively the semantic activation and the semantic proximity. We describe here a simplified version of this distance.

3.2.1 The Semantic Activation

The semantic activation of two words, M and N, is defined by their set of least common ancestors (LCA) in the graph (see [1]). The semantic activation paths correspond to paths linking both words M and N through each node in the set of least common ancestors.

In Figure 3, we have $LCA(W2, W3) = \{\backslash E\}$ and $LCA(W3, W4) = \{\backslash A, \backslash B\}$. The activation path between $W2$ and $W3$ consists of the nodes $W2 \backslash E\, W3$ with the functions $R14^{-1}$ and $R15$. The paths between $W3$ and $W4$ consist of $W3 \backslash E \backslash A\, W4$, $W3 \backslash F \backslash A\, W4$, and $W3 \backslash F \backslash B\, W4$.

We define the semantic activation distance as the number of arcs in theses paths divided by the number of paths. We denote it $d^\wedge$:

$$d^\wedge(W2, W3) = (1 + 1)/1 = 2 \quad (1)$$
$$d^\wedge(W3, W4) = ((2 + 1) + (2 + 1) + (2 + 1))/3 = 3 \quad (2)$$

Conceptually, the least common ancestors delimit small concept sets – small worlds (see [3]) – and provide a convenient access mode to them. They enable to extract a search space of potential semes together with a metric.

3.2.2 The Semantic Proximity

The semantic proximity between two words, M and N, uses sets of asymmetric ancestors that we call the Least Asymmetric Ancestors, LAA. $LAA(M, N)$ is the set of nodes that are common ancestors of both words, that are not member of the LCA set and where each member of the LAA set has at least one child, which is an ancestor of M and not an ancestor of N. Most of the time, the sets $LAA(M, N)$ and $LAA(N, M)$ are different. This is an essential feature of this metric, which reflects a semantic asymmetry.

In Figure 3, the set of the ancestors common to $W2$ and $W3$ that are not in the LCA set is equal to $\{\backslash A, \backslash Universe\}$. $\backslash A$ has a child $\backslash D$, which is an ancestor of $W2$ and which is not an ancestor of $W3$. Hence, $LAA(W2, W3) = \{\backslash A\}$ and $LAA(W3, W2) = \{\backslash A, \backslash Universe\}$ because $\backslash F$ and $\backslash B$ are children of respectively $\backslash A$ and $\backslash Universe$ and ancestors of $\backslash W3$ but not of $\backslash W2$.

The semantic asymmetry is the sum of distances of M to all the members of both LAA sets and N to all the members too:

$$SA(M, N) = \frac{\sum_{E \in LAA(M,N) \cup LAA(N,M)} d(M, E) + d(N, E)}{Card(LAA)} \quad (3)$$

We have:

$$SA(W2, W3) = (2 + 2)/1 = 4 \quad (4)$$
$$SA(W3, W2) = ((2 + 2) + (3 + 3))/2 = 5 \quad (5)$$

Finally, we define the semantic proximity as the sum of the semantic activation and the semantic asymmetry, $d_\perp = d^\wedge + SA$:

$$d_\perp(W2, W3) = (2 + 4)/2 = 3 \quad (6)$$
$$d_\perp(W3, W2) = (2 + 5)/2 = 3.5 \quad (7)$$

3.2.3 Examples of Semantic Activation and Semantic Proximity

In this section, we take the words *fleuriste* (florist) and *fleur* (flower, noun) to illustrate with concrete examples what the LCA and LAA sets are. The results enable to outline the componential structure of the dictionary and show understandable outputs in terms of semes. Although the words are entered in French, the concepts are roughly equivalent in English:

- $LCA(florist.n, flower.n) = \{\backslash Flower\ [T], \backslash RootOfTheNoun\ [\backslash Grammar]\}$
- $LAA(florist.n, flower.n) = \{\backslash TheWorldOfTheLiving\ [T], \backslash HumansAndSociety\ [T], X_i\ [T]\}$

where $X_i\ [T]$ denotes the remaining members of the LAA set.

The LAA set often contains the root of the whole dictionary. In our example, we obtain 107 LAA from *florist* to *flower.n*. The examination of the results shows that most of these LAA concepts are obtained through a small number of classes. To find them, we traverse the graph from the $\backslash Universe\ LAA$ down to the first class above *florist*. It enables to find that the differences between *florist* and *flower.n* originate in:

- $\backslash Person\ [\backslash N]$, which means that *flower.n* is not a person;
- $\backslash Person\ who\ sells\ something\ [\backslash N]$, which means that *flower.n* has no link with sales.

The study of the LCA set is also interesting. It contains:

- $\backslash Flower\ [T]$, which means that *flower.n* and *florist* share this seme;
- $\backslash RootOfTheNoun\ [\backslash Grammar]$, which means that both *flower.n* and *florist* share the noun part-of-speech.

In conclusion, this means that *florist* and *flower.n* are both nouns and that they both belong to the world of flowers. The difference between *florist* and *flower.n* is that a *florist* is a person and this person has the activity of *selling something*. These results show that LCA and LAA are powerful tools to derive common sense meaning and that they can be used to compare words.

3.2.4 Generalizing the Distances to Phrases

The parameters of the semantic activation and proximity can be generalized to phrases. If a parameter contains two or more words, it is replaced by a new virtual word in the graph that is obtained by merging the nodes representing all the content words in the phrase. The phrase *vendeur de fleurs* (seller of flowers) is represented in the graph by the virtual word *vendeur + fleur*. Semantically, it corresponds to an addition of their semes. It makes it possible to compute a distance between *vendeur de fleurs* and *fleuriste*.

In principle, a phrase can have any number of words. In some cases however, merging sub-graphs degrades the asymmetry properties of the resulting graph. In this case, we use other techniques to limit its effect. They use LAA, LCA, and positions of the words in the phrase. They can handle a more significant number of semes to represent complex phrases such as *a small-toothed wheel in the center of a bicycle wheel*.

3.3 Finding the Right Word

The algorithm finds words from definitions using two main mechanisms. The first one extracts the sets of words from the database that delimit the search space. In the definition *A person who sells daisies*, the algorithm extracts the hyponyms of *person*: the set of all the persons. It corresponds to more than 10,000 nouns in TID.

The second mechanism computes the distance between each candidate word in the set of persons and the words in the *differentia specifica*. To speed the algorithm for concepts covering a large number of words like $\backslash Person\ [\backslash N]$, a preliminary task attempts to reduce the search space (see [2] for details).

3.3.1 Extracting a Set of Hyponyms

The word sets are extracted using a function that finds for a given word all the hyponyms of one of its hypernyms. This extraction requires a composition of relations slightly more complex in the Integral Dictionary than in WordNet.

Figure 4 shows the hyponymy relationships of *flower.n* in both lexical networks. The word *begonia* is linked to *flower* by a single link in WordNet while TID requires two symmetrical links. A first link connects *fleur.n* to the class $\backslash Plante\ cultivée\ pour\ ses\ fleurs\ [\backslash N]$. A second link connects this class to *bégonia.n*. This feature makes the search more complex but adds more flexibility to describe the lexicon. In the end, it is possible to extract sets of related words using a composition of hypernymy and hyponymy functions in both networks. In TID, it corresponds to the $ToClassSpecific$ and $ToClassGeneric$ functions where $ToClassSpecific \circ ToClassGeneric(fleur.n) = \{bégonia.n, rose.n, tulipe.n, \ldots\}$. We call this composition $Specific$.

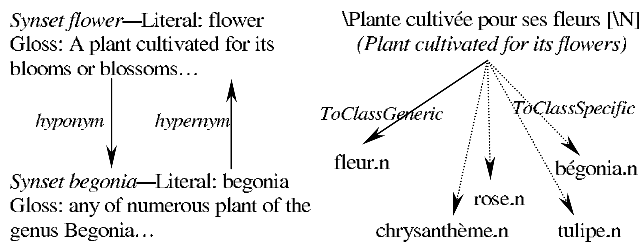

Figure 4. Hyponymy and hypernymy links in WordNet (left) and in the Integral dictionary (right).

3.3.2 Ranking the Extracted Words

To find answers to the query *vendeur de fleurs (seller of flowers)*, we first extract all the words corresponding to salespeople in the lexical database using the function $Specific(vendeur.n)$. It yields the set $\{vendeur, boulanger, boucher, papetier, fleuriste, bouquetière, etc.\}$ [3]. We then compute:

1. The semantic proximity between *vendeur de fleurs* and the extracted words: $d_\perp(vendeur\ de\ fleurs, X)$;
2. The semantic proximity between the extracted words and the phrase *vendeur de fleurs*: $d_\perp(X, vendeur\ de\ fleurs)$ where X is a member of the set $Specific(vendeur)$.

Let's consider the words *vendeur, bouquetière,* and *fleuriste* to see why both measures are necessary. Let's suppose that we only use the first asymmetric distance. Since the phrase *vendeur de fleurs* contains no seme describing the gender of the seller, it is impossible to distinguish between the masculine noun *fleuriste* and the feminine noun *bouquetière*. Both words cover all the semes of *vendeur de fleurs*. To make a difference between *fleuriste* and *bouquetière*, we need to compute $d_\perp(fleuriste, vendeur\ de\ fleurs)$ and $d_\perp(bouquetière, vendeur\ de\ fleurs)$. This second measure takes the feminine/female seme of *bouquetière* into account, which does not occur in *vendeur de fleurs*. On the contrary, *fleuriste* has all its semes covered by *vendeur de fleurs*.

[3] {*shop assistant, baker, butcher, stationer, florist, female flower seller, etc.*}

The seme sets of both *vendeur* and *fleuriste* are contained in that of *vendeur de fleurs*. Hence, the second measure can't differentiate between these two words although *vendeur* gives no hint about what the person is selling. On the contrary, the first measure makes a difference between *fleuriste* and the other members of the $Specific$ set. Using $d_{\perp}(vendeur\ de\ fleurs, fleuriste)$ and $d_{\perp}(vendeur\ de\ fleurs, vendeur)$, we observe that *fleuriste* is the only word that covers the semes of the query *vendeur de fleurs*. Figure 5 summarizes these results.

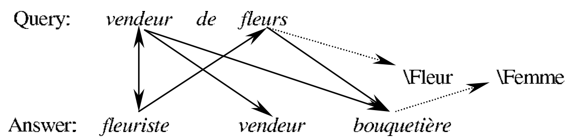

Figure 5. The first measure takes into account the seme *féminin/femme* and eliminates *bouquetière*. The second measure takes into account the seme *fleur* and eliminates *vendeur*.

4 RESULTS

Table 2 shows the words found by the algorithm for the phrase given in the introduction: *Personne qui vend des fleurs*. The probability to select the good answer (*fleuriste*) is 1/10,000. As we can see, the algorithm provides other words close to the definition: flower grower, flower seller, horticulturist, etc. These terms are ranked according to the proximity, the mean of the two measures given above. Lower numbers indicate a better relevance.

The algorithm was evaluated on a test corpus of approximately 200 definitions. The complexity of these definitions is comparable to that of the two examples in the introduction. Table 3 shows some results of lexical reductions we obtained. In this table, we considered a result as *good* if the correct answer was in the first 5 answers. The algorithm produces wrong results when the query contains:

- Generic words as *ce qui* in *ce qui effectue quelque chose* (what carries out something) or *relatif* in *relatif au Gabon* (relative to Gabon) because they are not linked to hypernyms in the graph.
- Negations. Both queries *small-toothed wheel* and *no small-toothed wheel* produce *gear wheel*.
- Functional relations that are meaningful to understand the definition. Both queries *person who sells something* and *person who sells someone* produce *seller*.

Table 2. Words corresponding to the phrase *Personne qui vend des fleurs*.

Rank	Word	English translation	Proximity
1	*Fleuriste*	Florist	1.34
2	*Floriculteur*	Flower grower	1.57
3	*Vendeur*	Shop assistant	1.77
4	*Bouquetière*	Flower seller in a street	1.84
5	*Horticulteur*	Horticulturist	2.21
6	*Rosiériste*	Rose grower	2.35
7	*Marchand*	Tradesman	2.57
8	*Maraîcher*	Market gardener	2.71
9	*Paysagiste*	Landscape gardener	3.12
10	*Fruiticulteur*	Fruit farmer	3.93

Table 3. Other results. The table shows only the word ranked first.

Query	Result
Crier pour un dindon	*Glouglouter*
Cry of a male turkey	Goggles
Vendeur de fleurs/magnolia/plantes	*Fleuriste*
Seller of flowers/magnolia/plants	Florist
Métal jaune	*Or/soufre*
Yellow metal	Gold/Sulfur
Métal de la finance	*Or/argent*
Metal of the finance	Gold/silver
Métal qui provoque des maladies	*Plomb/arsenic*
Metal which induces disease	Lead/arsenic
Petite roue dentée au centre d'une roue de bicyclette	*Pignon*
Small-toothed wheel in the center of a bicycle wheel	Sprocket-wheel

5 DISCUSSION AND PERSPECTIVES

We have described a lexical database and an algorithm to find words from definitions. We have presented examples of the results we obtained. The core of the algorithm rests on two functions, LCA and LAA, that query the database to find sets of semes describing similarities and differences between two words. In addition to finding words from definitions, the LCA and LAA functions help us to check the consistency of the lexical network. These functions should report semes corresponding to word differences and similarities. When the semes don't correspond, this generally indicates some faulty links in the network.

Currently, the algorithm has been applied mainly to the French part of TID. We intend to extend it to other languages. We are also applying the LCA and LAA functions to word-sense disambiguation.

REFERENCES

[1] Alfred V. Aho, John E. Hopcroft, and Jeffrey D. Ullman, 'On computing least common ancestors in trees', in *Proc. 5th Annual ACM Symposium on Theory of Computing*, pp. 253–265, (1973).

[2] Raphaël Berthelon, 'Le semiographe', Technical report, ISMRA, Caen, (1998). Rapport de stage.

[3] Ramon Ferrer I Cancho and Ricard V. Solé, 'The small-world of human language', *Proceedings of the Royal Society of London*, **B 268**, 2261–2266, (2001).

[4] Dominique Dutoit, 'A set-theoretic approach to lexical semantics', in *Proceedings of COLING*, (1992).

[5] Dominique Dutoit, *Quelques opérations sens→texte et texte→sens utilisant une sémantique linguistique universaliste a priori'*, Ph.D. dissertation, Université de Caen, 2000.

[6] Umberto Eco, *Lector in fabula*, Bompiani, 1979.

[7] *WordNet: An electronic lexical database*, ed., Christiane Fellbaum, MIT Press, 1998.

[8] Algirdas Julien Greimas, *Sémantique structurale*, Collection Champs sémiotiques, Presses universitaires de France, Paris, 1986.

[9] IST-2000-29388, 'Balkanet project', Technical report, European Union, (2000).

[10] Igor Mel'cuk, *Dictionnaire Explicatif et Combinatoire du français contemporain (DEC)*, volume III of *Recherche Lexico-sémantiques*, Presses de l'Université de Montréal, Québec, 1992.

[11] Bernard Pottier, *Linguistique générale, Théorie et description*, Klincksieck, Paris, 1974.

[12] Stephen D. Richardson, William B. Dolan, and Lucy Vanderwende, 'Mindnet: acquiring and structuring semantic information from text', in *Proceedings of COLING'98*, (1998).

[13] Roger C. Schank, 'Conceptual dependency: a theory of natural language processing', *Cognitive Psychology*, **3**, 552–631, (1972).

[14] Piet Vossen, 'Eurowordnet, final report', Technical Report D041, LE2-4003, LE4-8328, European Union, (1999).