Mixing Semantic Networks and Conceptual Vectors: the Case of Hyperonymy

Violaine Prince
LIRMM-CNRS and University Montpellier 2
161 Ada Street, 34392 Montpellier cedex 5
France
prince@lirmm.fr

Mathieu Lafourcade
LIRMM-CNRS and University Montpellier 2
161 Ada Street, 34392 Montpellier cedex 5
France
lafourca@lirmm.fr

Abstract

In this paper, we focus on lexical semantics, a key issue in Natural Language Processing (NLP) that tends to converge with conceptual Knowledge Representation (KR) and ontologies. When ontological representation is needed, hyperonymy, the closest approximation to the is-a *relation, is at stake. In this paper we describe the principles of our vector model (CVM: Conceptual Vector Model), and show how to account for hyperonymy within the vector-based frame for semantics. We show how hyperonymy diverges from* is-a *and what measures are more accurate for hyperonymy representation. Our demonstration results in initiating a 'cooperation' process between semantic networks and conceptual vectors. Text automatic rewriting or enhancing, ontology mapping with natural language expressions, are examples of applications that can be derived from the functions we define in this paper.* ***Keywords****: knowledge representation, cognitive linguistics, natural language processing.*

1 Introduction

Lexical semantics are a key issue in Natural Language Processing (NLP) since they represent the point of convergence with conceptual Knowledge Representation (KR) and ontologies. They also browse the area of lexical resources processing, so that many works in both NLP and AI have been devoted to lexical semantic functions, as a way to tackle the problem of word sense representation and discrimination. Among the well established trends in lexical semantics representations, two trends seem to be conflictual: the WordNet approach [11], [4], born from semantic networks, and KR-oriented, and the Vector approach, originated from the Saltonian representation in Information Retrieval [17], which has found a set of applications in NLP. The first is based on logic and the second on vector-space algebra.

The first is very efficient for *is-a* relationships (considered as the conceptual relation often embedded in hyperonymy) but is silent, or almost so, about several other interesting lexical functions such as antonymy and thematic association. Synonymy has been tackled [19], [13], but discrimination between synonymy and hyperonymy has often led researchers to look for a more flexible notion such as semantic similarity [14]. The vector approach is completely at the at the opposite. Offering very easily thematic association, it allows many fine-grained synonymy [7] and antonymy [20] functions to be defined and implemented, but is unable to differentiate or to valid the existence of hyperonymous relations.

In this paper we show how to account for hyperonymy within the vector-based frame for semantics, relying on a cooperation between semantic networks and conceptual vectors, and how this can be applied to new functions such as word substitution, and semantic approximation, that belong to the field of semantic similarity. We use a semantic network to enhance vector learning, and symmetrically we build customized semantic networks out of hyperonymous relations between vectors.

2 Hyperonymy and *is-a* relations

2.1 Defining Hyperonymy

Hyperonymy is a lexical function that, given a term t, associates to t one or many other terms that are more general, such as those used to define t in *genus* and *differentiae* (aristotelian definition). Its symmetrical function is called **hyponymy**.

Hyperonymy, in almost all KR papers, is assimilated to the general argument of the *is-a* relationship (fundamentals are given in [1]). Let us remind that the *is-a* relationship is such as if X is a class of objects, and X' a subclass of X, then $is-a(X', X)$ is true. The rightmost argument X is called the *general* argument whereas X' is said to be the *specific* argument. The problem is that linguistic hyperonymy is not a "pure" *is-a* relation. When the word *horse* is defined, we find: ”a herbivorous animal, with four legs, etc. . . ”. A good hyperonym for this definition of *horse* is *herbivorous mammal*. *Animal* is another hyperonym, since ’*herbivorous mammal is-a mammal* and *mammalis-a animal*’ is true. However, thematically, a *horse* is very close to a *herbivore*, whereas *herbivores* do not constitute a class but a set of individuals that may belong to different lines of the taxonomy (birds and insects and reptiles could be herbivorous, but also metaphorically, many other things). Thus, even if, in language, one wants to write that *a horse is a herbivore* eventhough *horse is-a herbivore* is false.

2.2 WordNet and Hyperonymy

WordNet is a built taxonomy of words, and as such, only captures *is-a* relations. A hyperonym is a linguistic superordinate, generally used in definitions that also captures particular properties that cannot act as classes by themselves. Polysemous words have many definitions, and thus many hyperonyms: a *horse* is also a *vehicle*, that is, a mean of transportation.This implies many *is-a* relations, which explains why WordNet is a network and not a tree. The only constraint in language is that a hyperonym needs to be more general (and thus *herbivore* could act as a hyperonym for *horse*) whereas in a semantic network, every step of the chain of classes and subclasses must verify the order relation.

2.3 Hyperonymy and Word Definition

As shown before, hyperonyms could be extracted, when they are not known, from most dictionary like definitions. Only general concepts, which tend to play the role of hyperonyms (and *is-a*) superclasses of many others, are not defined through aristotelian definition, but are explained by their hyponyms. This is why, in our CVM (Conceptual Vector Model) model presented in next section, we consider the existence of a ”hyperonymy horizon” beyond which definitions become inversed: hyperonyms are more difficult to find and less explicative than hyponyms. The word *action* is almost at the top of the WordNet taxonomy and dictionary definitions tend to explain it with more specific words.

3 The Conceptual Vector Model (CVM)

Vectors have been used in Information Retrieval for long [18] and for meaning representation by the LSI model [3] from latent semantic analysis (LSA) studies in psycholinguistics. In NLP, [2] 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 that considers a set of concepts as a generator to language has been long described in [16] (thesaurus hypothesis) and has been used by researchers in NLP (e.g. [21]). Polysemous words combine different vectors corresponding to different meanings. This vector approach is based on well known mathematical properties: it is thus possible to undertake formal manipulations attached to reasonable linguistic interpretations. Concepts are defined from a thesaurus (in our prototype applied to French, we have chosen [8] where 873 concepts are identified to compare with the thousand defined in [16]). To be consistent with the thesaurus hypothesis, we consider that this set constitutes a generator space for 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.

3.1 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 set of pairs (*CONCEPT*[intensity]) are ordered by increasing values): V(‹*door*›) = (*OPENING*[0.3], *BARRIER*[0.31], *LIMIT*[0.32], *PROXIMITY*[0.33], *EXTERIORR*[0.35], *INTERIOR*[0.37], . . .

In practice, the largest $\mathcal{C}$ is, the finer the meaning descriptions are. In return, the computer manipulation is less easy. As most vectors are dense (very few null coordinates), the enumeration of activated concepts is long and difficult to evaluate. We generally prefer to select the thematically closest terms, i.e., the *neighbourhood*. For instance, the closest terms ordered by increasing distance of ‹*door*› are: $\mathcal{V}$(‹*door*›)=‹*portal*›, ‹*portiere*›, ‹*opening*›, ‹*gate*›, ‹*barrier*›,. . .

To handle semantics within this vector frame, we use the common operations on vectors. An interesting measure is the angular distance that accounts for a similarity measure. As an example, we present, hereafter, the vector sum , the scalar product and the angular distance equations.

3.1.1 Vectors Sum

Let A and B be two vectors, we define V as their normed sum:

$$V = X \oplus Y \quad | \quad v_i = (x_i + y_i)/\|V\| \tag{1}$$

Intuitively, the vector sum of A and B corresponds to the union of semantic properties of A and B. This operator is idempotent as we have $A \oplus A = A$. The null vector $\vec{\mathbf{0}}$ is

[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 sometimes use *word* or *term* to speak about a *lexical item*.

a neutral element of the vector sum and, by definition, we have $\vec{\mathbf{0}} \oplus \vec{\mathbf{0}} = \vec{\mathbf{0}}$.

3.1.2 Vectors Product

The vector product is equivalent to a *normed term to term product*. Let X and Y be two vectors, we define V as *their normed term to term product*:

$$V = X \otimes Y \quad | \quad v_i = \sqrt{x_i y_i} \tag{2}$$

This operator is idempotent and $\vec{\mathbf{0}}$ is absorbent.

$$V = X \otimes X = X \quad \text{and} \quad V = X \otimes \vec{\mathbf{0}} = \vec{\mathbf{0}} \tag{3}$$

Intuitively, the vector product of A and B corresponds to the intersection of semantic properties of A and B. This is a crucial feature for hyperonymy since a hyperonym and its hyponym could be seen as one containing the properties of the other. But it is also important in synonymy and may give hints about polysemous properties of some conceptual vectors (intersections with many different vectors). A better function for emphasizing intersection is given in the paragraph about contextualisation.

3.1.3 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. We can express this function as:

$$Sim(A, B) = \cos(\widehat{A, B}) = \frac{A \cdot B}{\|A\| \times \|B\|}$$

with "$\cdot$" 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 follows:

$$\begin{aligned} & D_A(A, B) = \arccos(Sim(A, B)) \\ \text{with} \quad & Sim(A, B) = \cos(\widehat{A, B}) = \frac{A \cdot B}{\|A\| \times \|B\|} \end{aligned} \tag{4}$$

Intuitively, this function constitutes an evaluation of *thematic proximity* and is the measure of the angle between the two vectors. We would generally consider that, for a distance $D_A(A, B) \leq \frac{\pi}{4}$, (i.e. less than 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. In the following, we will speak of *distance* only when these last properties will be verified, otherwise we will speak of *measure*.

3.1.4 Contextualisation

When two terms are in presence of each other, some of the meanings of each of them are thus selected by the presence of the other, acting as a **context**. This phenomenon is called *contextualisation*. It consists in emphasizing common features of every meaning. Let X and Y be two vectors, we define $\gamma(X, Y)$ as the contextualisation of X by Y as:

$$\gamma(X, Y) = X \oplus (X \otimes Y) \tag{5}$$

These functions are not symmetrical. The operator γ is idempotent ($\gamma(X, X) = X$) and the null vector is the neutral element. ($\gamma(X, \vec{\mathbf{0}}) = X \oplus \vec{\mathbf{0}} = X$). We will notice, without demonstration, that we have thus the following properties of *closeness* and of *farness*:

$$\begin{aligned} & D_A(\gamma(X, Y), \gamma(Y, X)) \\ \leq \{ & D_A(X, \gamma(Y, X)), D_A(\gamma(X, Y), Y)\} \\ & \leq D_A(X, Y) \end{aligned} \tag{6}$$

The function $\gamma(X, Y)$ brings the vector X closer to Y proportionally to their intersection. The contextualization is a low-cost meaning of amplifying properties that are salient in a given context. For a polysemous word vector, if the context vector is relevant, one of the possible meanings is *activated* through contextualization. For example, *bank* by itself is ambiguous and its vector is pointing somewhere between those of *river bank* and *money institution*. If the vector of *bank* is contextualized by *river*, then concepts related to finance would considerably dim.

3.2 Implemented Lexical Functions: Synonymy and Antonymy

3.2.1 Synonymy

Two lexical items are in a synonymy relation if there is a semantic equivalence between them.
Synonymy is a pivot relation in NLP, but remains problematic, since semantic equivalence is not translatable into an equivalence relationship. It does not necessarily verify transitivity [10] and it could be, at least partially, confused with hyperonymy, when equivalence is reduced to semantic similarity [14]. A possible solution in a vector framework is to define a contextual synonymy (also proposed in [5]) represented by a three argument relation, which then supports the properties of an equivalence relationship. The suggested solution is called relative synonymy [7]. The functional representation is the following: We define the *relative synonymy* functionSyn_R, between three vectors A, B and C, the later playing the role of a pivot, as:

$$\begin{aligned} Syn_R(A, B, C) &= D_A(\gamma(A, C), \gamma(B, C)) \\ &= D_A(A \oplus (A \otimes C), B \oplus (B \otimes C)) \end{aligned} \tag{7}$$

The interpretation corresponds to testing the thematic closeness of two meanings (A and B), each one enhanced with what it has in common with a third (C). The advantage of such a solution is that it circumvents the effects of polysemy in cutting transitivity and symmetry. However, it does not provide a real distinction between a hyperonym of a given meaning of a word, and a true synonym of such a word. This problem is discussed in next section, when introducing more flexible notions such as **word substitution**.

3.2.2 Antonymy

Two lexical items are in antonymy relation if there is a symmetry between their semantic components relatively to an axis.
Three types of symmetry have been defined, inspired from linguistic research [12]. As an example, we expose only the 'complementary' antonymy proposed by [20]: The same method is used for the other types. *Complementary antonyms* are couples like *event/unevent, presence/absence*. Complementary antonymy presents two kinds of symmetry, (i) a value symmetry in a boolean system, as in the examples above, and (ii) a symmetry about the application of a property (*black* is the absence of color, so it is "opposed" to all other colors or color combinaisons). The functional representation is the following: The function $AntiLex_S$ returns the *n* closest antonyms of A in the context defined by C in reference to R. The partial function $AntiLex_R$ has been defined to take care of the fact that, in most cases, context is enough to determine a symmetry axis. $AntiLex_B$ is defined to yield a symmetry axis rather than a context. In practice, we have $AntiLex_B = AntiLex_R$. The last function is the *absolute antonymy function*. Their associated equations are given hereafter.

$$A, C, R, n \rightarrow AntiLex_S(A, C, R, n) \quad (8)$$

$$A, C, n \rightarrow AntiLex_R(A, C, n) = AntiLex_S(A, C, C, n) \quad (9)$$

$$A, R, n \rightarrow AntiLex_B(A, R, n) = AntiLex_S(A, R, R, n) \quad (10)$$

$$A, n \rightarrow AntiLex_A(A, n) = AntiLex_S(A, A, A, n) \quad (11)$$

An implementation of these functions in the CVM is detailed and commented in [20].

3.3 Conceptual Vectors Construction

Building conceptual vectors is achieved throug processing **definitions** from different sources (dictionaries, synonym lists, manual indexations, etc). Definitions are parsed with an NLP parser called SYGMART and the corresponding conceptual vector is computed according to a procedure defined as follows.

After filtering according to various morphosyntactic attributes, we attach to the leaf (terminal node of the conceptual tree) a conceptual vector that is computed from the vectors of its k definitions. The most straighforward way (not the best) to do so, is to compute the average vector: $V(w) = V(w.1) \oplus \cdots \oplus V(w.k)$. If the word is unknown (i.e. it is not in the dictionary), the null vector is taken instead.

The vectors are then propagated upward. Consider a tree node N with p dependands $N_i(1 \leq ip)$. The newly computed vector of N is the weighted sum of all vectors of N_i: $V(N) = \alpha_i N_1 \oplus \cdots \oplus \alpha_p N_p$. The weights α depend of the syntactic functions of the node. For instance, a governor would be given a higher weight ($\alpha = 2$) than a regular node ($\alpha = 1$).The vectors computed for *a boat sail* and for *a sail boat* would not be identical. Once the vector of the tree root is computed a downward propagation is performed. A node vector is contextualized by its parent: $V'(N_i) = V(N_i) \oplus \gamma(N_i, N)$. This is done iteratively until reaching a leaf. This analysis method shapes, from existing conceptual vectors and definitions, new vectors. It requires a bootstrap with a kernel composed of pre-computed vectors, manually indexed for the most frequent or difficult terms and already defined in [8] . One way to build an coherent learning system is to take care of the semantic relations beetween items, and among them, synonymy, antonymy and the most important, hyperonymy. A relevant conceptual vector basis is obtained after some iterations in the learning process. At the moment of writing this article, our system counts more than 71000 items for French and more than 288000 vectors. 2000 vectors are concerned with antonymy, almost all of them are concerned with synonymy and hyperonymy. The computed functions have allowed to enhance the representation of almost all vectors.

3.4 Importance of Hyperonymy in CVM

A framework for hyperonymy is very useful for enhancing vector construction, since most vectors are built by parsing hyperonymous definitions provided by on-line sources on the Web. In fact, all lexical functions appear to be a great help for such as task. Symmetrically, relations between vectors are crucial for a data driven approach : trying to extract semantic relations in corpora ([21]) and thus building a domain ontology, or trying to organize information in corpora by relying upon *is-a* hierarchies ([9], [15]).

4 Computing Hyperonymy

As our approach is both data driven and hierarchy-based, we first try to define the impact of hyperonymy by measuring distances in corpora. These distances help to define word substitution and semantic approximation (with a taxonomical aspect). The theoretical model, both within semantic networks and vector space, is the **inclusion model**: a subclass includes the properties of its superclass. We show in this section how inclusion is dealt with and what results we have obtained.

4.1 Co-occurrence Model

We define two measures between a term w and an **hyperonym candidate** h:

$$M_T(w, h) = \frac{H \cap W}{W} \quad \text{and} \quad M_S(w, h) = \frac{H \cap W}{H} \quad (12)$$

W (resp. H) represents the number of documents in a given corpus that contains the term w (resp. h). The value $H \cap W$ represents the number of documents that contains both terms h and w. M_T is reminiscent of the *recall* and M_S of the *precision* in Information Retrieval.

4.1.1 Hyperonymy, Word Substitution, Semantic Approximation: Building a local possible *is-a* Hierarchy

If we add the hypothesis that *h is possible hyperonym*, then the measure M_S evaluates to which extend w can be substitued by h and is thus a *word substitution measure*. Similarly, M_T is a taxonomy evaluation, and corresponds to a *semantic approximation*, the way one can approximate *horse* by *mammal*.

We have made the experiment by accessing Google *(www.google.com)* and the number of hits returned for each request. For example, we have the following result for the term *airplane*:

aircraft /$M_T = 0.2659$ $M_S = 0.025$
plane /$M_T = 0.1237$ $M_S = 0.1741$
flying plane /$M_T = 0.5317$ $M_S = 0.0007$
aircraft heavier than air /$M_T = 0.5238$ $M_S = 0.00004$

The most substituable candidate is *plane* (highest M_S), but it is the worst on the taxonomy evaluation. *flying plane* is the most precise term (highest M_T) but reasonably cannot be used instead of *airplane*. For a larger example on *horse*, refer to annex.

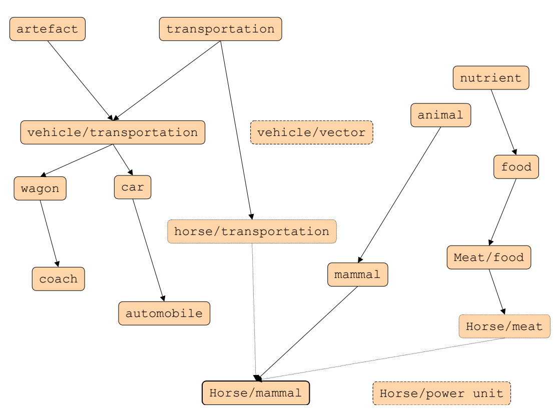

Figure 1. Hyperonym insertion in the built semantic network. Adding found hyperonyms can lead to the identification either (1) of new salient properties in already existing meanings or (2) of new meanings altogether. Thematic distance is used as a meaning selector.

In this case, we do create the new meanings (*horse/ transportation mean* and *horse/meat*) and link them to their hyperonyms. The problem is that starting from vectorized definitions, there is no way to catch these new meanings as they are not (yet) identified. Thus, to overcome this problem, we link each of these new meanings as hyperonym to its closest already existing counterpart. In the above example, we have:

- *horse/ transportation mean* is closer to *horse/mammal* than to *horse/power unit*. This relation can be checked on their respective vector, and (sometimes) by pattern matching on some part of (encyclopedic) definition.
- *horse/meat* is closer to *horse/mammal* than to *horse/power unit*.

4.2 Inclusion Model

If A is an hyperonym of B, then the properties of A are included in B. This can be mesured through vector intersection and distance:

$$H(A,B) \Rightarrow D_A(V(A), \gamma(V(A)\ V(B))) \leq D_A(V(B), \gamma(V(A), V(B))) \quad (13)$$

For example, we have the following measure between *horse/mammal* and *mammal*:
$D_A(V(\text{horse}), \gamma(V(\text{horse})\ V(\text{mammal}))) = 0.41$
$D_A(V(\text{mammal}), \gamma(V(\text{horse})\ V(\text{mammal}))) = 0.25$
From this result, we deduce that *mammal* properties are included in *horse* Morover, if we know that *horse* and *mammal* are in a hyperonymic relation, then *mammal* is the hyperonym. Of course to know the existence of this relation, the co-occurence model is the answer.

4.3 Limits of the Model

The model operates very well for vectors that has been computed from hyperonymic definitions. But for very general terms, where definitions tends to be hyponymic (a collection of examples), the vector inclusion is reversed. More precisely, this is called the *horizon limit*. The horizon is constituted by leaves concepts of the taxonomy on which the vector space is defined.

When the definition leads to a new vector, vectors of the terms present in this definition are mixed. Thus, the vector is flat compared to the main concept(s) involved. We have a formal measure for *flatness* which is the *coefficient of variation* C_V:

$$C_V(X) = \frac{s(X)}{\mu(X)} \quad \text{with} \quad s^2(X) = \frac{\sum(x_i - \mu(X))^2}{n} \quad (14)$$

The C_V is the standard deviation s of the component of the vector divided by the mean μ. This a unitless value.

By definition, C_V is only defined for non nul vectors. If $C_V(A) = 0$ then the vector A is flat, at its maximum value (around 29 when $n = 873$), we have a boolean vector (only one component is activated with 1 all others are zeros).

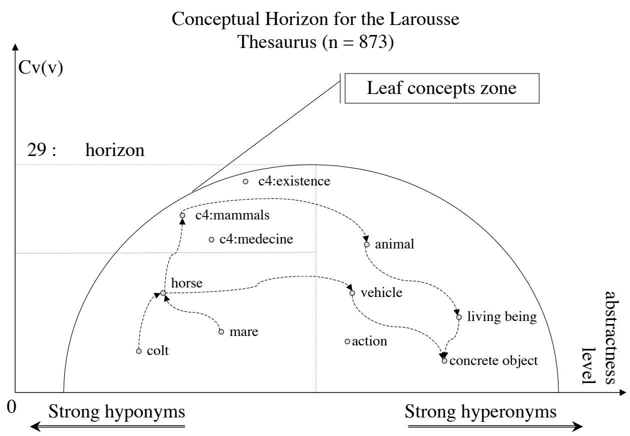

Figure 2. Graphical representation of the conceptual horizon. The horizon stands at the highest level of the variation coefficient which is the lowest level of the thesaurus hierarchy. On the left side, we have terms that are strictly specialization (by mixing) of concepts. On the right side, we have generalization of concepts, which similarly by vector mixing tend to lower the variation coefficient of vectors.

Over the horizon, we do have:

$$H(A, B) \Rightarrow D_A(V(A), \gamma(V(A)\ V(B))) \geq D_A(V(B), \gamma(V(A), V(B))) \quad (15)$$

How can we assess on which side of the *concept hill* a given vector stands. By itself, the variation coefficient just evaluates the general shape of the vector and its conceptuality relatively to the concept set. We have two ways to solve this problem:

1. focusing on a lexical approach mixing lexical functions and information to vectors. The co-occurence model is a possible answer and more generaly semantic graphs as well.

2. another approach is to include, as dimension of the vector space, every concepts of the hierachy and not only the leaves. This solution is only partial, because it cannot tackle the added problem of polysemy if we work on the lexical item and not on the acception level.

4.4 Discussion

We have given some numerical results in annex. The experiments we have conducted on a collection of nouns (and compound nouns), revealed the problem of the conceptual horizon. This horizon stands at the lowest level of the concepts hierarchy (we used [8] for French language). Because of the nature of vector composition, the inclusion model should be inverted when terms stand beyond this horizon.

The detection of crossing of the conceptual horizon is done through lexical models. More precisely, it can be done through the co-occurence model but also by the identification of hyponyms. The detailed presentation of hyponyms identification is beyond the scope of this paper, but enough is to say that more abstract terms (corresponding to large taxonomic classes) contain a large number of hyponyms. According to our model, hyperonymy and hyponymy functions are not strictly symmetrical (both in their usage and behavior in corpora) and can be used together to strengthen the built network.

An application of our model, still under development, is a paraphrase tool. From a given text, the system produces a new text where terms are substituted by hyperonyms (or quasi synonyms). Initial results shows that the most natural paraphrases are those which maximize substitution value and not taxomomic precision. Such a tool could be used for to globally assess the practical validity of our approach but also as a partial preprocess to Machine Translation.

5 Conclusion

In this paper we have tried to show how to account for hyperonymy within the vector-based frame for semantics, relying on a cooperation between semantic networks and conceptual vectors. After having assessed the importance of lexical functions such as synonymy and antonymy for lexical choice and conceptual vectors construction and usage, we have focused on hyperonymy, more difficult to discriminate in a numeric approach such as ours. As our approach is both data driven and hierarchy-based, we first tried to define the impact of hyperonymy by measuring distances in corpora. These distances help to define word substitution and semantic approximation (with a taxonomical aspect). The theoretical model, both within semantic networks and vector space, being the **inclusion model** we showed how inclusion has been dealt with and what results we have obtained. Although being satisfactory, these result stend to reflect the multifaceted properties of hyperonymy: by being more complex than an *is-a* relation, hyperonymy needs to be constrained by the task to perform. If text correction or explanation are at stake, then word substitution is a good usage of hyperonymic properties. If taxonomy building is the goal, then semantic approximation is a better candidate.

So, the same way other lexical functions such as synonymy and antonymy have been restricted by adding a notion of *relativity* when confronted to text bases, also hyperonymy appears as not absolute, the way an *is-a* relation is. It seems better to decline it into its functions and to define it according to processing goals.

References

[1] Brachman R. J. and James G. Schmolze. An overview of the KL-ONE knowledge representation system. *Cognitive Science*, 9(2):171–216, April–June 1985.

[2] Chauché Jacques. Détermination sémantique en analyse structurelle : une expérience basée sur une définition de distance. *TA Information*, 31(1): 17–24.1990

[3] Deerwester S. and S. Dumais, T. Landauer, G. Furnas, R. Harshman, Indexing by latent semantic analysis. *Journal of the American Society of Information science*,416(6): 391–407,1990.

[4] C.Fellbaum (ed). WordNet: An Electronic Lexical Database. MIT Press, Cambridge, Massachussetts,1998.

[5] Gwei G.M. and E. Foxley. A Flexible Synonym Interface with Application examples in CAL and Help environments. *The Computer Journal* 30 (6): 551–557,1987.

[6] M.A.Hearst . *Automated discovery of WordNet relations*, In C. Fellbaum ed. *WordNet : An Electronic Lexical Database* MIT Press, Cambridge, MA, 131–151, 1998.

[7] Lafourcade M. and V. Prince. *Relative Synonymy and Conceptual Vectors NLPRS01*, 127-134, 2001.

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

[9] Lee,J.H, M.H Kim and Y.J. Lee. Information Retrieval based on conceptual distance in IS-A hierarchies. *Journal of Documentation*, 49(2), 188–207,1993.

[10] Lewis, C. I. *The modes of meaning.* in Linsky ed, "Semantics and the philosophy of language". Urbana. NY, 1952.

[11] G.A Miller and C. Fellbaum. Semantic Networks in English. in Beth Levin and Steven Pinker (eds.) *Lexical and Conceptual Semantics* , 197–229. Elsevier, Amsterdam, 1991.

[12] F.R.Palmer. *Semantics: A New Introduction* . Cambridge University Press, 1976.

[13] Resnik.P. *Using Information Contents to Evaluate Semantic Similarity in a Taxonomy, IJCAI-95*, 1995.

[14] Resnik P. Semantic similarity in a taxonomy:An information-based measure and its application to problems of ambiguity in natural language. *Journal of Artificial Intelligence Research*, 11, 95–130,1999.

[15] P.Resnik . Disambiguating noun groupings with respect to WordNet senses. in S. armstrong, K. Church, P. Isabelle, E.Tzoukermann, S. Manzi and D. Yarouwsky (eds.) *Natural Language Processing using Large Corpora*, Kluwer Academic, Dordrecht, 1999.

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

[17] G. Salton *Automatic Information Organisation and Retrieval*, McGraw-Hill, New York, 1968.

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

[19] Sparck Jones K. *Synonymy and Semantic Classification*. Edinburgh Information Technology Serie, 1986.

[20] Schwab, D., and M. Lafourcade, V. Prince. Antonymy and Conceptual Vectors. *COLING'02*, vol 2/2, 904-910 , 2002.

[21] Yarowsky D. Word-Sense Disambiguation Using Statistical Models of Roget's Categories Trained on Large Corpora. *COLING'92*, 454–460, 1992.

6 Annex

For a larger example, we have the following results for the term *horse*:

mammal / $M_T = 0.81 \quad M_S = 0.0005$ (a)
animal / $M_T = 0.0986 \quad M_S = 0.1523$ (a)
domestic animal / $M_T = 0.133 \quad M_S = 0.0035$ (a)
kind of mammal / $M_T = 0.0481 \quad M_S = 0.00002$ (a)
specie / $M_T = 0.1376 \quad M_S = 0.0857$ (b)
horses / $M_T = 0.4673 \quad M_S = 0.2954$ (b)
equitation / $M_T = 0.3498 \quad M_S = 0.0991$ (c)
representation / $M_T = 0.0399 \quad M_S = 0.0505$ (d)
toy / $M_T = 0.1363 \quad M_S = 0.0184$ (e)
child toy / $M_T = 0.2387 \quad M_S = 0.0004$ (e)
wooden horse / $M_T = 0.2025 \quad M_S = 0.0012$

(e)
woman / $M_T = 0.0363 \quad M_S = 0.4012$ (f)
manlike woman / $M_T = 0.5692 \quad M_S = 0.00003$ (f) unit / $M_T = 0.033 \quad M_S = 0.0647$ (g)
arbitrary unit / $M_T = 0.067 \quad M_S = 0.00004$ (g)
power unit / $M_T = 0.1042 \quad M_S = 0.0003$ (g)

We here have several meanings for *horses*: (a) the animal, (b) the class of horses or specie, (c) horse riding, (d) the representation of a horse, (e) the wooden horse, (f) the manlike women, (g) the power unit. *mammal* is the most precise for the taxonomy but *animal* is a better susbsitution term. *specie* is too vague compared to *horses*. *child toy* is more precise than *toy* but is not as good as a substitute.

Generally short terms are better substitutes (partly because of the economy principle in linguistics) but most of the time they are, on a taxonomic point of view, quite vague or ambiguous.

Another example of the French term *peinture*:

art / $M_T = 0.133 \quad M_S = 0.6913$ (a)
art de peindre / $M_T = 0.649 \quad M_S = 0.0016$(a)
ouvrage / $M_T = 0.2248 \quad M_S = 0.0955$ (b)
ouvrage d'un artiste / $M_T = 1.0 \quad M_S = 0.00001$(b)
matière / $M_T = 0.2543 \quad M_S = 0.1644$ (c)
produit / $M_T = 0.2301 \quad M_S = 0.1755$ (c)
produit à base de pigments / $M_T = 1.0 \quad M_S = 0.00004$ (c)
produit à base de pigments en suspension / $M_T = 1.0 \quad M_S = 0.00004$ (c)
produit à base de pigments en suspension dans un liquide / $M_T = 1.0 \quad M_S = 0.00004$ (c) couche / $M_T = 0.1443 \quad M_S = 0.0876$ (d)
couche de couleur / $M_T = 0.4939 \quad M_S = 0.0004$ (d)
description / $M_T = 0.2049 \quad M_S = 0.1216$ (e)

The term *peinture* could be: (a) the *art*, (b) *painting*, (c) the *coloring matter*, (d) the *color layer*, and (e) a *description*. We can see that very precise term are not good substitutes (see different cases for (c)). And inversely best substitutes are often more general and possibly polysemous terms.

Generally short terms are better substitutes (partly because of the economy principle in linguistics) but most of the time they are, on a taxonomic point of view, quite vague or ambiguous.