Hardening of Acception Links Through Vectorized Lexical Functions

Didier Schwab and Mathieu Lafourcade

LIRMM
Laboratoire d'informatique, de Robotique
et de Microélectronique de Montpellier
MONTPELLIER - FRANCE.
{schwab,lafourca}@lirmm.fr
http://www.lirmm.fr/ ~{schwab, lafourca}

Abstract. In the framework of the Papillon project, we have defined strategies for populating a pivot dictionnary of interlingual links from monolingual vectorial bases. There are quite a number of acception per entry thus, the proper identification may be quite troublesome and some added clues beside acception links may be usefull. We improve the integrity of the acception base through welsl known semantic relations like synonymy, antonymy, hyperonymy and holonymy relying on lexical functions agents. These semantic relation agents can compute the pertinence of a semantic relation between two acceptions thanks to various lexical informations and conceptual vectors. When a given pertinence score is above a threshold they create a semantic link which can be walked throught by other agents in charge of WSD ot lexical transfert. Base integrity agents walk throw the acceptions and according to their speciality creates semantics links, look for incoherences in the base and emit warning toward lexicographs when neened.

1 Introduction

Research in meaning representation in NLP is an important problem still addressed through several approaches. The NLP team from LIRMM currently works on thematic text analysis and lexical disambiguation [*Lafourcade*, 2001]. To this purpose, we built a system, with automated learning capabilities, based on conceptual vectors for meaning representation. Vectors are supposed to encode 'ideas' associated to words or expressions. The conceptual vectors learning system automatically defines or revises its vectors from definitions in natural language contained in electronic dictionaries for human usage. In the framework of the Papillon project, we have defined strategies for populating a pivot dictionnary of interlingual links from monolingual vectorial bases. One given acception corresponds to a meaning of an entry of a monolingual dictionnary. The overall architecture being as defined in [*Sérasset and Mangeot*, 2001] and [*Mangeot*, 2001] Such a pivot dictionnary can be used with great avantages for word sense disambiguation and lexical transfert. As there are quite a number of

word meaning per entry (roughly 5 for French in our experiments with about 87000 entries) thus, the proper identification of corresponding acceptions may be quite troublesome and some added clues beside acceptions links may be usefull. To do so, we improve the integrity of the acception base through well known semantic relations like synonymy, antonymy, holonymy and hyponomy relying on lexical functions agents. These semantic relation agents can compute the pertinence of a semantic relation between two acceptions by combining various lexical informations and conceptual vectors. When a given pertinence score is above a threshold, they create a semantic link which can be walked throught by other agents in charge of WSD ot lexical transfert. Base integrity agents walk throw the acceptions and according to their speciality create semantic links, look for incoherences in the base and emit warnings toward lexicographs when neened. In this paper, we first expose the conceptual vectors model, then, the acceptance model, a few reviewing on semantics relations in an acceptions base and then we present precisely a detailled example with antonymy.

2 Conceptual Vectors

We represent thematic aspects of textual segments (documents, paragraph, syntagms, etc) by conceptual vectors. Vectors have been used in information retrieval for long [*Salton et MacGill*, 1983] and for meaning representation by the LSI model [*Deerwester et al*, 90] from latent semantic analysis (LSA) studies in psycholinguistics. In computational linguistics, [*Chauché*, 90] proposes a formalism for the projection of the linguistic notion of semantic field in a vectorial space, from which our model is inspired. From a set of elementary notions, concepts, it is possible to build vectors (conceptual vectors) and to associate them to lexical items [1]. The hypothesis[2] that considers a set of concepts as a generator to language has been long described in [*Rodget*, 1852]. Polysemic words combine different vectors corresponding to different meanings. This vector approach is based on well known mathematical properties, it is thus possible to undertake well founded formal manipulations attached to reasonable linguistic interpretations. Concepts are defined from a thesaurus (in our prototype applied to French, we have chosen [*Larousse*, 1992] where 873 concepts are identified)/. To be consistent with the thesaurus hypothesis, we consider that this set constitutes a generator space for the words and their meanings. This space is probably not free (no proper vectorial base) and as such, any word would project its meaning on this space according to the following principle. Let be $\mathcal{C}$ 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] are ordered by decreasing

[1] Lexical items are words or expressions which constitute lexical entries. For instance, ‹*car*› or ‹*white ant*› are lexical items. In the following we will (some what) use sometimes *word* or *term* to speak about a *lexical item*.

[2] that we call thesaurus hypothesis.

values): V(‹*door*›) = (*OPENING*[0.8], *BARRIER*[0.7], *LIMIT*[0.65], *PROXIMITY*[0.6], *EXTERIOR*[0.4], *INTERIOR*[0.39], ...

In practice, the largest $\mathcal{C}$ is, the finer the meaning descriptions are. In return, the computing is less easy. It is clear that, for dense vectors[3], the enumeration of activated concepts is long and difficult to evaluate. We would generally prefer to select the thematically closest terms, i.e., the *neighbourhood*. For instance, the closest terms ordered by increasing distance to ‹*door*› are: $\mathcal{V}$(‹*door*›)=‹*portal*›, ‹*portiere*›, ‹*opening*›, ‹*gate*›, ‹*barrier*›,...

2.1 Angular Distance

Let us define $Sim(A, B)$ as one of the *similarity* measures between two vectors A et B, often used in information retrieval [*Morin*, 1999]. We can express this function as: $Sim(A, B) = \cos(\widehat{A, B}) = \frac{A \cdot B}{\|A\| \times \|B\|}$ with "·" as the scalar product. We suppose here that vector components are positive or null. Then, we define an *angular distance* D_A between two vectors A and B as $D_A(A, B) = \arccos(Sim(A, B))$. 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 degrees) 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)
D_A(V(‹*tit*›), V(‹*color*›))=0,59 (33)

The first one has a straightforward interpretation, as a ‹*tit*› cannot be closer to anything else than itself. The second and the third are not very surprising 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 a large angle between them. One can wonder why there is 32 degrees angle between ‹*tit*› and ‹*insect*›, which makes them rather close. 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 seems clearer. In fact, the thematic is by no way an onthological distance.

2.2 Conceptual Vectors Construction.

The conceptual vector construction is based on definitions from different sources (dictionaries, synonym lists, manual indexations, etc). Definitions are parsed and

[3] Dense vectors are those which have very few null coordinates. In practice, by construction, all vectors are dense.

the corresponding conceptual vector is computed. This analysis method shapes, from existing conceptual vectors and definitions, new vectors. It requires a bootstrap with a kernel composed of pre-computed vectors. This reduced set of initial vectors is manually indexed for the most frequent or difficult terms. It constitutes a relevant lexical items basis on which the learning can start and rely. One way to build an coherent learning system is to take care of the semantic relations beetween items. Then, after some fine and cyclic computation, we obtain a relevant conceptual vector basis. At the moment of writing this article, our system counts more than 84000 items for French and more than 315000 vectors.

3 Lexical base description

The model is composed of two parts: monolinguals dictionnaries and an intermediary base, the acceptions[4] base. Every entry of the monolingual dictionnaries are linked to one acception (cf fig. 1).

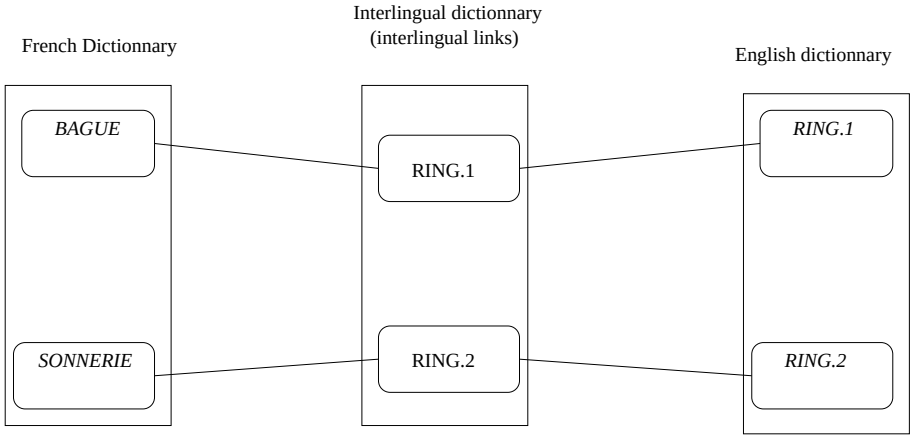

Fig. 1. Lexical Base Architecture

The base acception is build from vectorial bases and each acception has his own vector.

The acception base is very large and is built by differents agents which can be human or artificial. We argue that it is very difficult to maintain base integrity without control: an agent (certainly human) can create a new acception when an adequate acception already exists.

One way to assess this integrity is to check the semantic links between acceptions. For example, if an acception is the opposite of an other, it cannot be also

[4] Acceptions represent each sense or meaning of each entry of a monolingual dictionnary [*Sérasset and Mangeot*, 2001]. For exemple, the english item ‹*ring*› has at least two acceptions the jewel and the sound.

its synonym. We consider several semantic links in lexicon and we show how to use them to verify integrity of the base or how a non-specialist agent can walk through links and evaluate news.

4 Semantic Links and Integrity Constraints

The semantic relations between lexical units help structuring the lexicon on the paradigmatic plane. These well known relations are often described in two types: the hierarchical relations (hyponymy, hyperonymy, meronymy and holonymy) and the equivalence/opposition relations (synonymy, antonymy). In some linguistic research ([*Polguère, 2001*], [*Lehmann et Martin-Berthet*, 98]) they are defined as boolean relations i.e. they either hold between two words or they don't. For acceptions, which are monosemic, hierarchical relations are transitive and synonymy is considered as an equivalence. If we had to explicit all relations, we may hit the problem of links number explosion. It is a praticial problem (memory size) but it also push forward the question of the link relative relevance: their discriminanting power is inversely proportionnal to their number. For instance, every noun acceptions would be linked to a general term like ‹*concrete object*› by an hyperonymy link. For the acception ‹*cat*›, ‹*feline*› seems a better hyperonym than ‹*animal*› or ‹*mammal*›. To avoid this problem, and to be able to compare two links, we rely on *valued semantic relations*.

4.1 Valued Semantic Relations

Valued semantic relations (RSV) are not boolean and hold a value which express the relevance of the relation between two acceptions. A valued semantic relation $\mathcal{R}$ is a relation which gives, for two items, a value between 0 and 1:

$$\mathcal{R} : w^2 \rightarrow [0, 1]$$

with w as the set of the lexical units.

The nearest from 1 the value is, the more relevant the relation between the two items is. The nearest from 0 the value is, less relevant the relation between the two items is. If the value is strictly 0 then we consider that the relation doesn't apply between these two terms. We can view this value as the probability that the relation exists.

4.2 Links Creation and Deletion

We want to add semantics links to the acception base in order to assess the integrity of the base. Agents which can evaluate a given semantic relation can create valued semantics links if the valuation result is above a threshold th. For exemple, if one antonymy agent evaluates antonymy between ‹*cold*› and ‹*hot*› at a value above th, it builds a semantic link between the two acceptions valued at th. This threshold is not fixed in advance, and it evolves according to the number of

links already built. Thus, this value will evolve during time. The system learns from new data (new monolingual or bilinguals dictionnaries) or by revising old data so agents can compute again a relation and if the condition to preserve the link, to be above th, is not verified the link is deleted. These materialised valued links can be walked through by standart agents (which can't compute the semantic relation value by themselves) which can, with few simple rules, quickly evaluate relations values between two acceptions. For example, an agent can use the transitivity property of hyperonymy to evaluate le values of $Hyp(A, C)$ from $Hyp(A, B)$ and $Hyp(B, C)$. This can be usefull if the acception base is used for a word sense desambiguation, for example. In no case, a non-expert agent can build a link.

We presents the different semantic relations and the rules that can be applied to verify base integrity and to deduct new relations from existing ones.

4.3 Hierarchical Relations

Hierarchical relations between lexical acceptions hold when they are not at the same level in the hierarchy lattice. If $\mathcal{R}$ is a hierarchical relation between two acceptions A and B then it exists a symmetrical relation $\overline{\mathcal{R}}$:

$$\mathcal{R}(A, B) = \overline{\mathcal{R}}(B, A)$$

Agents use these properties to avoid to compute the symmetrical relation if it already exist. We have chosen somewhat arbitrary to choose the relation which goes from the general to the particular i.e. hyperonymy and holonymy.

These relations are hierachical and as such, they should verify transitivity.

$$A\mathcal{R}B \wedge B\mathcal{R}C \rightarrow A\mathcal{R}C$$

Non-expert agents can evaluate $\mathcal{R}(A, C)$ with a derived rule:

$$\mathcal{R}(A, B) = Min_{i \in I}(\mathcal{R}(A, X_i) \times \mathcal{R}(X_i, B))$$

where I is the set of the acceptions to which are linked A and B (cf fig. 2)

We consider that, for a simple hierarchical relation (when there is only one way to go from A and B) the relation semantic value is given by the product of the RSV which constitute the path. We can compare it as independant events in probability. The precedent relation is more general and consider that several path could exit between A to B. In this case, we choose the worst path (the least probable), thus the evaluated RSV is:

In this category, there are two pairs of relations: hyperonymy/hyponymy and meronymy/holonymy.

hyperonymy and hyponymy *The hyponymy relation is a hierarchical relation which links hyponym to a more general item, the hyperonym.* As an exemple, we have ‹*car*› as hyponym of ‹*vehicle*› and ‹*vehicle*› as hyperonym of ‹*car*›. We can say that A hyponym is a *kind of* B hyponym. Acceptions can be either hyponym

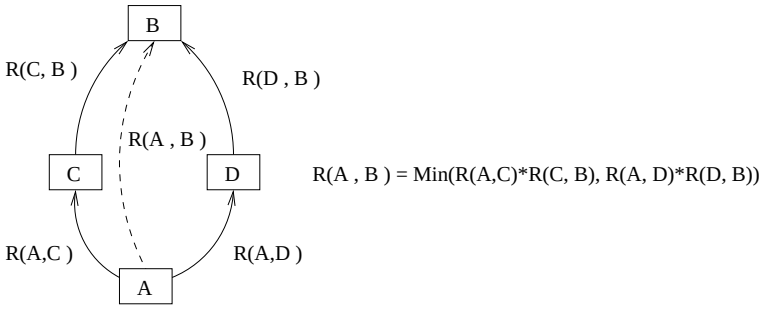

Fig. 2. Valued Semantic Relation on Hierarchies

and hyperonym: ‹*seat*› is the hyponym of ‹*furniture*› and the hyperonym of chair. The figure 3 shows an example of hyperonymy/hyponymy hierarchy centered around *seat*.

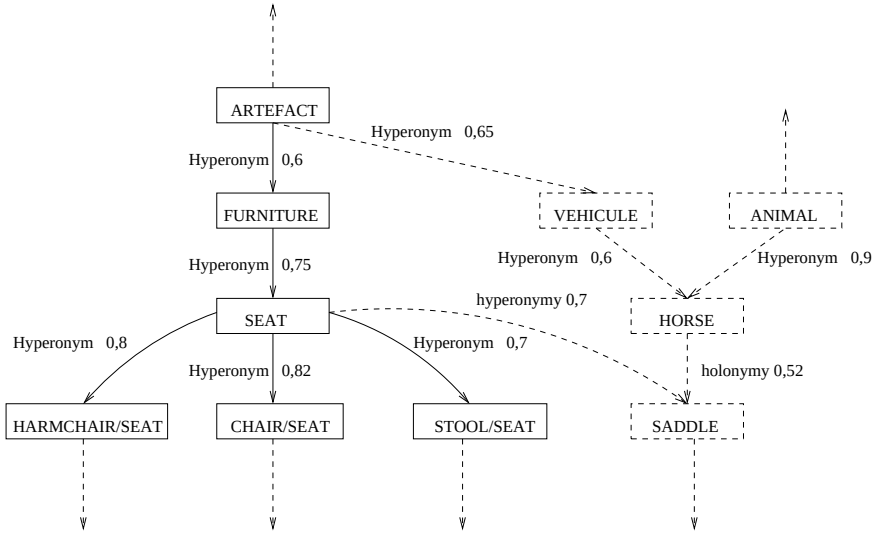

Fig. 3. hyponymy/hyperonymy relation

The part-whole relation : meronymy and holonymy The part-whole relation is *the hierachical relation that holds between two terms. One, the meronym, is a part and the other, the holonym, the whole* : ‹*sail*›/‹*boat*›, ‹*arm*›/‹*body*›, ‹*nail*›/‹*finger*›, ... We have, ‹*nail*› as the *meronym* of ‹*finger*› and ‹*finger*› as the holonym of ‹*nail*›. We can say that *A* meronym is a *part of B* holonym. As hyperonym/hyponym, units can be either meronym and holonym: ‹*arm*› is the holonym

of ‹*finger*› and the meronym of ‹*body*›. The figure 4 shows an example of meronymy/ holonymy hierarchy centred around *body*. The right dashed part shows that the semantics relations can be mixed.

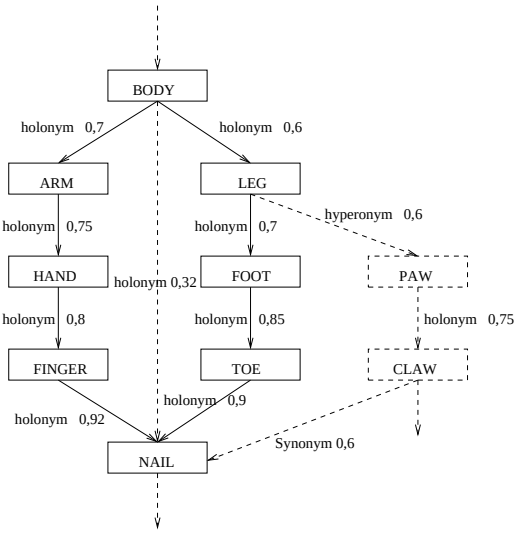

Fig. 4. meronymy/holonymy and others relations

4.4 Symmetric relations.

Synonymy This is the semantic relation that holds between two lexical units that could, in a given context, express the same meaning. For example, ‹*airplane*› and ‹*aeroplane*› are synonyms.

Contrary to lexical units, acceptions are monosemic by definition. In this context, we can define synonymy as *the semantic relation that holds between two acceptions that express the same meaning.*

A non-expert agent can evaluate the Synonymy between two acceptions with the following formula:

$$Syn(A, C) = Min_{i \in I}(Min(Syn(A, X_i), Syn(X_i, B)))$$

where I is the set of the items to whom are linked to A and B

If there is a path between A and B, we consider that the RSV between A and B is the less RSV of the path. When there are several paths between A and B, the same idea as for hierarchical relation is used, we choose the worst path (the least probable) to evaluate RSV.

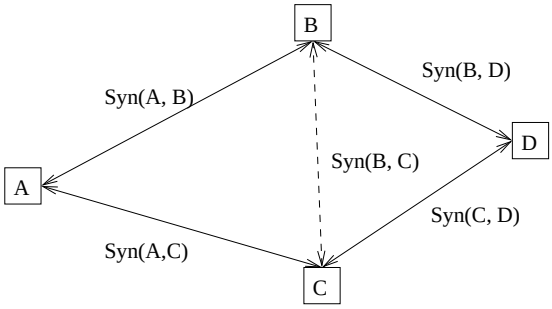

Fig. 5. Evaluation of the Synonymy Relation

$$Syn(B,C) = Min(Min(Syn(A,B)Syn(A,C)), Min(Syn(B,D), Syn(C,D)))$$

Antonymy We proposed in [*Schwab and al*, 2002] a definition of antonymy compatible with the vectorial model used. *Two lexical items are in antonymy relation if there is a symmetry between their semantic components relatively to an axis.* For us, antonym construction depends on the type of the medium that supports symmetry. For a term, either we can have several kind of antonyms if several possibilities for symmetry exist, or we cannot have an obvious one if a medium for symmetry is not to be found. From the point of view of semantic relations, if we compare synonymy and antonymy, we can say that synonymy is the search of resemblance with the test of substitution (x is synonym of y if x may replace y), antonymy is the research of the symmetry, that comes down to investigating the existence and nature of the symmetry medium. We have identified three type of symmetry by relying on [*Lyons*, 1977], [*Palmer, 1976*] and [*Muehleisen*, 1997] : complementary, scalar and dual.

The **complementary antonymy** Ant_c holds couples like *event/unevent, existence/ non-existence, presence/absence*. It corresponds to the exclusive disjonction relation. In this frame, the affirmation of one of the term implies obligatory the negation of the other. The complementary antonymy presents two kind of symmetry : a value symmetry in a boolean system, as in the examples above a symmetry about an proporty application : *black* is colorness, so it is "opposed" to all other colors or colors combinaison.

The **scalar antonymy** Ant_s concerns scaled values. They have a symmetry relatively to a reference value which is not always represented by a unit. We can find in this category antonyms like *cold/hot* or *small/tall*. In these antonymy, the two opposites can be not verified. This man is "neither tall nor small" indicates generally a medium height. It doesn't indicate that the property doesn't apply like *alive/dead*. There is, in this case, a neutral value from which all other value refer to. Here, symmetry is done in relation to these reference value.

The last, the **dual antonymy** Ant_d, is divided in two sub-families, the **conversive duals** and the **contrastive duals**. In this type, symmetry comes from the use and nature of the objects themselves. The conversive duals (or recip-

rocals) are couples like *buy/sell, husband/wife, father/son*. If*Jack is John's son* then *John is Jack's father*. In the case of the conversives, if we replace, in a sentence, a term A by his reciprocal B, we can restore the synonymy between the two sentences if we permute the syntaxic arguments related by A. So, for the conversives, there is a symmetry relatively to the situation of the arguments. The contrastives duals are introduced to take care of a particular effect of relationship between terms. The symmetry is about culturals (consecrated by the usage) or spatio-temporal functions. The contrastives are culturaly associated units like *sun/moon*, units which come intuitively together like *question/answer* or are the expression of the passage from a state to another like *birth/death*. The symmetry is that if one of the predicate is true, there is a value for which the other is true too.

Items without antonyms Some items cannot have an antonym. For instance, it is the case of material objects like *car, bottle, boat, etc.* The question that raises is about the continuity the antonymy functions in the vector space. How to define the antonym of a word which doesn't have an antonym? We could either consider $\mathbf{0}$, the null vector, as the *default* antonym or consider that such word is a fixed point of the function. In other terms, we assume that the antonym of a word without antonym is the word itself. The first approach doesn't seem to be relevant. From a linguistic point of view , it is equivalent to consider that the opposite of a non-opposable word is the empty idea. In fact, if we want to compute the antonym of a ‹*motorcycle*›, which is a ROAD TRANSPORT, NOISY and FAST, we don't want to have a ‹*turtle*›, a ‹*slug*› or anything SILENCIOUS and SLOW but rather a ROAD TRANSPORT, (SILENCIOUS and SLOW), something like a ‹*bicycle*› or an ‹*electric car*›. With this method, fixed points can be considered on the symmetry axis which is compatible with our general theory. In the following, we will make no distinction between a lexical item without an antonym and a lexical item which is its own antonym.

5 Antonymy function

5.1 Antonym vectors of concept lists

Anti functions are context-dependent and cannot be free of concepts organisation. They need to identify for every concept and for every kind of antonymy, a vector considered as the opposite. We had to build a list of triples $\langle concept, context, vector \rangle$. This list is called *antonym vectors of concept list* (AVC).

AVC construction. The Antonym Vectors of Concepts list is manually built only for the conceptual vectors of the generating set. For any concept we can have the antonym vectors such as:

$AntiC(\textsc{existence}, V) = V(\textsc{non-existence})\ \forall V$
$AntiC(\textsc{non-existence}, V) = V(\textsc{existence})\ \forall V$
$AntiC(\textsc{agitation}, V) = V(\textsc{inertia}) \oplus V(\textsc{rest})\ \forall V$
$AntiC(\textsc{play}, V) = V(\textsc{play})\ \forall V$
$AntiC(\textsc{order}, V(order) \oplus V(disorder)) = V(\textsc{disorder})$
$AntiC(\textsc{order}, V(classification) \oplus V(order)) = V(\textsc{classification})$

As items, concepts can have, according to the context, a different opposite vector even if they are not polysemic. For instance, *destruction* can have for antonym *preservation, construction, reparation* or *protection*. So we have defined for each one, one conceptual vector which will allow the selection of the best antonym according to the situation. Also, the concept *existence* has the vector *non-existence* for antonym for any context. The concept *disorder* has the vector of *order* for antonym in a context constituted by the vectors of *order* $\oplus$*disorder*[5] and has *classification* in a context constituted by *classification* and *order*.

The function $AntiC(C_i, V_{context})$ returns for a given concept C_i and the context defined by $V_{context}$, the complementary antonym vector in the list.

5.2 Construction of the antonym vector: the *Anti* Function

Definitions We define the relative antonymy function $Anti_R(A, C)$ which returns the opposite vector of A in the context C and the absolute antonymy function $Anti_A(A) = Anti_R(A, A)$. The usage of $Anti_A$ is delicate because the lexical item is considered as being its own context. We don't have this problem for acceptions which are monosemic. We should stress now on the construction of the antonym vector from two conceptual vectors: V_{item}, for the item we want to oppose and the other, V_c, for the context (referent).

Construction of the Antonym Vector The method is to focus on the salient notions in V_{item} and V_c. If these notions can be opposed then the antonym should have the inverse ideas in the same proportion. That leads us to define this function as follows:

$$Anti_R(V_{item}, V_c) = \bigoplus_{i=1}^{N} P_i \times AntiC(C_i, V_c)$$

$$\text{with} \quad P_i = V_{item_i}^{1+CV(V_{item})} \times max(V_{item_i}, V_{c_i})$$

We crafted the definition of the weight P after several experiments. We noticed that the function could not be symmetric (we cannot reasonably have $Anti_R(\mathrm{V}(\text{‹}hot\text{›}),\mathrm{V}(\text{‹}temperature\text{›})) = Anti_R(\mathrm{V}(\text{‹}temperature\text{›}),\mathrm{V}(\text{‹}hot\text{›}))$). That is why we introduce this power, to stress more on the ideas present in the vector we want to oppose. We note also that the more conceptual[6] the vector is,the more

[5] $\oplus$ is the normalised sum $V = A \oplus B \mid v_i = \frac{x_i + y_i}{\|V\|}$

[6] In this paragraph, conceptual means: *closeness of a vector to a concept*

important this power should be. That is why the power is the variation coefficient[7] which is a good clue for "conceptuality". To finish, we introduce this function *max* because an idea present in the item, even if this idea is not present in the referent, has to be opposed in the antonym. For example, if we want the antonym of ‹*cold*› in the ‹*temperature*› context, the weight of ‹*cold*› has to be important even if it is not present in ‹*temperature*›.

5.3 Antonymy Evaluation Measure

It seems relevant to assess wether two lexical items can be antonyms. To give an answer to this question, we have created a measure of antonymy evaluation. Let A and B be two vectors. The question is precisely to know if they can reasonably be antonyms in the context of C. The antonymy measure $Manti_{Eval}$ is the angle between the sum of A and B and the sum of $Anti_{c_R}(A, C)$ and $Anti_{c_R}(B, C)$. Thus, we have:

$$Manti_{Eval} = D_A(A \oplus B, Anti_R(A, C) \oplus Anti_R(B, C))$$

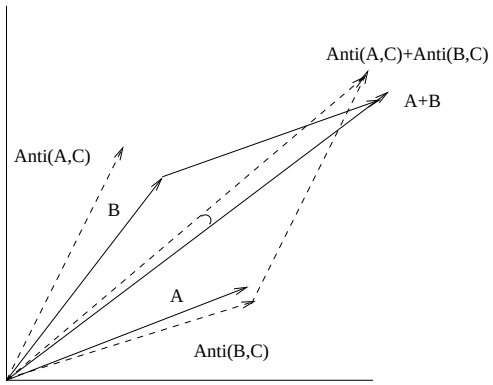

Fig. 6. 2D geometric representation of the antonymy evaluation measure $Manti_{Eval}$

The antonymy measure is a pseudo-distance. It verifies the properties of reflexivity, symmetry and triangular inequality only for the subset of items which doesn't accept antonyms. In this case, nonewithstanding the noise level, the measure is equal to the angular distance. In the general case, it doesn't verify reflexivity. The conceptual vector components are positive and we have the property: $Dist_{anti} \in [0, \frac{\pi}{2}]$. The smaller the measure, the more 'antonyms' the two lexical items are. However, it would be a mistake to consider that two synonyms

[7] The variation coefficient is $\frac{SD(V)}{\mu(V)}$ with SD as the standart deviation and μ as the arithmetic mean.

would be at a distance of about $\frac{\pi}{2}$. Two lexical items at $\frac{\pi}{2}$ have not much in common[8]. We would rather see here the illustration that two antonyms share some ideas, specifically those which are not opposable or those which are opposable with a strong activation. Only specific activated concepts would participate in the opposition. A distance of $\frac{\pi}{2}$ between two items should rather be interpreted as these two items do not share much idea, a kind of *anti-synonymy*. This result confirms the fact that antonymy is not the exact inverse of synonymy but looks more like a 'negative synonymy' where items remains quite related. To sum up, the antonym of w is not a word that doesn't share ideas with w, but a word that opposes some features of w.

Examples In the following examples, the context has been ommited for clarity sake. In these cases, the context is the sum of the vectors of the two items.

$$\begin{aligned}
Manti_{Eval}(\textsc{existence},\textsc{non-existence}) &= 0.03\\
Manti_{EvalC}(\text{‹}\textit{existence}\text{›}, \text{‹}\textit{non-existence}\text{›}) &= 0.44\\
Manti_{EvalC}(\textsc{existence}, \textsc{car}) &= 1.45\\
Manti_{EvalC}(\text{‹}\textit{existence}\text{›}, \text{‹}\textit{car}\text{›}) &= 1.06\\
Manti_{EvalC}(\textsc{car}, \textsc{car}) &= 0.006\\
Manti_{EvalC}(\text{‹}\textit{car}\text{›}, \text{‹}\textit{car}\text{›}) &= 0.407
\end{aligned}$$

The above examples confirm what presented. Concepts *existence* and *non-existence* are very strong antonyms in complementary antonymy. The effects of the polysemy may explain that the lexical items ‹*existence*› and ‹*non-existence*› are less antonyms than their related concepts. In complementary antonymy, *car* is its own antonym. The antonymy measure between *car* and *existence* is an example of our previous remark about vectors sharing few ideas and that around $\pi/2$ this measure is close to the angular distance (we have $D_A(existence, car) = 1.464$.).

Valued Semantic Relation of Antonymy We define the valued semantic relation of antonymy as:

$$Ant_i(X,Y) = 1 - \frac{2}{\pi} Manti_{Eval_i}(X,Y) \quad i \in \{c, s, d\}$$

It is the conversion from $[0, \frac{\pi}{2}]$ to $[1, 0]$.

6 Hardening of Base Integrity

6.1 Links between Synonymy and antonymy

For acceptions, because of their quasi-monosemy, antonymy and synonymy are contrary relationship and cannot be found together. Thus, for any kind of antonymy:

[8] This case is mostly theorical, as there is no language where two lexical items are without any possible relation.

$$A\ Syn\ B\ \Rightarrow \neg(A\ Ant\ B)$$

$$A\ Ant\ B\ \Rightarrow \neg(A\ Syn\ B)$$

If A and B are two antonyms, they can't be synonym.

General schema of coherence From the previous two relations, we can deduce a general schema which must to be verified in the acceptance base (fig. 7) otherwise the base is not coherent.

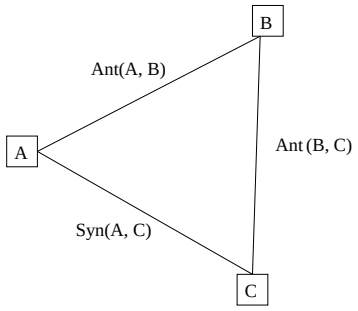

Fig. 7. General schema to evaluate Antonymy and Synonymy

It shows that if we have A as antonym of B and B as antonym of C, then we must have A and B synonym. It shows too that if A and B are antonym and A and C synonym then B and C are Antonym. If one of the relations is the opposite (i.e. antonymy replace synonymy or synonymy replace antonymy) or not mateiralised, the base is not coherent. Specialist agents search these triangular relations and emit warning if they locate an incoherence. The lexicograph, helped by monolingual dictionnaries indicate if two acceptions need to be splited or if one link should't be materialised.

Evaluation of synonymy and antonymy The general schema of coherence (fig. 7) can also help non-specialist agents to evaluate a non-materialised link. If A is antonym of B and B is antonym of C, then we have A and B synonym. In general case, we have

$$Syn(A, C) = Min_i(Min(Ant(A, X_i), Ant(X_i, C)))$$

The figure 8 shows an exemple of evaluation.
By a similar way, we can also evaluate antonymy:

$$Ant_i(A, C) = Min_i(Min(Syn(A, X_i), Ant(X_i, C)))$$

The figure 9 is an exemple of antonymy evaluation.

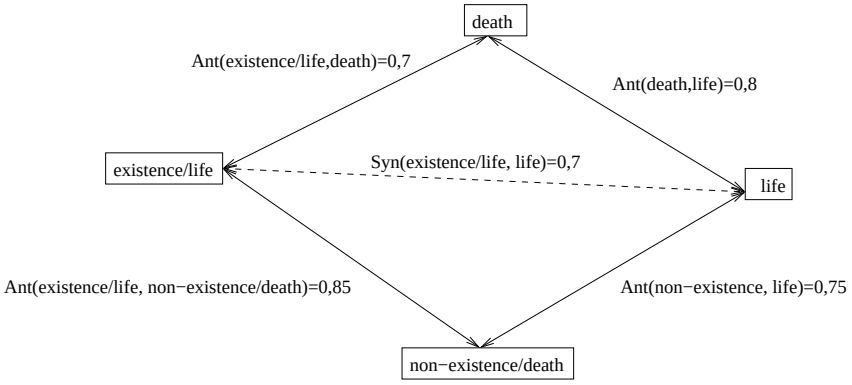

Fig. 8. Exemple of Synonymy evaluation:

$$Syn(\text{existence/life, life}) = Min($$

$$(Min(Ant(\text{existence/life, non-existence/death}), Ant(non-existence/death, life)),$$
$$Min(Ant(\text{existence/life, death}), Ant(\text{death, life})))$$

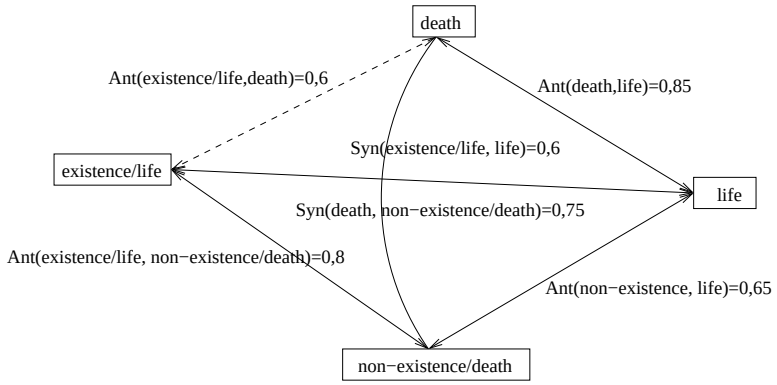

Fig. 9. Exemple of Antonymy evaluation:

$$Ant(\text{existence/life, death}) = Min($$

$$Min(Syn(\text{existence/life, life}, Ant(\text{death, life}),$$
$$Min(Syn(\text{existence/life, life}), Ant(\text{existence/life, non-existence/death})))$$

7 Conclusion

In this paper, we have presented a way to improve the integrity of an acception base through well known semantic relations like synonymy, antonymy, hyperonymy and holonymy. We have presented the values semantics relations (VSR) which can be comparated to the probability that a relation exists between two items or acceptions. These semantics relations values are computed by specialist agents thanks to various lexical informations and conceptual vectors to create materialised links between two acceptions if the SRV is above a threshold. Base integrity agents walk throw the acceptions and look for incoherences in the base and emit warning toward lexicographs when neened. We have shown how non-specialist agents (which are in charge of lexical transfert or word sense desambiguation) can evaluate non-materialised links from materialised ones.

This is a preliminary work, we have to study in wich cases base integrity agents could find by themself what to do (split acception, definition revision,...) if they find an incoherence in the acception base.

References

[*Chauché*, 90] Jacques Chauché, *Détermination sémantique en analyse structurelle : une expérience basée sur une définition de distance.* TAL Information, 31/1, pp 17-24, 1990.

[*Deerwester et al*, 90] Deerwester S. et S. Dumais, T. Landauer, G. Furnas, R. Harshman, *Indexing by latent semantic anlysis.* In Journal of the American Society of Information science, 1990, 416(6), pp 391-407.

[*Lafourcade et Prince*, 2001] Lafourcade M. et V. Prince *Synonymies et vecteurs conceptuels.* Proc. of Traitement Automatique du Langages Naturel (TALN'2001) (Tours, France, Juillet 2001), pp 233-242.

[*Lafourcade*, 2001] Lafourcade M . *Lexical sorting and lexical transfer by conceptual vectors.* Proc. of the First International Workshop on MultiMedia Annotation (Tokyo, Janvier 2001), 6 p.

[*Larousse*, 2001] Larousse. *Le Petit Larousse Illustré 2001.* Larousse, 2001.

[*Larousse*, 1992] Larousse. *Thésaurus Larousse - des idées aux mots, des mots aux idées.* Larousse, ISBN 2-03-320-148-1, 1992.

[*Lehmann et Martin-Berthet*, 98] Lehmann A. et Martin-Berthet F. *Introduction à la lexicologie. Sémantique et morphologie.* Paris, Dunod (Lettres Sup), 1998.

[*Lyons*, 1977] Lyons J. *Semantics.* Cambridge : Cambridge University Press, 1977.

[*Mangeot*, 2001] . Mangeot-Lerebours M. *Environnements centralisés et distribués pour lexicographes et lexicologues en contexte multilingues.* PhD thesis,Université Joseph Fourier, 2001.

[*Mel'čuk and al*, 95] Mel'čuk I., Clas A.et Polguère A. *Introduction à la lexicologie explicative et combinatoire.* Éditions Duculot, 1995.

[*Morin*, 1999] Morin, E. *Extraction de liens sémantiques entre termes à partir de corpus techniques.* Thèse de doctorat de l'Université de Nantes, 1999.

[*Muehleisen*, 1997] Muehleisen V.L. *Antonymy and semantic range in english.* Northwestern university Phd, 1997.

[*Palmer, 1976*] Palmer, F.R. *Semantics : a new introduction.* Cambridge University Press, 1976.

[*Polguère, 2001*] Polguére A. *Notions de base en lexicologie.* Observatoire de linguistique sens-texte, 2001.

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

[*Rodget*, 1852] *Thesaurus of English Words and Phrases.* Longman, London, 1852.

[*Salton et MacGill*, 1983] Salton G. et MacGill M.J. *Introduction to modern Information Retrieval.* McGraw-Hill, New-York, 1983.

[*Schwab*, 2001] Schwab D. *Vecteurs conceptuels et fonctions lexicales : application à l'antonymie.* Mémoire de DEA, Universitè Montpellier 2, LIRMM, 62 p, 2001.

[*Schwab and al*, 2002] Schwab D., Lafourcade M. et Prince V. *Antonymy and Conceptual Vectors.* COLING 2002 processings to apear, 8 p.

[*Sérasset and Mangeot*, 2001] Sérasset G., Mangeot M. *Papillon lexical databases project: monolingual dictionnaries & interlingual links.* NLPRS 2001 processings, pp 119-125.