Modelling, Detection and Exploitation of Lexical Functions for Analysis

Didier Schwab, Mathieu Lafourcade

Modelling, Detection and Exploitation of Lexical Functions for Analysis

Didier Schwab (1) and Mathieu Lafourcade (2)

(1) Computer-Aided Translation Unit (UTMK), School of Computer Sciences, Universiti Sains Malaysia Penang, Malaysia
(2) TAL-LIRMM, Université Montpellier II-CNRS 161 rue ada, 34392 Montpellier Cedex 5 - France

ABSTRACT

Lexical functions (LF) model relations between terms in the lexicon. These relations can be knowledge about the world (Napoleon was an emperor) or knowledge about the language (‹*destiny*› is synonym of ‹*fate*›). In this article, we show that LF instanciation in texts is useful both for semantic analysis (for example, resolution of lexical ambiguities or prepositional attachment and synthesis, i.e. natural language generation. We describe the architecture of a Semantic Lexical Base and the way how LFs are modeled, detected and used. More precisely, we show how each LF is modelled using thematic (conceptual vectors) and lexical (materialised relations between database objects) information and how we exploit the results in the base. We also describe how these functions allow the database to be explored continuously rather than in a discrete way.

1. INTRODUCTION

Many applications in Natural Language Processing, like automatic summarization (AS), information retrieval (IR) or machinal translation (MT), perform a semantic analysis (SA) which consists of, among other things, computing a thematic representation for the whole text and its components. In our case, thematic information is computed as a set of conceptual vectors which represent ideas and provide a quick estimation whether texts or their components (paragraphs, sentences or words) are part of the same semantic field, i.e. whether they have anything in common or not. At least four main problems should be solved during this step. (1) *lexical (word sense) ambiguity* (2) *references* i.e. anaphora resolution and identification of the coferents ; (3) *prepositional attachments* i.e. determination of the governor or head of the prepositional phrase ; (4) *interpretation paths* i.e. compatibility of the various ambiguities.

One way to resolve these different type of ambiguities is to use Lexical Functions (LF). LFs model typical relations between terms in the lexicon. Such relations are synonymy, the different types of antonymy, intensification ("*strong fear*", "*heavy rain*") or the typical relation of instrument (‹*to cut*› for ‹*knife*›, ‹*shovel*› for ‹*to dig*›). In this paper, we show that LFs are needed to model both world knowledge ("*Napoleon was an emperor*") and language specific knowledge (‹*destiny*› is synonym of ‹*fate*›). We will also show the central role this notions plays for semantic analysis and for resolving various kinds of ambiguities.

Finally, we present the architecture of a lexical semantic database built to model, detect and exploit LFs. We show that these LFs need a database composed of three types of lexical objects (LEXICAL ITEM, ACCEPTION, LEXIE) connected by materialised links and thematic information (conceptual vectors). They are automatically built from heterogenous resources like various kind of dictionaries (classic, synonym or antonym, etc.), thesauri, . . .

We present the *construction LF* in order to build conceptual vectors from other conceptual vectors. For example, an antonymy function allows the conceptual vector of ‹*existence*› to be built from the conceptual vector of ‹*non-existence*›. We present a neighborhood function allowing the estimation of the most appropriate word in the case of language generation. This is based on *evaluation LFs* which permit to estimate the relevance of a relation between two lexical objects. Hence, in our lexical database, relations are not directly materialised as in Wordnet [8] or FrameNet [33], they are computed from both thematic (conceptual vectors) and lexical (materialised links[1]) information. This allow us to explore data in a continuous way rather than in the classical discrete way.

2. SEMANTIC ANALYSIS

There are at least four kinds of semantic ambiguities which need to be resolved during SA : *lexical ambiguities*, *references*, *prepositional attachments* and *interpretation paths*.

2.1 Lexical Ambiguity

Words can have several meanings. This phenomenon known for ages[2] leads to one of the most important problems in NLP, lexical disambiguation (also often called Word Sense Disambiguation). It involves selecting the most appropriate acception of each word in the text. We define an acception as a particular meaning of a lexical item acknowledged and recognized by usage. It is a semantic unit acceptable in a given language [41]. For example, the lexical item ‹*mouse*› has at least three acceptions: the nouns referring to the ‹*computer device*›, to the ‹*rodent*› and the verb denoting the ‹*hunting*› of the animal. Unlike lexical items, acceptions are monosemantic.

Word Sense Disambiguation (WSD), i.e. the task of resolving lexical ambiguity, is a widely studied problem in SA [15]. For MT, it is essential to know which particular meaning is used in the source text as otherwise the wrong translation is likely to occur. For example, the English word ‹*river*› can be translated in French as ‹*fleuve*› or ‹*rivière*›. It is also important in information retrieval, as it helps eliminating documents which contain only inappro-

[1] Here, reader should begin to understand that we distinguish materialised links between lexical objects in the BLS and relations between these objects modelised thanks to LF.

[2] Sumerian known for their invention of the writing system, about 3200 years prior to our era, have a very polysemous language [11].

priate senses of a word with regard to the request, thereby increasing recall and precision.

2.2 References

Anaphora resolution is the phenomena whereby a pronoun is properly related to another element of the text. For example, in "*The cat climbed onto the seat, then it began to sleep.*", "it" refers to "cat" and not to "seat". Anaphoric resolution in MT is important as it associates pronouns to content nouns. Indeed, genders often vary according to the language. Thus, anaphoric resolution can help to translate the word which supports it. Therefore, in French, "it" can be translated either as "il" (masculine), as here in our case, or "elle" (feminine) whereas in German it could be either "er", "sie" or "es" since German has three genders. Note that in German the pronoun would be "sie" (feminine) and not masculine, as in French ("*Die Katze klaetterte auf den Sitz und (sie) begann dann zu schlafen*").

Identity is the phenomenon whereby two words refer to the same entity in real world as "cat" and "animal" in the following two sentences "*The cat climbed onto the chair. The animal began to sleep.*".

2.3 Prepositional Attachment

Prepositional attachment concerns finding the dependence link between a prepositional phrase and a syntactic head (verb, noun, adjective) [10]. In "*He sees the girl with a telescope.*" the prepositional phrase "*with a telescope*" can be attached either to the noun phrase "*the girl*" or to the verb phrase "see". Proper attachment is crucial in MT in particular. For a language like English, prepositions considerably modify verb meaning. In "*The man took a ferry across the river.*", the most logical attachment for ‹*across*› should be the verb ‹*to take*›, which in French would yield "*L'homme traversa la rivière en ferry.*". If it were attached to ‹*ferry*› we would express a different translation "*L'homme pris un ferry à travers la rivière.*".

2.4 Interpretation Paths

Due to semantic ambiguities, a sentence can have several interpretations. Such ambiguities occur often, especially in short texts as they contain less information. These ambiguities can be of various sorts, and they can be introduced on purpose by the author. The interested reader can find a good discussion and various examples concerning this phenomenon in [26]. We will show just one example here, "*The sentence is too long.*", which can be interpreted either as a phrase with a non-trivial length or as a condemnation with a non-trivial duration.

3. LEXICAL FUNCTIONS

3.1 Lexical and World Knowledge

The existence of a distinction between lexical knowledge (LK) and world knowledge (WK) has been subject of great debate ever since the beginning of the 1980's. According to John Haiman [12], there is no difference between the two, while Wierzbicka [45] argues that they are completely different. An interesting review can be found in Kornél Bangha's dissertation thesis [1] with respect to the status of lexical knowledge versus world knowledge and their respective roles in the process of interpretation. Here, we take an intermediary stance, close to Kornél Bangha's. We consider that knowledge can be divided into three categories: (1) *WK which is not directly lexicalised*, hence, which is not LK. For example, someone may know some facts concerning geography (location of New York), history (How and when did JFK die?) or everyday life (current price of the latest Ferrari). However, none of this information is lexicalised. The information can only be expressed via statements; (2) *WK which is directly lexicalised*. For example, the sentence "*During monsoon season, Penang has heavy rain*" is the expression of the fact in the real world that there is a certain amount of rain falling in Penang during Monsoon lexicalised as ‹*heavy*›; (3) *some LK which cannot be considered as lexicalisation of WK*. This is the case for grammatical gender in languages like French and German. Thus, the French lexical items ‹*voiture*› (‹*car*›) and ‹*piscine*› (‹*swimming pool*›) are feminine, yet there is no slightest correlation between the grammatical gender of these words and the objects they stand for.

3.2 LF for Linguistic Knowledge (LFLK)

LFLK are similar to Mel'čuk's LF [23]. They model LFs which correspond to linguistic knowledge. One must be aware of the fact that these functions also represent a state of the world, but this state is represented by a particular, but arbitrary (synchronically) item in the language. Thus, the sentence "*John had a strong fear*" corresponds to the real world situation describing the intense fear experienced by John, and is lexicalised by the *magnitude* LF *Magn* and one of its values, ‹*strong*›. There are two kinds of LFLK, *paradigmatics* which formalise classical semantic relations (synonymy, antonymy, ...) and *syntagmatics* which formalise collocations, "*combinations of lexical items which prevail on others without any obvious logical reason.*" [29]. In the first category we have:

- synonymy (*Syn*) which characterises different forms with the same meaning due only to use and without any direct relationship to reality. *Syn*(‹*plane*›)={‹*airplane*›, ‹*aeroplane*›,...};
- antonymies (*Anti*) which concern items whose semantic features are symetric relatively to an axis [39]. *Anti*(‹*life*›) = {‹*death*›,...}; *Anti*(‹*hot*›) = {‹*cold*›,...}
- generics (*Gener*) which correspond to substitution hypernyms i.e. terms of the hierarchy which are preferred to others as reference by use. To take an example, we do not say "*The vehicle has landed*" but "*the aircraft has landed*", hence Gener(‹*plane*›)={‹*aircraft*›} but not Gener(‹*plane*›)≠{‹*vehicle*›}. This function is different from hypernymy where Hyper(‹*plane*›)={‹*aircraft*›,‹*vehicle*›};

Concerning the syntagmatic LF, we have,

- adjectival LF like intensification (*Magn*) or confirmation (*Ver*). Magn(‹*tea*›)={‹*strong*›}; *Magn*(‹*rain*›)={‹*heavy*›}; *Ver*(‹*agreement*›)={‹*good*›, ‹*positive*›,...}
- collective Mult(‹*dog*›)={‹*pack*›} and its opposite Sing Sing(‹*rice*›)={‹*grain*›}

3.3 LF for the World Knowledge (LFWK)

LFWK allow the modelling of knowledge about the world. The following LFWKs are examples :

- *hypernymy* (*Hyper*) which is the class hypernymy contrary to *Gener* which is the substitution hypernymy. As already mentioned, the world knowledge "*a chair is a seat* " is retranscribed in language by the fact that ‹*seat*› is a hypernym of ‹*chair*› which is a LK. Hyper(‹*plane*›)={‹*aircraft*›,‹*vehicle*›,...};
- it's opposite relation, *hyponymy*. Hyponymy can be seen as the transcription in language of the property that a class is a subclass of another. Hypo(‹*aircraft*›)={‹*plane*›}, Hypo(‹*vehicle*›)={‹*plane*›,‹*car*›,‹*boat*›};
- *instance*(*Inst*) : *Inst*(‹*writer*›)={‹*Ernest Hemingway*›, ‹*Victor Hugo*›, ...}, *Inst*(‹*horse*›)= {‹*Tornado*›, ‹*Black*›,...};
- its opposite relation, *Class* : *Class*(‹*Ernest Hemingway*›)={‹*writer*›, ‹*American*›, ...}, *Inst*(‹*Black*›)= {‹*horse*›,...};
- *meronymy* (*Mero*), the part-of relation and its opposite *holonymy* (*Holo*). *Mero*(‹*plane*›)= {‹*fuselage*›, ‹*wing*›,...};
- verbal relations as *instrument* (*Instr*) which links an action to its typical instrument (*Instr*(‹*to dig*›)= {‹*pick*›,...} *Instr*(‹*to write*›)= {‹*pen*›,‹*keyboard*›,...} the *agent* relation (*agt*) which links an action to its typical agent and *patient* which links an action to its typical patient influenced by it. *agt*(‹*to eat*›)= ‹*cat*›; *pt*(‹*cat*›)= ‹*food*›.

3.4 Using of Lexical Functions

3.4.1 For Applications

Machine translation is certainly the main application for lexical functions. Indeed, Igor Mel'čuk introduced them in the early 60's to resolve some MT problems. He was then looking for "*a simple method allowing to avoid thousands of tedious tests necessary for a computer in order to find the russian equivalents of English lexemes...*" [23]. He noticed a phenomena common to most languages and well-known by translators : some terms are associated with others, whereas their direct equivalents are not used to mark a similar idea. Thus, we speak of "*grosse fièvre*" in French, but not of *"*big fever*" in English, where "*high fever*" will be used instead. Likewise, in Spanish we say ‹*fiebre*› ‹*alta*› or ‹*mucha*› but not ‹*gran*›. These phenomena are modelled by what is called lexical functions. They can be applied to any language in the same manner and are considered as universal. In MT, LF can be used as an *interlingua* i.e. as an intermediate language like in [14].

Information Retrieval can be divided into two phases. The first one, *documents indexing* consists of building a computational representation for each document. The second one, the *search phase*, consists of transforming the request into a similar representation and to extract the closest documents according to the given criteria. LFs can be useful to find synonymy of values. For example, we can imagine that the text representation does not directly refer to text segments like "*a high fear*" or "*crushing majority*" but rather to *Magn*(‹*fear*›) and *Magn*(‹*majority*›). Then, documents with "*a high fear*" or "*a strong fear*" and "*crushing majority*" or "*landslide majority*" would be more easily found than with simple distributional techniques used in systems like SMART [35] or *Latent Semantic Analysis* [6].

3.4.2 For solving SA Problems

LFs can provide some clues which can help in the various tasks discribed in section 2.

Lexical Disambiguation : The two types of LFs can help us:

- *LFLK*:to identify the syntagmatic relations between two words or at least to estimate its existence can help to identify the possible meanings for the corresponding lexical item. Thus, in "*At the time of his recent election to the senate, Mr Smith obtained a crushing majority.*" ‹*majority*› can be partly disambiguated thanks to the LF *Magn*. Indeed, we can consider that ‹*majority*› expresses a notion of age (some kind of adulthood), the proportional superiority in terms of vote or assembly, yet only *Magn*(*majority*/vote) = ‹*crushing*› and *Magn*(*majority*/assembly) = ‹*crushing*› exist. In the same vein, synonyms or generics can indirectly contribute to the clarification via identity relation.

- *LFWK*:These functions formalise world relations which can exist between the terms. Hence, information such as "*Renault has connection with cars*" or "*Napoleon was an emperor*" (the man at the head of a state and not the penguin) may contribute to lexical disambiguation. Clarification can be achieved here again, though indirectly, by disambiguating the identity relations thanks to hypernymy or instantiation.

Identity Relations Identification : These relations are partly supported by equivalent terms in context. They can be synonyms but also hypernyms. Knowing or identifying these relations in a text can thus be a determining element for the meaning reconstruction.

Prepositional Attachments : collocation information which are described with some LFLK (like the adjectival functions) can contribute to resolving prepositional attachments. A Web based method was tested in [10] where a large corpus was created to automatically extract lexical and statistical information on attachments to deduce the most probable ones in dependency syntactic analysis.

4. LF MODELLING : LEXICAL AND THEMATIC INFORMATION

4.1 Conceptual Vectors

4.1.1 Principle and Thematic Distance

We represent thematic aspects of textual segments (documents, paragraph, phrases, etc) by conceptual vectors. Vectors have long been used in information retrieval [34], for meaning representation in the LSI model [6] and for latent semantic analysis (LSA) studies in psycholinguistics. In computational linguistics, [4] proposed a formalism for the projection of the linguistic notion of semantic field in a vectorial space. Our model is inspired by this approach. Given a set of elementary concepts, it is possible to build vectors (conceptual vectors) and to associate them to any linguistic object. This vector approach is based on known mathematical properties. It is thus possible to apply well founded formal manipulations associated to reasonable linguistic interpretations. Concepts are defined from a thesaurus (in our prototype applied to French, we used the Larousse thesaurus [19] where 873 concepts are identified) to compare it with the thousand defined in Roget's thesaurus [16]). Let $\mathcal{C}$ be a finite set of n concepts, a conceptual vector V is a linear combinaison of elements c_i of $\mathcal{C}$. For a meaning A, a vector $V(A)$ is the description (in extension) of activations of all concepts of $\mathcal{C}$. For example, the different meanings of ‹*door*› could be projected on the following concepts (the CONCEPT⌈*intensity*⌋ is ordered by decreasing values): V(‹*door*›) = (OPENING⌈*0.8*⌋, BARRIER⌈*0.7*⌋, LIMIT⌈*0.65*⌋, PROXIMITY⌈*0.6*⌋, EXTERIOR⌈*0.4*⌋, INTERIOR⌈*0.39*⌋, ...

Comparison between conceptual vectors is based on angular distance. For two conceptual vectors A and B, $D_A(A,B) = arccos(Sim(A,B))$ where Sim is $Sim(X,Y) = \cos(\widehat{X,Y}) = \frac{X \cdot Y}{\|X\| \times \|Y\|}$. Intuitively, this function constitutes an evaluation of the *thematic proximity* and measures the angle between the two vectors. We would generally consider that, for a distance $D_A(A,B) \leq \frac{\pi}{4}$ (45°), A and B are thematically close and share many concepts. For $D_A(A,B) \geq \frac{\pi}{4}$, the thematic proximity between A and B would be considered as loose. Around $\frac{\pi}{2}$, they have no relation. D_A is a real distance function. It verifies the properties of reflexivity, symmetry and triangular inequality. We have, for example, the following angles (values are in radian and degrees).

D_A(V(‹*tit*›), V(‹*tit*›))=0 (0°)
D_A(V(‹*tit*›), V(‹*bird*›))=0.55 (31°)
D_A(V(‹*tit*›), V(‹*sparrow*›))=0.35 (20°)
D_A(V(‹*tit*›), V(‹*train*›))=1.28 (73°)
D_A(V(‹*tit*›), V(‹*insect*›))=0.57 (32°)

The first one has a straightforward interpretation, as a ‹*tit*› cannot be closer to anything else than to itself. The second and the third are not very surprising either since a ‹*tit*› is a kind of ‹*sparrow*› which is a kind of ‹*bird*›. A ‹*tit*› has not much in common with a ‹*train*›, which explains the large angle between them. One may wonder why ‹*tit*› and ‹*insect*›, are rather close with only 32° between them. If we scrutinise the definition of ‹*tit*› from which its vector is computed (*Insectivourous passerine bird with colorful feather.*) perhaps the interpretation of these values would seem clearer. Indeed, the thematic distance is by no way an ontological distance.

4.1.2 Limitation of Conceptual Vectors

*4.1.2.a For LF Detection*As shown in [2], distances computed on vectors are influenced by shared components and/or distinct components. Angular distance is a good tool for our aims because of its mathematical characteristics, its simplicity to understand and to linguistically interpret and ultimately allow it efficient implementation. Whatever chosen distance, used on this kind of vectors (represanting ideas and not term occurences), the smaller the distance, the bigger the number of lexical objects in the same semantic field (Rastier uses the term isotopy for this [31]).

In the framework of semantic analysis as outlined here, we use angular distance to take advantage of mutual information carried by conceptual vectors in order to make disambiguate words pertaining to the same or closely related semantic fields. Thus, "*Zidane scored a goal.*" can be disambiguated thanks to common ideas concerning sport, while "*The lawyer pleads at the court.*" can be disambiguated thanks to those of justice. Furthermore, vectors allow to attach properly prepositions due to knowledge about vision. For example, the prepositional phrase "*with a telescope*" would be attached to the verb "*saw*" in the sentence "*He saw the girl with the telescope.*".

On the contrary, conceptual vectors cannot be used to disambiguate terms pertaining to different semantic fields. Actually, an analysis solely based on them might lead to misinterpretation. For example, the French noun ‹*avocat*› has two meanings. It is the equivalent of ‹*lawyer*› and the equivalent of the fruit‹*avocado*›. In the French sentence "*L'avocat a mangé un fruit.*", "*The lawyer has eaten a fruit*", ‹*to eat*› and ‹*fruit*› convey the idea of ‹*food*›, hence the interpretation computed by conceptual vectors for ‹*avocat*› will be ‹*avocado*›. It would have been good to realize that "*a lawyer is a human*" and "*a human eats*", yet this is not possible by using only conceptual vectors. They are simply not sufficient to exploit the instanciation of LFs in texts, however, a lexical network can help to overcome these shortcomings. These kind of limitations have been shown in experiments for the semantic analysis using ant algorithms in [17].

4.1.2.b For LF Modelling.

We have shown in several publications that such a hybrid approach is needed for LF Modelling. For paradigmatic LFs, [40] used it for the three types of antonyms and [18] for generics and hypernyms.

For syntagmatic LF modelling, it seems difficult to model seemingly arbitrary collocations (as they do not have a common theme) with conceptual vectors.

4.2 Lexical Networks

4.2.1 Principles

Natural language processing has used lexical networks for more than fourty years, with Ross Quillian's work going back to the end of the sixtie's [30]. Authors differ concerning the network type and the way to use them. Some authors use directly graph microstructures (cliques, hubs) while others use them indirectly through similarity operations and/or activation of nodes (neural networks, pagerank).

The types of networks depends on entities chosen for nodes (lexical items, meanings, concepts) and on lexical relations chosen for edges. We can consider two families of lexical networks : (1) *semantic lexical networks* such as Quillan's [5], or, more recently, [43], WordNet [8], [7], where nodes correspond to lexical items, concepts or meanings and, usually, there are several kind of edges to qualify a relation (synonymy, antonymy, hypernymy, ...); (2) *distributional lexical networks* such as [44] where two terms are linked with an edge provided they cooccur in a corpus. In this kind of network there is only one type of edge.

For semantic analysis, lexical networks are used only for lexical disambiguation. On the other hand, Jean Véronis, for example, showed that distributional networks are *small worlds* and used this property to find every possible meaning for a word [44]. He made partitions on graphs to extract the different components organised around a hub, a central node to which are linked terms used in a same context. For a semantic analysis, these components are exploited while searching for the partition containing the words in the co-text of the target term.

The direct exploitation of the graph structure is also used with semantic network as in [42], following works of [28]. Only synonymy edges are used, their function being to look for cliques around the target word. In the given disambiguation examples, the complementary use of distributional data allows to guess the privileged meaning of an adjective depending on the noun to which it is related to.

With regard to the indirect use of the structure of the graph, it is done step by step by mutual activations and excitation of the nodes to cause compatible solution to emerge. [43], for example, use a technique inspired by *"neural networks"* on a graph made from dictionaries definitions while [24] built a network with words of a sentence and their possible meanings and edges weighted according to a similarity between definitions. Excitation of nodes is done with a *pagerank* [3] algorithm.

Very few authors use edge labels in their experiments. We have found only [27] who uses the *Leacock and Chodorow measure* [21] on WordNet based on *is-a* relations.

4.2.2 Limits of Lexical Networks

All these methods help to solve only one of the problems mentionned in section 3.i.e. lexical ambiguity. They provide a way to make a preference concerning the meaning of each word of a text taken individually. This last feature makes it impossible to even obtain the compatible paths of interpretation. By their very nature, it is hard to imagine how to extend the above mentioned methods in order to solve at least one of the other problems. Indeed, they all consider that the important information to be found in the networks lie only in the node, whereas in reality they **also** lie in the edges. However, as mentionned in part 3.4.2, to find the relations between items in a statement can contribute to the resolution of other types of ambiguity (e.g. lexical ambiguity).

Of course, this last comment has to be considered with respect to the specifically used networks. In the previous examples, none present both paradigmatic and syntagmatic information as the network we manage to build. Nevertheless, some research converges towards this idea. Syntagmatic information is crucially lacking in a network like WordNet. This phenomenon is known as the *tennis problem*. The lexical item ‹*racket*› is in one area while ‹*court*› and ‹*player*› are in others. Of course this is true, no matter what field chosen. Syntagmatic and paradigmatic relations are essential for natural and flexible access to the words and their meaning. Michael Zock and Olivier Ferret have made a very interesting proposal in this respect [9].

4.3 Hybrid Representation of Meaning : Mixing Conceptual Vectors and Lexical Network

While lexical networks offer unquestionable precision, their recall is poor. It is difficult to represent all possible relations between all terms. Indeed, how can we represent the fact that two terms are in the same semantic field? They may be absent from the network, because they are not connected by “traditional” arcs. Introducing arcs of the type “semantic field” is also problematic for us, because of two reasons, implied by the fuzzy and flexible nature of this relation: (1) the first one is related to the database creator’s understanding concerning this relation: when do two synsets belong to the same semantic field? In an unfavourable case there would be very few arcs, while in the extreme, opposite case we could have an explosion of arcs; (2) the second and more fundamental problem is related to the representation itself. How could a fuzzy relation, the essence of a continuous field, be represented by discrete elements?

Thus, the continuous domain offered by conceptual vectors provides flexibilities that the discrete domain offered by the networks cannot. They enable us to see connections between words including less common ones. A network, on the other hand, cannot do so, no matter how common the ideas are. Conceptual vectors and thematic distance can correct the weak recall inherent to lexical networks. This being so, conceptual vectors and lexical networks complemente each other, they are complementary tools: the weaknesses of one are alleviated by the strenght of the other.

4.4 Automatic Construction of a Semantic Lexical Database

In order to model, detect and exploit lexical functions for a semantic analysis, we need to build a database which allows to represent the meaning of as many words as possible. We call this database, semantic lexical Database (SLB). Let us present here quickly what kind of lexical objects are stored in the database, how they are linked and how the database is built. Our approach grounded on the following six hypotheses. For details, consult [38].

The first hypothesis, *hybrid representation of meaning based on a mixture of thematic (conceptual vector) and lexical approach (relations)* is the consequence of the ideas developed in section 4.3. Meaning is represented in the database by lexical objects, composed of a conceptual vector and lexical information like morphology, frequency concerning usage, lexical relations, etc. Each term of the lexicon is represented as a lexical object called LEXICAL ITEM.

A lexical item is a pointer concerning the particular meaning it can take in a text. To represent these meanings, our database stores one lexical object called ACCEPTIONS for each (hypothesis II, *Internal semantic relations* *of a lexical item*).

In classical dictionnaries like Larousse [20] or Robert [32] for French, there are about 80000 terms, most of which are polysemous. In our experience on French, dealing with more than 120000 entries, the polysemy rate is about 55%. For polysemous terms, there is an average of 5 definitions for each entry, hence we would have to index about 400000 ACCEPTIONS, which would be unreasonable to be done manually. Hypothesis III is the *automatic generation* of the ACCEPTIONS. This automation is done by bootstrapping from a reduced core of manually indexed ACCEPTIONS (approximately one thousand) and from information extracted from heterogeneous sources like traditional dictionaries, synonyms, antonyms dictionaries, Web sites, . . . A third kind of lexical object is defined by this hypothesis: a LEXIE gathers all information extractable from a definition.

The fourth hypothesis is to use a *multi-source analysis* in order to overcome the shortcomings of definitions (coverage of the lexicon, metalanguage).

The fifth hypothesis which allows the regular update of the base as well as the stabilization of the data is the idea of *permanent learning*.

The last hypothesis, is the *double loop*. It has been presented in previous publications [37] [40] [38], namely that not only a conceptual vector database could be improved by using conceptual vectors obtained by the lexical functions, but also that the results of these same functions are clearly improved by the use of lexical information and the corresponding vectors. Hence, not only do the functions improve, but their results, exploited by the method of training, can be used for new vector construction. The entire system grows richer by the contribution of the functions which themselves grow richer due to their contribution to the whole system.

Following this idea, we have developed a multi-agent system in order to build this database.

4.5 Modelling of Lexical Functions

4.5.1 Construction Lexical Functions

Construction LFs allow to build conceptual vectors from ohers. We saw in section 3.4.2 that LFs can help in semantic analysis. We will illustrate it here with an example on antonymy LFs. Let us consider the term ‹*unsuitable*› "*which is not suitable*", a definition extracted from the French dictionary [20] for the term. It is obvious, that it is not enough to find the correct ACCEPTION of the adjective ‹*suitable*›, in order to obtain an adequate conceptual vector. In this particular case, a construction lexical function of antonymy is necessary as we need to build an antonym vector from ‹*suitable*›. Likewise, in the case of the analysis of a synonym dictionary, we will build the vector of a synonym thanks to a construction lexical function of synonymy.

4.5.2 Evaluation Lexical Functions

Evaluation LFs measure the relevance of a lexical relation between several terms. These LFs have differents roles in our lexical database :

- for *relevance evaluation*, to allow evaluation of the global relevance of the database by checking the correspondence between links existing in language compared to those existing in the base;
- for *analysis*, to allow the ACCEPTION selection to evaluate whether two items in a text can be connected by a particular relation;
- for *generation*, to help in finding the best lexical item to use in a particular situation, i.e. item with the best evaluation according to a lexical function.

4.5.3 Thematic and Lexical Characteristic of LF

4.5.3.a Relations of both Thematic and Lexical Characteristic. This types of relations can be partly modelled with thematic information (conceptual vectors) which require to be supplemented by lexical information as we have shown with antonymy [39] and to a lesser extent with synonymy [38] and hypernymy [18].

Relations of both thematic and lexical characteristic exist with the two types of LFs :

- *LF for linguistic knowledge* : They correspond to Mel'čuk's paradigmatics. They are synonyms, antonyms and generics whose modelling for conceptual vectors is the same as hypernyms;
- *LF for world knowledge* : They are hypernymy, l'hyponymy, instance and the class function.

4.5.3.b Relations of a purely Lexical Characteristic. These relations cannot be represented using thematic information. We distinguish between:

- *LF for linguistic knowledge* : apart from synonymy, antonymy and generics, all the LFLK are purely lexical. They correspond, according to the typology of [29], to the syntagmatic LF which model collocations which are, as previously mentioned, "*combinations of lexical items which prevail on others without sign of logical reason.*". As there does not seem to be any logical reason for these relations, their nature being purely lexical.
- *LF for world knowledge* : a majority of the LFWK are purely lexical. For example, if we consider the meronymy relation, nothing in the theme of the items ‹*hand*› and ‹*finger*›, nor anything concerning ‹*mast*› and ‹*boat*› allows anyone to guess that finger is part of the hand while mast is part of the boat. In a similar vein, no linguistic information allows to predict that ‹*shovel*› is a typical instrument for performing the action of ‹*digging*› (relation *Instr*), or, that the place where sport activities are typically carried out is a ‹*stadium*› or a ‹*gymnasium*› (relation *loc*).

5. GENERALITIES ABOUT THE NETWORK

As we saw, the meaning representation of the lexical objects in the semantic lexical base uses partly relational nature information (cf. section 4.3). In the same way, whole or part of the modeling of a LF always requires explicitly specifying its relation in the semantic lexical base (cf. section 4.5.3). These relations are thus stored in the semantic lexical base. However, construction hypotheses of the semantic lexical database (SLB), the acquisition of these explicited relations is done automatically and thus cannot be boolean in nature. This is why we use Valued Lexical Relations (VLR).

5.1 Valued Lexical Relations

In traditional semantic networks, an arc links two nodes if a semantic relation exists between the two terms which correspond to them. Thus, one finds a meronymy relation between *‹leg›* and *‹body›* or an antonymy relation between *‹brother›* and *‹sister›* while there should be none between *‹elephant›* and *‹sister›* or between *‹leg›* and *‹to steal›*.

The valued lexical relations (VLR) are not boolean and have a value which expresses the probability of existence of a relation between two lexical objects (LEXICAL ITEMS, ACCEPTIONS, LEXIES). Thus, a VLR $\mathcal{R}$ is a relation which gives, for two lexical objects, a value between 0 and 1:

$$\mathcal{R} : \sigma^2 \rightarrow [0,1] \quad (1)$$

where σ is the set of the LEXICAL OBJECTS. The closer the value is to 1, the more likely is the existence of the relation between the two items, and symetrically, the closer the value to 0, the less likely the existence of the relationship between the two items. If the value is 0, we can consider that the relation does simply not hold between the two terms. For example, one can consider that R_{Anti}(‹*elephant*›,‹*sister*›) $= 0$ or that R_{Mero}(‹*leg*›,‹*plane*›) $= 0$ but R_{Anti}(‹*brother*›,‹*sister*›) and R_{Mero}(‹*leg*›,‹*body*›) should be close to 1.

Figure 1 presents an example of a valued lexical network. It is clear that in our base, links with a zero value are not explicitly specified,unlike the one between *‹leg›* and *‹plane›* which is present as in this example.

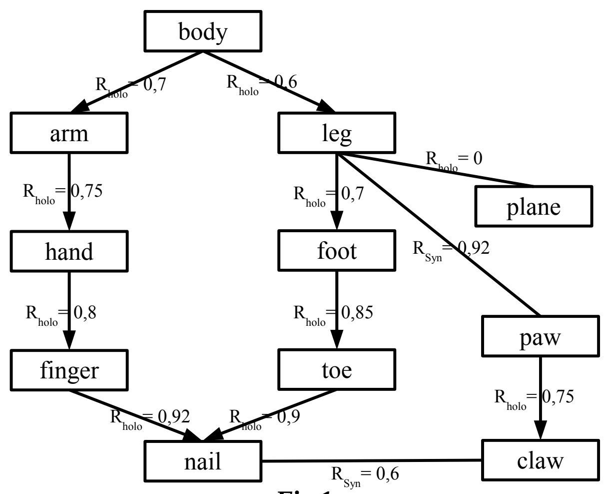

Fig.1:
EXAMPLE OF VALUED LEXICAL NETWORK.

5.2 Why use VLR in our approach?

5.2.1 VLR between LEXICAL ITEMS.

According to hypothesis IV, known as *multi-source analysis*, as a maximum number of sources is used to build lexical objects of the semantic lexical base. Hence, we can use traditional dictionaries, as well as semantic relation dictionaries or corpora like the Web.

The relations extracted from these sources are, of course, of unequal quality. Extraction from traditional dictionaries or specialized dictionaries of synonymy or antonymy is easy and of suitable quality, because attested already by lexicographers. Automatic extraction from corpora is much more problematic, though it has become the object of much research [13], [25], [5]. Thus, while one might consider information as quasi-foolproof if it comes from dictionaries, one cannot do the same if it is automatically extracted from a corpus. Weighting can be helpful to quantify the relevance of the discovered link.

5.2.2 VLR between ACCEPTIONS.

To be rigorously exact, one should not say that two terms are related but rather that two of their acceptions are related. It would thus be necessary that the lexical objects ACCEPTIONS are connected by VLR.

According to hypothesis III, objects construction of the lexical base is done automatically. Thus, it is by an automatic way that the majority of the links will be created. Uncertainties related to these automatic creations make necessary the use of VLR.

5.2.3 VLR between different lexical objects.

Our approach is based on a three-level hierarchy: LEXIES which correspond to the meaning of a term based on a particular source, ACCEPTIONS which gather information concerning the different LEXIES having the same meaning, and finally the LEXICAL ITEMS which gather all information concerning the ACCEPTIONS of this specific term. Network construction is made not only automatically from a single source (hypothesis III), but from several sources (hypothesis IV) and continuously (hypothesis V) to ensure that the base become coherent due to the repeated crossings of various information sources while at the end dubious, idealized, only ACCEPTIONS should be connected. Hence, VLR can connect various lexical objects, including ones of different type, during the network construction. One can find information which makes it possible to connect a LEXICAL ITEM resulting from a dictionary with others from the same dictionary, or some LEXIES with some LEXICAL ITEMS, with some ACCEPTIONS, etc. None of these are entirely foolproof, this is why it is wise to use VLRs.

Figure 2 presents an example of a lexical network with LEXICAL ITEMS and ACCEPTIONS.

6. LFs MODELLING

6.1 Construction and Evaluation LFs

6.1.1 Construction LFs

We have shown in section 4.5.3 the thematic and lexical characteristics of the LF. Creation of construction lexical function depends on this characteristic.

• *relations of both thematic and lexical characteristic*, we have shown that it is indeed possible to create such functions for synonymy [38] and antonymy [39]. For hyper-

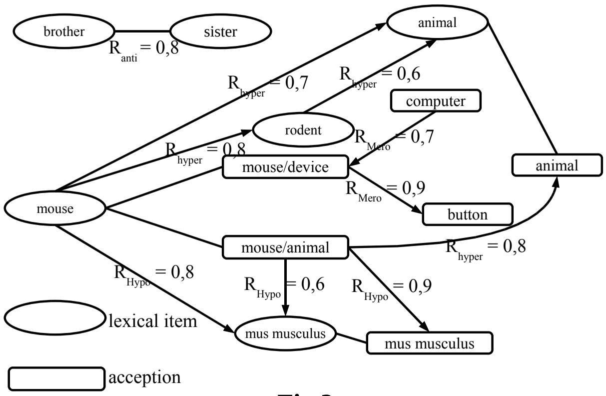

Fig.2:
EXAMPLE OF LEXICAL NETWORK IN OUR LSB

nymy and holonymy, it acts at the same time a difficult and useless operation. Indeed, we compute conceptual vectors thanks to dictonaries which use aristotelian definitions i.e. in *genus* (the hypernym) and *differentiae* (differences between hypernym and hyponym) which is exactly what could be done by a hypernymy function. A complete demonstration can be found in [38];

- *for relations of purely lexical kind*, such fonctions are impossible and useless to create.

6.1.2 Evaluation LF

An evaluation lexical function is a function which measures the relevance of the corresponding relation between two lexical objects. The value range lies between 0 and $\frac{\pi}{2}$ to be compatible with the evaluation LFs already presented (synonymy and antonymy) and with the thematic distance in order to ease the calculations using these tools.

A lexical function f which evaluates the relevance of a relation between the lexical objects x and y according to the lexical objects $z_1, \dots, z_m$ has the following characteristics :

$$\sigma^2 \times \sigma^m \longrightarrow [0, \frac{\pi}{2}] : x, y, z_1, \dots, z_m \longrightarrow f = F(x, y, z_1, \dots, z_m) \quad (2)$$

where σ is the set of lexical objects.

For relations of a purely lexical characteristic, the only information that we are likely to have is the existence probability of the relations on which the lexical object is dependent. We will consider that the evaluation is function of the probability of the relation.

Evaluation LF *for relations of both thematic and lexical character* are different according to the relations. We only mention them briefly here since we have examined them previously. For synonymy and antonymy, we thus showed that evaluation LFs based on the vectors and the lexical objects exist. On the contrary, for hypernymy, hyponymy and also instance or generic (which are close to the firsts), the creation of such a function is impossible [18] [38]. Here also, we consider, as for purely lexical characteristic functions, that the evaluation is function of relation probability if it exists.

Thus, we consider for all LF other than synonymy or antonymy that the corresponding evaluation LF is computed by using the following formula :

$$f = \frac{\pi}{2} R_f \quad (3)$$

This is the linear transformation from the interval $[0,1]$, that one of VLR, to the interval $[0, \frac{\pi}{2}]$, that of evaluation LFs. This passage is linear since it is based on the assumption that the more likely the relation the more important the corresponding VLR.

6.1.3 Important Points

- It is important to note that we clearly make a distinction between the explicit links in the LSB and the evaluation of a relation between objects (with evaluation LFs). We use the former combined, for some relations, with conceptual vectors to compute the latter;

- it is not because some LFs do not use conceptual vectors for modelling of their FLA that their VLR is not computed using conceptual vectors. For example, we can use conceptual vectors to make a decision concerning the preference between the ACCEPTIONS *mouse*/animal and *mouse*/computer for the hypernymy VLR between the lexical items ‹*mouse*› and ‹*rodent*› because *mouse*/animal and ‹*rodent*› share ideas about animals.

6.2 Neighbourhoud

6.2.1 Principle

The neighborhood function $\mathcal{V}$ is the function which returns the n closest LEXICAL OBJECTS to a lexical object x according to a ELF f and the lexical objects $u_1, \dots, u_m$:

$$\begin{aligned} &\mathcal{F} \times \sigma^m \times \mathbb{N} \rightarrow \sigma^n : \\ &f, x, u_1 \dots, u_m, n \rightarrow E = \mathcal{V}(f, x, u_1, \dots, u_m) \end{aligned} \quad (4)$$

where $\mathcal{F}$ is the set of evaluation lexical functions and σ the set of lexical objects. The function $\mathcal{V}$ is defined by :

$$\begin{aligned} &\left|\mathcal{V}(f, x, u_1, \dots, u_m)\right| = n, \\ &\forall y \in \mathcal{V}(f, x, u_1 \dots, u_m),\ \forall z \notin \mathcal{V}(f, x, u_1, \dots, u_m), \\ &f(x, y, \dots, u_m) \leq f(x, z, u_1, \dots, u_m) \end{aligned} \quad (5)$$

Neighborhood functions can be used for learning to check the overall relevance of the semantic base or to find the more appropriate word to use for a statement. Thus, they give us new tools to access words through a proximity notion to add to those described in [45] and issued from psycholinguistic considerations like form, part of speech, navigation in a huge associative network. They allow to navigate in a continuous way rather than in a discrete way as this is commonly done in semantic networks.

6.2.2 Examples

We consider here that the generalization of the neighbourhoud function can take as argument the thematic distance D_A which is not a LF :

$\mathcal{V}$(*Anti*, ‹*death*›, 7)=(‹*life*› 0.4) (‹*killer*› 0.449) (‹*murderer*› 0.467) (‹*blood sucker*› 0.471) (‹*strige*› 0.471) (‹*to die*› 0.484) (‹*to live*› 0.486)

$\mathcal{V}(D_A$, ‹*death*›, 7)=(‹*death*› 0) (‹*murdered*› 0.367) (‹*killer*› 0.377) (‹*age of life*› 0.481) (‹*tyrannicide*› 0.516) (‹*to kill*› 0.579) (‹*dead*› 0.582)

7. APPLICATION FOR FRENCH

We have implemented BLEXISMA (*Base LEXicale Sémantique Multi-agent*, multi-agent semantic lexical database), a multi-agent architecture which focuses on the integration of all functionalities to create, enhance and exploit one or several Semantic Lexical Database. Our first experiment was on French. The database contained about 121 000 LEXICAL ITEMS, 276 000 ACCEPTIONS, 842 000 LEXIES and 503 000 VLR (essentially antonymy and synonymy).

This experiment shows that the developpement of a such base is possible. It has been used for semantic analysis using ant algorithms which allow the resolution of some of the problems presented in section 2.[36]. We showed how it is possible to model lexical functions: construction LF to exploit synonymy and antonymy dictionaries and evaluation LFs based on VLR automatically built. Grounded on these last function a neighborhood can be performed for all LFs.

8. CONCLUSIONS AND PERSPECTIVES

We have presented in this article a Lexical Semantic Database which permits to model, detect and exploit Lexical Functions. We have presented its architecture composed of three types of lexical objects (LEXICAL ITEM, ACCEPTION, LEXIE) linked by materialised relations (VLR). They are automatically built from heterogenous resources like dictionaries, thesaurus, synonymy and antonymy dictionaries. We presented construction LFs to build conceptual vectors from these sources, evaluation LF to estimate the relevance of a relation between lexical objects and the neighborhood function which allows the database to be explored continuously rather than in a classic discrete way.

The database presented here allows the use of LF for both analysis and generation. Unlike classic semantic databases (Wordnet, MindNet or Cyc), relations between terms are not only in the links but also in thematic aspects (conceptual vectors) and can be interpreted only through lexical functions.

We are currently following the same principle to develop a multilingual project between French, English and Malay. As in Papillon [22], the idea is to etablish links between axies (interlingual acceptions).

The authors would like to thank Michael Zock and anonymous referees for helpful comments and suggestions. We are, of course, responsible for any remaining errors.

References

[1] Kornél Robert BANGHA. "*La place des connaissances lexicales face aux connaissances du monde dans le processus d'interprétation des énoncés*". PhD thesis, Université de Montréal, Montréal, Québec, Canada, 2003.

[2] Romaric BESANÇON. "*Intégration de connaissances syntaxiques et sémantiques dans les représentations vectorielles de texte*". Thèse de doctorat (PhD. thesis), École Polytechnique Fédérale de Lausanne, Laboratoire d'Intelligence Artificielle, 2001.

[3] Sergey BRIN et Lawrence PAGE. "The anatomy of a large-scale hypertextual Web search engine". *Computer Networks and ISDN Systems*, pp 107–117, 1998.

[4] Jacques CHAUCHÉ. "Détermination sémantique en analyse structurelle : une expérience basée sur une définition de distance". *TAL Information*, pp 17–24, 1990.

[5] Vincent CLAVEAU. "*Acquisition automatique de lexiques sémantiques pour la recherche d'information*". Thèse de doctorat (PhD. thesis), Rennes I, 2003.

[5] Alan COLLINS et Ross QUILLIAN. "Retrivial time from semantic memory". *Verbal learning and verbal behaviour*, pp 240–247, 1969.

[6] Scott C. DEERWESTER, Susan T. DUMAIS, Thomas K. LANDAUER, George W. FURNAS, et Richard A. HARSHMAN. "Indexing by Latent Semantic Analysis". *Journal of the American Society of Information Science*, pp 391–407, 1990.

[7] Dominique DUTOIT. "*Quelques opérations Sens → texte et texte → Sens utilisant une sémantique linguistique univerliste a priori*". Thèse de doctorat (PhD. thesis), Université de Caen, 2000.

[8] Christiane FELLBAUM, . *WordNet: An Electronic Lexical Database*. The MIT Press, 1988.

[9] Olivier FERRET et Michael ZOCK. "Enhancing Electronic Dictionaries with an Index Based on Associations". In the proceedings of *Proceedings of the 21st International Conference on Computational Linguistics*, pp 281–288, Sydney, Australia, July 2006.

[10] Núria GALA PAVIA. "Une méthode non supervisée d'apprentissage sur le Web pour la résolution d'ambiguïtés structurelles liées au rattachement prépositionnel. ". In the proceedings of *TALN'2003*, pp 353–358, Batz-sur-Mer, France, 2003.

[11] Jean-Jacques GLASSNER. "L'invention de l'écriture sumérienne : système de notation ou langage?". *Les actes de lecture*, pp 94–103, 2001.

[12] John HAIMAN. "Dictionaries and encyclopedias". *Lingua*, pp 329–357, 1980.

[13] Marti HEARST. "Automatic Acquisition of Hyponyms from Large Text Corpora". In the proceedings of *COLING'1992*, pp 539–545, Nantes, France, 1992.

[14] Dirk HEYLEN, Kerry G. MAXWELL, et Marc VERHAGEN. "Lexical functions and machine translation". In the proceedings of *COLING'1994*, volume 1, pp 1240–1244, Kyoto, Japan, 1994.

[15] Nancy IDE et Jean VERONIS. "Word sense disambiguation: the state of the art". *Computational Linguistics*, pp 1–41, 1998.

[16] Betty KIRKPATRICK, . *Roget's Thesaurus of English Words and Phrases*. Penguin books, London, 1987.

[17] Mathieu LAFOURCADE et Frédéric GUINAND. "Ants

for Natural Language Processing". *International Journal of Computational Intelligence Research*, 2006. to appear

[18] Mathieu LAFOURCADE et Violaine PRINCE. "Mixing Semantic Networks and Conceptual Vectors: the Case of Hyperonymy". In the proceedings of *ICCI-2003 (2nd IEEE International Conference on Cognitive Informatics)*, pp 121–128, South Bank University, London, UK, 2003.

[19] LAROUSSE, . *Thésaurus Larousse - des idées aux mots, des mots aux idées*. Larousse, 1992.

[20] LAROUSSE, . *Le Petit Larousse Illustré 2004*. Larousse, 2004.

[21] C. LEACOCK et M. CHODOROW. "*WordNet: An electronic lexical database*", Combining local context and WordNet similarity for word sense identification. MIT Press, 1998.

[22] Mathieu MANGEOT-LEREBOURS, Gilles SÉRASSET, et Mathieu LAFOURCADE. "Construction collaborative d'une base lexicale multilingue : Le projet Papillon". *TAL (Traitement Automatique des langues) : Les dictionnaires électroniques*, pp 151–176, 2003.

[23] Igor MEL'ČUK. "*Lexical Functions in Lexicography and Natural Language Processing*", Lexical Functions: A Tool for the Description of Lexical Relations in the Lexicon, pp 37–102. Benjamins, Amsterdam/Philadelphia, 1996.

[24] Rada MIHALCEA, Paul TARAU, et Elizabeth FIGA. "PageRank on Semantic Networks, with Application toWord Sense Disambiguation". In the proceedings of *COLING'2004*, pp 1126–1132, Geneva, Switzerland, 2004.

[25] Emmanuel MORIN. "*Extraction de liens sémantiques entre termes à partir de corpus techniques*". Thèse de doctorat (PhD. thesis), Université de Nantes, 1999.

[26] Peter NORVIG. "Multiple simultaneous interpretation of ambiguous sentences". In the proceedings of *10th annual conference of the cognitive science society*, Août 1988.

[27] Siddharth PATWARDHAN, Satanjeev BANERJEE, et Ted PEDERSEN. "Using Measures of Semantic Relatedness for Word Sense Disambiguation". In the proceedings of *the Fourth International Conference on Intelligent Text Processing and Computational Linguistics*, Mexico City, February 2003.

[28] Sabine PLOUX et Bernard VICTORRI. "Construction d'espaces sémantiques à l'aide de dictionnaires informatisés des synonymes". *Traitement automatique des langues*, 1998.

[29] Alain POLGUÈRE. *Lexicologie et sémantique lexicale*. Les Presses de l'Université de Montréal, 2003.

[30] Ross QUILLIAN. "*Semantic Informatic processing*", Semantic memory, pp 227–270. MIT Press, 1968.

[31] François RASTIER. "*L'isotopie sémantique, du mot au texte*". Thèse de doctorat d'État, Université de Paris-Sorbonne, 1985.

[32] Le ROBERT, . *Le Nouveau Petit Robert, dictionnaire alphabétique et analogique de la langue française*. Éditions Le Robert, 2000.

[33] Joseph RUPPENHOFER, Michael ELLSWORTH, Miriam R.L. PETRUCK, Christopher R. JOHNSON, et Jan SCHEFFCZYK. "FrameNet II: Extended theory and practice". , Berkeley, 2006. http://framenet.icsi.berkeley.edu/book/book.pdf.

[34] Gerard SALTON et Michael MCGILL. *Introduction to Modern Information Retrieval*. McGrawHill, New York, 1983.

[35] Gerard SALTON. "The Smart Document Retrieval Project". In the proceedings of *Proc. of the Fourteenth Annual International ACM/SIGIR Conference on Research and Development in Information Retrieval*, pp 357–358, Chicago, IL, 1991.

[36] Didier SCHWAB et Mathieu LAFOURCADE. "Lexical Functions for Ants Based Semantic Analysis". In the proceedings of *ICAI'07- The 2007 International Conference on Artificial Intelligence*, 2007. to appear.

[37] Didier SCHWAB. "Société d'agents apprenants et sémantique lexicale : comment construire des vecteurs conceptuels à l'aide de la double boucle". In the proceedings of *RECITAL'2003*, pp 489–478, Batz-sur-Mer, France, 2003.

[38] Didier SCHWAB. "*Approche hybride - lexicale et thématique - pour la modélisation, la détection et l'exploitation des fonctions lexicales en vue de l'analyse sémantique de texte.*". Thèse de doctorat (PhD. thesis), Université Montpellier 2, 2005.

[39] Didier SCHWAB, Mathieu LAFOURCADE, et Violaine PRINCE. "Antonymy and Conceptual Vectors". In the proceedings of *COLING'2002*, volume 2/2, pp 904–910, Taipei, Taiwan, 2002.

[40] Didier SCHWAB, Mathieu LAFOURCADE, et Violaine PRINCE. "Vers l'apprentissage automatique, pour et par les vecteurs conceptuels, de fonctions lexicales. L'exemple de l'antonymie". In the proceedings of *TALN 2002*, volume 1, pp 125–134, Nancy, 2002.

[41] Gilles SÉRASSET et Mathieu MANGEOT. "Papillon lexical databases project: monolingual dictionaries and interlingual links". In the proceedings of *NLPRS 2001*, pp 119–125, 2001.

[42] Fabienne VENANT. "*Représentation et calcul dynamique du sens : exploitation du lexique adjectival du français*". Thèse de doctorat (PhD. thesis), École des hautes études en sciences sociales, 2006.

[43] Jean VÉRONIS et Nancy IDE. "Word Sense Disambiguation with Very Large Neural Networks. Extracted from Machine Readable Dictionaries". In the proceedings of *COLING'1990*, volume 2, pp 389–394, 1990.

[44] Jean VÉRONIS. "Hyperlex : lexical cartography for information retrieval". *Computer, Speech and Language*, pp 223–252, 2004.

[45] Anna WIERZBICKA. *Semantics: Primes and Universals*. Oxford University Press, 1996.

[45] Michael ZOCK. "Sorry, What Was Your Name Again, Or How to Overcome The Tip-Of-The Tongue with the help of a computer?". In the proceedings of *SemaNet'02: Building and Using Semantic Networks*, Taipei, Taiwan, 2002.