Guessing Hierarchies and Symbols for Word Meanings through Hyperonyms and Conceptual vectors

Mathieu Lafourcade

LIRMM - MONTPELLIER - FRANCE.
lafourca@lirmm.fr

Abstract. The NLP team of LIRMM currently works on lexical disambiguation and thematic text analysis [*Lafourcade*, 2001]. 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. In the framework of knowledge and lexical meaning representation, we devise some conceptual vectors based strategies to automatically construct hierarchical taxonomies and validate (or invalidate) hyperonymy (or superordinate) relations among terms. Conceptual vectors are used through the thematic distance for decision making and link quality assessment.

1 Introduction

In the framework of meaning representation, the NLP team of LIRMM currently works on strategies for automatically populating hierarchical taxonomies. Such strategies are based on the simultaneous exploitation of the conceptual vector model, definitions found in human usage dictionaries, and free text. The conceptual vector model aims at representing thematic activations for chunks of text, lexical entries, locutions, up to whole documents. Roughly speaking, vectors are supposed to encode *ideas* associated to words or expressions. The main applications of the model are thematic text analysis and lexical disambiguation [*Lafourcade*, 2001] and can found interesting approaches for vector refinement through the lexical implementation of taxonomies. Practically, we have built a system, with automated learning capabilities, based on conceptual vectors and exploiting monolingual dictionaries for iteratively building and refining them. So far, from French, the system learned 87000 lexical entries corresponding to roughly 350000 vectors (the average meaning number being 5). We are conducting the same experiment for English.

With these lexical and vector resources, we can, in conjunction with simple hyperonym (or superordinate) extraction methods [*Hearst*, 92], automatically construct many partial hierarchies. In our context, the hyperonymy relation is (perhaps abusively) considered as the inverse of the *hyponymy* relation, more often refered in software engineering as *specialization*. The *hierarchy soup* composed of hierarchy fragments is built through an iterated process that involves several strategies. The ideas, applied to French in our experiment, are generic and could be extended to any language. The bootstrapping consists in producing a set of hyperonyms from definition dictionaries that are corresponding directly to meanings as defined in our French dictionary. Filtering and selection are done with the help of thematic distance on the vectors associated to the items. The adjunction of hyponyms extracted from definitions, permits to add new meanings or salient properties to the hierarchies. At least, it allows us to strengthen our links or to detect inconsistencies. Free texts can also be exploited although (contrary to entry definitions), in case of polysemy, a word meaning identification should be carried out.

Beside NLP, taxonomy extraction can find applications in intelligent assistance in domain modeling and software engineering. This is specially critical, when (at least) two sets of classes have to be merged, as strategies based uniquely on class definitions fall short because of their lack of interpreting capabilities of naming entities. The name of a class or of an attribute has normally been chosen by designers for their evocating power, and is definitively (at least) a very strong clue for semantic induction and (at most) sometimes the only information available [*Rayside, 2001*]. So far, automated strategies rely only on symbol matching but never on the semantic association carried by the symbols. Similarly to metrics used in software engineering and class hierarchy factorization [*Dao*, 01], the thematic distance helps evaluating similarity. The main difference between Software Engineering and Lexical Semantics remains for the latter that meaning is a blurred halo in the semantic space and is susceptible of slippage (through metaphor and meronymy, notably).

In this paper, we first expose the conceptual vectors model and the notion of semantic distance and contextualization. Then, we expose the hierarchy building strategies that associate meanings to hyperonyms through sets of correspondences and conceptual distances.

2 Conceptual Vectors

We represent thematic aspects of textual segments (documents, paragraphs, 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 originates. From a set of elementary notions, concepts, it is possible to build vectors (conceptual vectors) and to associate them to lexical items. The hypothesis that considers a set of concepts as a generator to language has been long described in [*Rodget*, 1852] (*thesaurus hypothesis*). Polysemous words combine the different vectors corresponding to the 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.

2.1 Thematic Projection Principle

Let $\mathcal{C}$ be a finite set of n concepts, a conceptual vector V is a linear combination 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 ‹*quotation*› could be projected on the following concepts (the *CONCEPT*[intensity] are ordered by decreasing values): V(‹*quotation*›) = *STOCK EXCHANGE*[0.7], *LANGUAGE*[0.6], *CLASSIFICATION*[0.52], *SYSTEM*[0.33], *GROUPING*[0.32], *RANK*[0.31], *ORGANIZATION*[0.30], *ABSTRACT*[0.25], . . .

In practice, the larger $\mathcal{C}$ is, the finer the meaning descriptions are. In return, computer manipulation is less easy. It is clear, that for dense vectors[1] the enumeration of the activated concepts is long and difficult to evaluate. We would generally prefer to select the thematically closest terms, i.e., the *neighborhood*. For instance, the closest terms ordered by increasing distance of ‹*quotation*› are: $\mathcal{V}$(‹*quotation*›) = ‹*management*›, ‹*stock*›, ‹*cash*›, ‹*coupon*›, ‹*investment*›, ‹*admission*›, ‹*index*›, ‹*abstract*›, ‹*stock-option*›, ‹*dilution*›, . . .

2.2 Angular Distance

Let us define $Sim(A, B)$ as one of the *similarity* measures between two vectors A et B (eq. 1), often used in information retrieval [*Morin*, 1999]. Then, we define an *angular distance* D_A between two vectors A and B (eq. 2). We suppose here that vector components are positive or null, and "·" refers to the scalar product.

$$Sim(A, B) = \cos(\widehat{A, B}) = \frac{A \cdot B}{\|A\| \times \|B\|} \tag{1}$$

$$D_A(A, B) = \arccos(Sim(A, B)) \tag{2}$$

Intuitively, this function constitutes an evaluation of the *thematic proximity* and is the measure of the angle between the two vectors. We would generally consider that, for a distance $D_A(A, B) \le \frac{\pi}{4}$, (i.e. less than 45 degrees) A and B are thematically close and share many concepts. For $D_A(A, B) \ge \frac{\pi}{4}$, 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 can have, for example, the following angles[2] (values are in degrees):

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

[2] Examples are extracted from: `http://www.lirmm.fr/~lafourca`

D_A(‹*profit*›, ‹*profit*›)=0°
D_A(‹*profit*›, ‹*benefit*›)=10°
D_A(‹*profit*›, ‹*finance*›)=19°
D_A(‹*profit*›, ‹*market*›)=28°
D_A(‹*profit*›, ‹*product*›)=32°
D_A(‹*profit*›, ‹*goods*›)=31°
D_A(‹*profit*›, ‹*sadness*›)=65°
D_A(‹*profit*›, ‹*joy*›)=39°

The first value has a straightforward interpretation, as ‹*profit*› cannot be closer to anything else than itself. The second and third are not very surprising since a ‹*benefit*› is quite synonymous of ‹*profit*›, in the ‹*finance*› field. The words ‹*market*›, ‹*product*› and ‹*goods*› are less related which explains a larger angle between them. The idea behind ‹*sadness*› is not much related to ‹*profit*›, contrary to its antonym ‹*joy*› which is thematically closer (either because of metaphorical meanings of ‹*profit*› or other semantic relations induced by the definitions). The thematic proximity is by no way an ontological distance but a measure of how strongly meanings may relate to each others.

The graphical representations of the vectors of ‹*exchange*› and ‹*profit*› shows that these terms are indeed quite polysemous. Two other terms (‹*cession*› and ‹*benefit*›) seems to be more focused on specific concepts. These vectors are the average of all possible meanings of their respective word in the general Thesaurus [*Larousse*, 1992]. It is possible to measure the level of *fuzziness* of a given vector as a clue of the number of semantic fields the word meaning is related to.

Because of the vagueness related either to polysemy or to lacks of precision (only 873 general concepts), we have to *plunge* our vectors into a specialized semantic space. However, we cannot cut loose from the general ones for two reasons. First, even non-specialized words may turn out to be pivotal in word sense disambiguation of specialized ones. Second, we cannot know beforehand whether a given occurrence of a word should be understood in its specialized acception or more a general one.

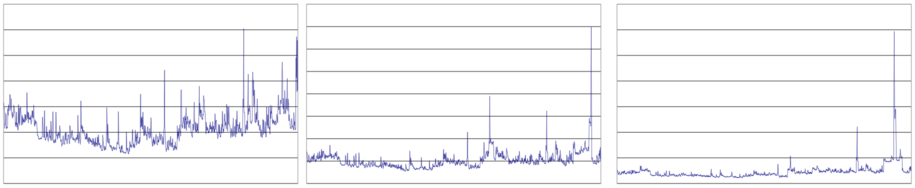

Fig. 1. Graphical representation of (more to less polysemous) terms *exchange*, *benefit* and *cession* (from left to right).

2.3 Vector Operators

Vector Sum. Let X and Y be two vectors, we define their *normed sum* V as:

$$V = X \oplus Y \quad | \quad v_i = (x_i + y_i)/\|V\| \tag{3}$$

This operator is idempotent (we have $X \oplus X = X$). The null vector $\mathbf{0}$ is by definition the neutral element of the vector sum. Thus we write down $\mathbf{0} \oplus \mathbf{0} = \mathbf{0}$. We derive by deduction (without demonstration) the *closeness properties* associated to this operator (both local and general closeness).

$$\begin{aligned} D_A(X \oplus X, Y \oplus X) = D_A(X, Y \oplus X) \leq D_A(X, Y) \\ \text{and} \qquad D_A(X \oplus Z, Y \oplus Z) \leq D_A(X, Y) \end{aligned} \tag{4}$$

Normed Term to Term Product. Let X and Y be two vectors, we define V as *their normed term to term product*:

$$V = X \otimes Y \quad | \quad v_i = \sqrt{x_i y_i} \tag{5}$$

This operator is idempotent ($X \otimes X = X$) and $\mathbf{0}$ is absorbent ($X \otimes \mathbf{0} = \mathbf{0}$).

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:

$$\gamma(X, Y) = X \oplus (X \otimes Y) \tag{6}$$

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

$$D_A(\gamma(X, Y), \gamma(Y, X)) \leq \{D_A(X, \gamma(Y, X)), D_A(\gamma(X, Y), Y)\} \leq D_A(X, Y) \tag{7}$$

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 it 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 be considerably dimmed.

3 Hyperonym identification and hierarchy construction

From term definitions found in human usage dictionaries and from free texts, we extract hyperonym and hyponym sets. The technic used is directly inspired from [*Hearst*, 92] for the use of simple and low cost pattern recognition. The main difficulty is that for a term t with k meanings, the proper word sense should be identified before figuring out its place in a hierarchy. When using (highly) specialized hierarchies [*Llorens*, 2001], we may skip this disambiguation process, although it still might lead to interpretation problems [*Barrière et al*, 01]. We use dictionary definitions and conceptual vectors for, at the same time (1) extracting the hyperonym from a well identified meaning, and (2) disambiguating the hyperonym candidate when needed.

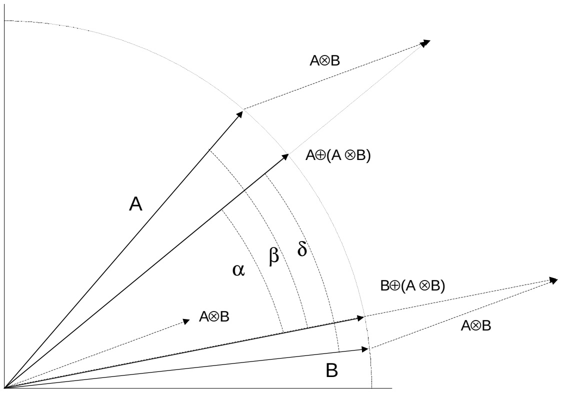

Fig. 2. Geometric represention (in 2D) of the contextualization function. The α angle represents the distance between A and B contextualized by each other.

3.1 Hyperonym identification

For a given term and from several vectorized dictionaries, we extract an hyperonym set. For instance, for the French term *émeraude* (emerald), we have two meanings with the following hyperonym sets:

- Hyper(émeraude.1) = pierre précieuse (src 1), pierre (src 1), [type de] béryl (src 2), gemme (src 3). (Eng. precious stone, stone, [kind of] beryl, gem.)
- Hyper(émeraude.2) = couleur [de l'émeraude], vert, vert lumineux (Eng. color [of emerald], green, shiny green)

The name of the source is given along the potential hyperonyms. Several candidates can be proposed for one definition as several patterns may be applied, and in general the frontier between under and over-contraction of definitions is dim. Parts of hyperonym between brackets are trimmed.

The difficulty here is to find one (or several) acceptable hyperonyms for each meaning. For each set, we compute the conceptual vector of each hyperonym candidate $V(\text{Hyper-cand/meaning}_i)$.

$$V(\text{Hyper-cand/meaning}_i)) = \gamma(V(\text{Hyper-cand}), V(\text{meaning}_i))$$

We contextualize the vector obtained from the definition of the hyperonym candidate with the vector of the definition it has been extracted from. If the hyperonym candidate exist as a term in the conceptual vector lexical database, then its vector is used. Otherwise, its vector is computed by composition of the vector of its sub-term (after a morphological and syntactical analysis).

The selected hyperonym candidate is the term which vector is the closest (in thematic distance terms) to $V(\text{Hyper-cand/meaning}_i))$. We are now able to create a node for each $\text{Hyper-cand/meaning}_i$ with a link to the corresponding disambiguated hyperonym candidate (cf Fig. 3 stage 1). If the term doesn't exist in the lexical database, it is added along its computed vector.

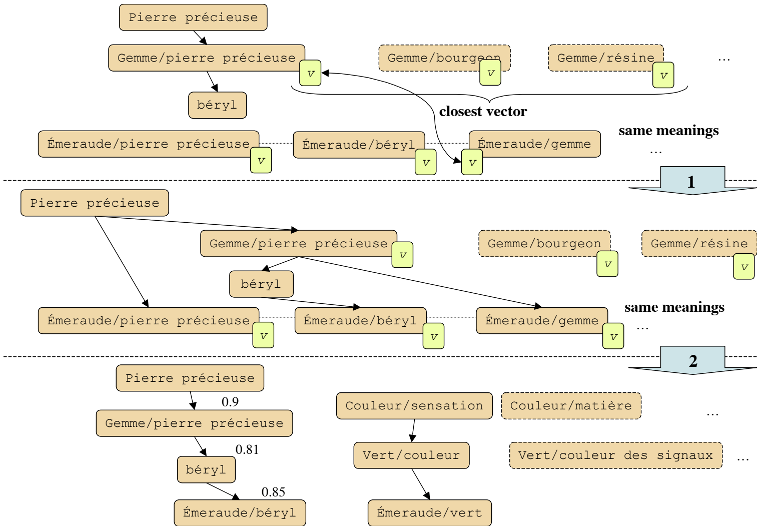

Fig. 3. (1) Linking of each meaning equivalent to its hyperonym. If an hyperonym is by itself ambiguous, its proper meaning is selected by minimizing the thematic distance between vectors. (2) Trimming of redundant meanings. When several meaning equivalents are competing, the one linked to the most specific hyperonym is selected. Other meanings are deleted as they related to upper items in the partial hierarchy.

3.2 Identical meaning trimming

For a given meaning, we have added all disambiguated terms and links to hyperonym to the partial hierarchy. We need to delete all but one of the equivalent terms (cf Fig. 3 stage 2). The objective is to identify the most adequate item. Again, the strategy invoked here is straightforward, as only the link to the most specific (lowest in the hierarchy) is kept. In case of doubt (same level in the hierarchy or incompleteness of the hierarchy), the item which vector is the closest to its hyperonym is kept. In other words, only the most similar couple (term, hyperonym) is chosen.

This above procedure gives us a symbol for naming a given word meaning (this is useful only when the word is polysemous). This symbol is constructed with the simplest possible form: the concatenation of the term and of its most specific hyperonym. From a psycholinguistic point of view (which is out of scope here), the concision of the *symbol* would also be taken into consideration. Furthermore, such symbols are human readable and machine parsable. If the vector of *emerald/green* is not available, we (human and machine) can guess from the symbol that it might be a kind of green, and use the vector of *green* as a substitute. Of course, from the name of the hyperonym, which itself could be polysemous, we cannot without the hierarchy guess the proper meaning. But, in case the hierarchy is lost, the mutual information shared between the hyponym and the hyperonym would in most cases disambiguate both.

3.3 Hyponym added information

In dictionaries, many definitions of very general terms make extensive use of examples. Basically, these examples constitute hyponyms (the inverse relation relatively to hyperonyms) and could be exploited with benefit. The most obvious use of hyponyms is to cross-check hyperonyms, nevertheless we can also extract information that are not directly accessible from normal (hyperonymic) definitions. By the use of hyponymy, the hierarchy cannot take the form of a tree but of a (partial) lattice (cf Fig. 4). Indeed, a meaning can have several hyperonyms. For instance, we have extracted the following hyponyms (among others):

- Hypo(moyen de transport) = véhicule, voiture, avion, train, automobile, **cheval**, $\cdots$ (Eng. Hypo(transport means): vehicle, car, plane, train, motorcar, horse, $\cdots$)
- Hypo(viande) = poulet, agneau, boeuf, **cheval**, mouton, $\cdots$ (Eng. Hypo(meat): chicken, lamb, beef, horse, mutton, $\cdots$)

Here, we can observe that *horse* (cheval) is a particular *meat* (viande) and also a *means of transportation* (moyen de transport). Although, we have the following hyperonyms (familiar meanings excluded):

cheval.1: mammifère. (Eng. mammal)
cheval.2: art de monter à cheval. (Eng. art of ridding horses)
cheval.3: unit de mesure. (Eng. measure unit)

We have seen clearly (through vector contextualization and thematic distance) that the two hyponym sets seem to induce two new meanings that were not given through definitions. In this case, we do create the new meanings (*cheval/moyen de transport* and *cheval/viande*) 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:

- *cheval/moyen de transport* is closer to *cheval/mammifère* than to *cheval/unité de puissance*. This relation can be checked on their respective vector, and (sometimes) by pattern matching on some part of (encyclopedic) definition.
- *cheval/viande* is closer to *cheval/mammifère* than to *cheval/unité de puissance*.

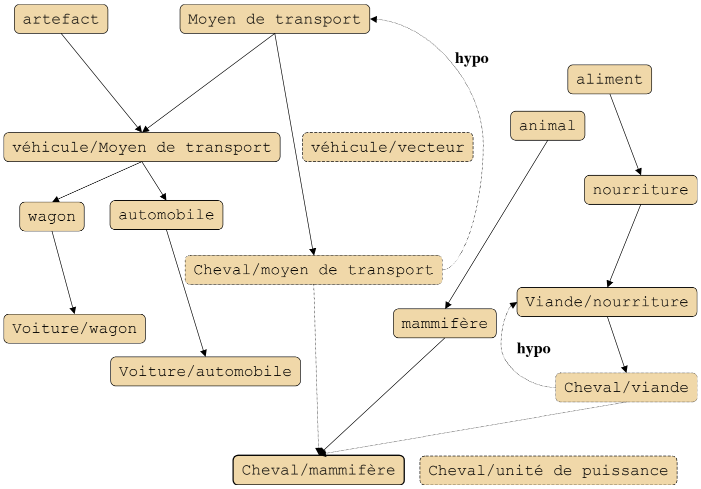

Fig. 4. Hyponym insertion. Adding found hyponyms can lead to the identification either (1) of new salient properties in already existing meanings or (2) of new meanings altogether. Thematic distance is used as a meaning selector.

4 Conclusion

This paper has presented a strategy for hierarchy construction through low cost hyperonym extraction (from definitions) associated to disambiguation and linking decision based on conceptual vectors. By itself, the overall process consists in symbolizing word meaning (giving a unique name to each item that are members of a meaning set). It is now possible to handle a meaning not only by exemplifying its vector, but also by referring to it thanks to its symbol.

Our strategies have been prototyped and have been included in our (conceptual) vector lexical database. It is mainly used for comforting vector calculation and detecting inconsistencies. The overall process is by itself iterative and incremental. And a global hierarchy is being built by fusion of partial ones. Only some hierarchy parts are actually exploited during NLP process, mainly those which are really useful for word sense disambiguation. The experiment has been conducted (and is still in process) on 50000 nouns (for roughly 87000 words). Comforting enough is the fact the constructed hierarchy is really close to some Aristotelian classification. This is basically explained by the fact that the structure of the dictionary definitions draws much on this tradition. The main departure is the multiple inheritance schema that originates from property salience (as show on the term *horse*). In general, the frequency of this phenomenon is inversely proportional to the technicality of the domain.

The produced partial specialization hierarchies enable some automatic refinement of domain representation. In effect, when domains are too specialized, the fine meaning difference cannot be apprehended through conceptual vectors (unless enlarging considerably the vector space). Allowing agents to process informations both at the vectorial and symbolic (as defined above) levels seems definitively a way to solve some aspects of the symbol grounding problem. Beside Natural Language Processing and information retrieval, possible application of this research is to provide intelligent assistance in advanced software engineering. Such assistance would mainly rely on guessing designer intentions through the inspection of names of entities. Relating lexical information to common knowledge could pave the way to more flexible domain representations.

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.

[*Barrière et al*, 01] Barrière C. and T. Copeck *Building Domain Knowledge from Specialized Texts* In Proc. of TIA 2001, 2001, 8 p.

[*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.

[*Hamon*, 01] Hamon T. et A. Nazarenko *La structuration de terminologie : une nécessaire coopération* In Proc. of TIA 2001, 2001, 8 p.

[*Hearst*, 92] Hearst Marti A. *Automatic Acquisition of Hyponyms from Large Text Corpora.* In Proc. of the Fourteenth International Conference on Computational Linguistic COLING'92, 1992, Nantes, France, 8 p.

[*Dao*, 01] Dao, M. M. Huchard, H. Leblanc, T. Libourel, C. Roume. *A new Approach to Factorization - Introducing Metrics* In Proc. of the METRICS 2002, 12 p.

[*Lafourcade et Prince*, 2001] Lafourcade M. et V. Prince *Synonymy and conceptual vectors.* Proc. of NLPRS'2001, Tokyo, Japan, August 2001, pp 127-134.

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

[*Llorens*, 2001] Llorens J. and H. Astudillo *Automatic Generation of Hierarchical Taxonomies from Free Texts Using Linguistic Algorithms.* In Procs of MASPEGHI 2002, Lecture Notes in Computer Science, 7 p.

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

[*Rayside, 2001*] Rayside D. and G. T. Campbell *An Aristotelian Understanding of Object-Oriented Programming* Minneapolis, Minnesota, October 2000. Edited by Doug Lea. pp 337- 353.

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

[*Riloff*, 1995] Riloff E. and J. Shepherd *A corpus-based bootstrapping algorithm for Semi-Automated semantic lexicon construction.* In. Natural Language Engineering 5/2, 1995, pp. 147-156.

[*Resnik*, 1995] Resnik P. *Using Information contents to evaluate semantic similarity in a taxonomy.* In. Proc. of IJCAI-95, 1995, 8 p.

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