Relative Synonymy and Conceptual Vectors

Mathieu Lafourcade and Violaine Prince
LIRMM,
University Montpellier 2
161, rue ADA, 34392 Montpellier Cedex 5, France
{lafourca,prince}@lirmm.fr

Abstract

Synonymy is a pivot relation in NLP but remains problematic. Putting forward, we introduce the notion of relative synonymy, to circumvent some difficulties among which possible polysemy and contextual interpretation. In the framework of conceptual vectors, it is then possible to formalize test functions for synonymy and to experiment their use in thematic analysis that will help text classification.

1 Introduction

Synonymy is, with hyperonymy, one of the most useful lexical functions in Natural Language Processing (NLP) (Sparck Jones, 1986). Whereas synonymy is generally provided through linguistic or ontologic expertise (e.g. Wordnet based works such as (Hearst, 1998)), few systems try to recognize synonymy in context (Gwei and all., 1987). Nevertheless, synonymy is a studied relation in NLP since it has demonstrated its usefulness in:

- creating machine readable dictionaries that help disambiguation (Milne, 1986);
- performing an information retrieval more sophisticated than a character string pattern retrieval (Salton, 1968);
- avoiding the multiplication of concepts in knowledge bases (the same concept would be associated to a list of *synonyms* (Schank, 1973));
- suggesting a good stylistic quality in text generation.

Synonymy is supposed to have the good properties of a semantic equivalence relation between terms[1]. If this property is to be transferred to the formal side, synonymy should be reflexive, symmetric and transitive. Unfortunately these properties are seldom verified, as we will discuss it later.

In this article, we deal with the problems arising from the transposition of synonymy, as a linguistic function, to its detection and measure in a NLP system. This brings us to define, along with other researchers (such as Gwei and Foxley) different types of synonymies, that we have sketched in a broad fashion in a previous work. In this communication, we focus on the concept of *relative synonymy*, which we show as a good way to avoid blind choice between possible synonyms of a polysemous word. We then propose an overview of ideas and processes related to the formalism of conceptual vectors ((Salton, 1968); (Salton and MacGill, 1983)), on which we have built all our framework. We finally describe the test functions that measure semantic closeness and relative synonymy, and we discuss the benefits of the latter.

2 Synonymy in a nutshell

2.1 Synonymy and formal properties

One of the first known *inconvenience* of synonymy, as a relation between terms, is that it does not necessarily verify transitivity (Lewis, 1952). For instance, ‘*(to) run*’ and ‘*(to) walk*

[1] a term is represented by a word or an expression having the status of a lexical entry in a dictionary

fast›, ‹*(to) run*› and ‹*(to) perform*›, are synonymous by pairs but are such that ‹*(to) walk fast*› and ‹*(to) perform*› are not synonymous. Examples are numerous but we chose here the concept of *SORTING*[2] because, as we have mainly worked on French (presently, we are transferring our system to English for the needs of automatic translation assistance), we have tested verbal synonymy with some very polysemic verbs such as ‹*(to) sort (out)*›. Fortunately, we have almost exactly the same type of polysemy for this verb in both French and English. This will help explaining the problem and housing information for the forthcoming translation system. Practically, many concepts are invoked by *SORTING*

- *CLASSIFYING* such as in [*sorting out a list of fifty elements by ascending order*]
- *SELECTING* such as in [*sorting out stamps for a collection*]
- *SEPARATING* such as in [*sorting clothes between clean and dirty*]
- *MENDING* such as in [*sorting one's bike*]
- *SOLVING* such as in [*sorting out a problem*]
- *UNDERSTANDING* such as in [*sorting out what happened this morning*]
- *PUNISHING* such as in [*sorting somebody out*]

English and French share the first three concepts as related to the verbal form ‹*(to) sort (out)*›. This multiplicity shows that if *SORTING* is synonymous to any of the related concepts, the later are not equivalent to each others. (Fischer, 1973) asserts that synonymy is at best a tolerance relation[3].

The second disadvantage of synonymy is that it could be, at least partially, confused with hyperonymy[4]. For instance, ‹*(to) parcel*› could be replaced by ‹*(to) divide up*› although it is a hyponym, specialised in usage about land division.‹*(to) divide up*› is in turn a hyponym of ‹*(to) cut*› which in turn could means as well ‹*(to) remove*› in [*cutting a piece of bread*] as ‹*(to) split*› in [*cutting the wood for fire*]. This *misfit* shows a weakness in the symmetry of the relation and thus tempers with its status as a (strong) tolerance relation. Hyperonymy is not a symmetrical relation and a hyperonym playing out the role of a synonym transfers the constraints of its original function.On the opposite site, we will also note that a polysemous term may have several hyperonyms coming out of different branches of the conceptual representation. Last, two hyponyms of a given terms are not necessarily synonyms. For instance, ‹*knifing*› and ‹*shooting*› are hyponyms of ‹*murdering*› and they are not synonymous since they pinpoint at the use of a different weapon.

This brings us to define *synonymy* as **the ability for two terms, to share the most important number of semantic features or to have the widest possible basis in common.** This definition leads naturally to a large family of possible *neighbours*. It is thus important to define those among the possible *synonymy relations* which will present the most robust features for applications such as indexation and information retrieval in a corpus.

2.2 Relative Synonymy

To circumvent the *untransitivity* of synonymy, a notion of *relative synonymy* has been defined that is built upon the following principle: two terms may be synonymous according to the central idea (or topic) developed in a third term, or by one of them (Lafourcade and Prince, 2001). The related approaches to relative synonymy could be seen in Sabah's *near synonymy* (Sabah, 1984) in a context of semantic networks, and Gwei and Foxley's *contextual synonymy* in text analysis. Thus ‹*(to)*

[2] we represent the lexical entries by ‹*entry*›, corresponding to words and expressions in the dictionary, and the concepts by *CONCEPT*. In general, when a concept is also represented by a verb, the conceptual form is VERB-*ing*.

[3] a tolerance relation could be symmetric and reflexive but not transitive. There are several levels of tolerance, according to the verified properties.

[4] a hyperonym is the parent concept or term of a given term. For instance ‹*parent*› is hyperonym of ‹*father*› and of ‹*mother*›. In turn, both of them are hyponyms of ‹*parent*›.

sort (out)› and ‹*(to) select*› are synonymous with respect to the discriminant concept of *SELECTING*, whereas ‹*(to) sort (out)*› and ‹*(to) classify*› are synonymous with respect to *CLASSIFYING*. With a third term, it may funtion in the same way. The concept *CLASSIFYING*, helps relating ‹*(to) sort (out)*› and ‹*(to) organize*›, ‹*(to) sort (out)*› and ‹*(to) put in order*›. All the items that are synonymous with respect to a same third one (which could be one of them) are synonymous by pairs. Thus, the synonymy relative to a third is a transitive function. For instance ‹*(to) choose*› and ‹*(to) select*› are synonymous with respect to *SELECTING*,therefore, ‹*(to) sort (out)*› and ‹*(to) choose*› are also synonymous with respect to *SELECTING*. The benefit of such a relation is that it then becomes an equivalence relationship (a formal demonstration of this is given in (Prince, 1991)) thus restoring to synonymy all its deductive features.

3 Conceptual Vectors

We focus on meaning representation in NLP and its application to thematic analysis and information retrieval. Thus we have chosen to represent the thematic aspects of segments of text (such as documents, paragraphs, phrases and so forth) in the shape of *conceptual vectors* such as in (Lafourcade and Sandford, 1999). This approach originates from the works of (Chauché, 1990) through the use of a predetermined set of concepts out of the French Thesaurus (Larousse, 2001). The same idea has been exploited in (Gwei, 1984), (Gwei and all., 1987) and the Roget thesaurus for English. The formalism is issued from the vectorial model defined in Salton's work, namely in (Salton and MacGill, 1983), and from the LSI model (Deerwester and all., 1990) in which the interdependence of concepts is recognized and employed. However, the originality of our approach reposes on the fact that we use a morphosyntactic parsing of texts to extract vectors instead of a keyword surface analysis as presented in (Salton, 1988). The morphosyntactic parsing generates structural analysis trees, the geometry of which is one of the parameters of our vectorial calculus. Generally speaking, the documents are processed independently, unlike LSI, and we focus on lexical selection in context. When comparing with (Resnik, 1995), our approach differs with respect to the exclusive usage of taxonomies.

We rapidly present the main mathematical properties conceptual vector model that we use in the next section. From a paradigmatic point of view, this model operates as the projection (in the scope of a well-known mathematical model) of the linguistic notion of semantic field.

3.1 Basic assumptions

The main assumptions grounding our approach are the following:

- *(assumption 1)* Since linguists have organized the conceptual framework for language in thesaurii (i.e. Larousse for French, Roget for English), we rely upon the conceptual hierarchy provided for each language as the generating family of conceptual vectors.

- *(assumption 2)* For French, Larousse has enumerated 873 concepts, organized in a tree with 4 levels of abstraction (the Roget proposes around 1000 concepts, however, many of the exceeding concepts belong to religious descriptions that are absent in French). Thus the dimension of the generated vector space is 873, although the vectors representing these concepts do not form a free space (in fact, the "real" dimension could be smaller). But as we have no way to define a real basis for the vector space, we will restrain our ambition to a space with a generator.

- *(assumption 3)* We assume that every term, and thus every segment, has a projection in this vector space, and could be represented by a linear combination of vectors.

- *(assumption 4)* Every vector generated in the vector space represents a *meaning*, whether this meaning is embodied in a term or a segment (assumption 3) or not.

In fact, the largest $\mathcal{C}$ is, the finer would be all meaning descriptions offered by vectors, but the heavier would be their computational handling. Building a *conceptual lexicon* (the set of *(term, morphological variables, vector)* triplets) is automatically performed from definition corpora (Lafourcade, 2001)). At the present moment (when writing this article) the French definition corpus corresponds to 160000 definitions associated to 60000 lexical entries (unflexed) (about 27000 monosemous words and 33000 polysemous words. For the latter,the average number of definitions, some of them possibly redudant, is about le 4.54). This amounts to around 2 GigaBytes of text.

3.2 Basic Principles and Properties

Let $\mathcal{C}$ be a finite set of n concepts. A conceptual vector V is a linear combination of the c_i elements of $\mathcal{C}$. (assumption 3). For a meaning A, the vector V_A is the description (in extension) of those concepts $\mathcal{C}$ activated to represent A . For instance, the meanings of ‹*(to) classify*› and of ‹*(to) cut*› have been projected on the following concepts, with an *intensity* calculated on the different text analysis structural trees. The presentation we give is by pairs CONCEPT[*intensity*] ordered by decreasing intensity:

$V_{\text{(to) classify}}$ = (CHANGE[*0.84*], VARIATION[*0.83*], EVOLUTION[*0.82*], ORDER[*0.77*], SITUATION[*0.76*], STRUCTURE[*0.76*], RANK[*0.76*] ...).
$V_{\text{(to) cut}}$ = (PLAY[*0.8*], LIQUID[*0.8*], CROSS[*0.79*], PART[*0.78*] MIXING[*0.78*] FRACTION[*0.75*] SUFFERING[*0.75*] WOUND[*0.75*] DRINK[*0.74*] ...).

The intensity calculus is made through a learning process presented in (Lafourcade, 2001). It is obvious that, for *dense* vectors (i.e. vectors with very few null components), the enumeration of activated concepts is very quickly tedious and especially difficult to evaluate. We will prefer to select thematically close terms . For instance the terms close to ‹*(to) classify*› and ‹*(to) cut*›, ordered by decreasing thematic distance are:

‹*(to) classify*› : ‹*(to) sort (out)*›, ‹*(to) catalog*›, ‹*(to) select*›, ‹*(to) put in order*›, ‹*(to) distribute*›, ‹*(to) group*›, ‹*(to) dispatch*›, ‹*(to) align*›, ‹*(to) arrange*›, ‹*(to) clean* ›, ‹*(to) distribute*›, ‹*(to) discriminate*›, ‹*(to) ajust*› ...

‹*(to) cut*› : ‹*(to) scissor*›, ‹*(to) slice thinly*›, ‹*(to) saw*›, ‹*(to) cut up*›, ‹*(to) trim*›, ‹*(to) clip*›, ‹*(to) intersperse*›, ‹*(to) break*›, ‹*(to) mutilate*›, ‹*(to) plough*›, ‹*(to) geld*›, ‹*(to) wring*›, ‹*(to) mangle*›, ‹*(to) tear*›, ‹*(to) decimate*›, ...

The thematically close vectors are those which have a small *angular distance*, a notion that is defined in the next subsection.

3.3 Angular Distance

Angular distance is the mean of semantic closeness of two vectors and thus can be viewed as the semantic proximity the terms represented by these vectors. Let $Sim(X,Y)$ be the measure of *similarity* between two vectors, usually employed in information retrieval, defined according to the formula (1) defined hereafter (with "·" as the scalar product). We assume here that the vector components are always positive or null (it is not necessarily true). We define an *angular distance* function D_A between two vectors X and Y according to formula (2).

$$Sim(X,Y) = \cos(X,Y) = \frac{X \cdot Y}{\|X\| \times \|Y\|} \quad (1)$$

$$D_A(X,Y) = \arccos(Sim(X,Y)) \quad (2)$$

Intuitively this function plays the role of an evaluation of thematic *closeness*. Practically, it is the measure of the angle between the two vectors. Generally speaking, we will consider that for a distance $D_A(X,Y) \leq \pi/4$ rad (around 22.5°), X and Y are semantically close, and share some concepts. When $D_A(X,Y) \geq \pi/4$,the semantic closeness of A and B will be considered weak. Around $\pi/2$ (angle = 90°),the meanings are unrelated. Synonymy, in its broader sense is included in thematic closeness, however it demands concordance in morphosyntactic categories. The opposite is, of course, not true. Thematic distance is a true distance, unlike the measure of similarity, and it verifies the needed properties of reflexivity (3), symmetry (4) and triangular inequality (5):

$D_A(X,Y)$	sort	classify	choose	hierarchize	dispatch	file	distribute
sort	0	0.517	0.662 d_1	0.611 d_2	0.551	0.441	0.462
classify		0	0.829	0.6	0.523	0.409 d_4	0.444
choose			0	0.848 d_3	0.77	0.796	0.758
hierarchize				0	0.595	0.523	0.519
dispatch					0	0.471	0.391
file						0	0.36
distribute							0.0

Table 1: Angular distances between possible synonyms of ‹*(to) sort (out)*›

$$D_A(X,X) = 0 \quad (3)$$
$$D_A(X,Y) = D_A(Y,X) \quad (4)$$
$$D_A(X,Y) + D_A(Y,Z) \geq D_A(X,Z) \quad (5)$$

By definition $D_A(\vec{\mathbf{0}},\vec{\mathbf{0}}) = 0$ and $D_A(X,\vec{\mathbf{0}}) = \pi/2$ with $\vec{\mathbf{0}}$ representing the null vector[5]. We consider, in a generalising fashion, the extension of the image domain of D_A to $[0,\pi]$ in order to compare vectors with negative values on some components. This generalization does not modify the properties of D_A. We also note that angular distance is unsensitive to vector norming (α and β being scalars):

$$D_A(\alpha X, \beta Y) = D_A(X,Y) \quad \text{with} \quad \alpha\beta > 0 \quad (6)$$
$$D_A(\alpha X, \beta Y) = \pi - D_A(X,Y) \quad \text{with} \quad \alpha\beta < 0 \quad (7)$$

For instance[6], in table 1, we present the angular distances (in rads) between vectors representing several terms. The table is symmetrical (D_A being symmetric) and the diagonal is always equal to 0 (D_A being reflexive). Let us mention that a value is meaningful relatively to another. For instance, it is satisfying to have, in the table:

1. a) $d_1 \leq d_3$ and $d_2 \leq d_3$ matching the fact that ‹*(to) sort (out)*› and ‹*(to) classify*› on one hand, and ‹*(to) sort (out)*› and ‹*(to) select*› on the other, are "more synonymous" than ‹*(to) classify*› and ‹*(to) choose*›
2. b) d_4 is the smallest value of D_A((to) classify, Y) because the concepts *CLASSIFYING* and *DISPATCHING* are relatively close. Moreover, ‹*(to) classify*› is also polysemous (*HIERARCHIZING*, *ASSEMBLING* and *FILING*) and only *CLASSIFYING* is present in the table.

[5]The null vector obviously corresponds to no word of any language. It is an idea that activates ... no concept! It is a necessary fiction that plays the role of the mathematical neutral for vector addition.

[6]all the examples of this articles are extracted from our French knowledge source and manually translated for the moment

3.4 Vectors Operators

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

$$V = X \oplus Y \quad | \quad v_i = (x_i + y_i)/\|V\| \quad (8)$$

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

$$D_A(X \oplus X, Y \oplus X) = D_A(X, Y \oplus X) \leq D_A(X,Y) \quad (9)$$

$$D_A(X \oplus Z, Y \oplus Z) \leq D_A(X,Y) \quad (10)$$

Vector Substraction. Let X and Y be two distinct vectors. We define V as their *normed difference* as following:

$$V = X \ominus Y \quad | \quad v_i = (x_i - y_i)/\|V\| \quad (11)$$

This operator is not idempotent and we have, by definition : $V = X \ominus X = \vec{\mathbf{0}}$. Let us notice that, in general, the v_i values may be negatice, and that the distance function has its image over $[0, \pi]$.

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

$$V = X \otimes Y \quad | \quad v_i = \sqrt{x_i y_i} \tag{12}$$

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

$$\begin{aligned} V = X \otimes X = X \\ V = X \otimes \vec{\mathbf{0}} = \vec{\mathbf{0}} \end{aligned} \tag{13}$$

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. For operational goals, we also define the opposite function named *anti-contextualisation*. Let X and Y be two vectors, we define $\Gamma(X, Y)$ (resp. $\overline{\Gamma}(X, Y)$) as the contextualisation (resp. the anti-contextualisation) of X by Y as:

$$\Gamma(X, Y) = X \oplus (X \otimes Y) \tag{14}$$
$$\overline{\Gamma}(X, Y) = X \ominus (X \otimes Y) \tag{15}$$

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

$$\begin{aligned} & D_A(\Gamma(X, Y), \Gamma(Y, X)) \\ \leq & \{D_A(X, \Gamma(Y, X)), D_A(\Gamma(X, Y), Y)\} \\ & \leq D_A(X, Y) \end{aligned} \tag{16}$$

$$\begin{aligned} & D_A(\overline{\Gamma}(X, Y), \overline{\Gamma}(Y, X)) \\ \geq & \{D_A(X, \overline{\Gamma}(Y, X)), D_A(\overline{\Gamma}(X, Y), Y)\} \\ & \geq D_A(X, Y) \end{aligned} \tag{17}$$

The function $\Gamma(X, Y)$ brings the vector X closer to Y proportionally to their intersection. $\overline{\Gamma}(X, Y)$ procedes symmetrically.

3.5 Examples

In table 2, we have, in the upper part, the reminding of the angular distance values **(a)** $D_A(\Gamma(X, Y), \Gamma(Y, X))$ and in the lower part the values of **(b)** $D_A(\overline{\Gamma}(X, Y), \overline{\Gamma}(Y, X))$. The interpretation corresponds exactly to the one presented hereabove, that is, testing the thematic closeness of two meanings (A and B), each one enhance with what it has in common with a third (C).

4 Relative Synonymy

We define the *relative synonymy* function Syn_R, between three vectors A, B and C, the later playing the role of a pivot, as:

$$\begin{aligned} Syn_R(A, B, C) &= D_A(\Gamma(A, C), \Gamma(B, C)) \\ &= D_A(A \oplus (A \otimes C), B \oplus (B \otimes C)) \end{aligned} \tag{18}$$

The interpretation corresponds exactly to the one presented thereabove, that is, testing the thematic closeness of two meanings (A and B), each one enhanced with what it has in common with a third (C).

4.1 Properties

We verify the three theoretical properties of an equivalence relationship, as we claimed to obtain with relative synonymy as such:

1. $Syn_R(A, A, C) = 0$
   Reflexivity is inherited from angular distance D_A.
2. $Syn_R(A, B, C) = Syn_R(B, A, C)$
   Symmetry of the two first arguments (those which are compared) comes also out of angular distance.
3. $Syn_R(A, B, E) + Syn_R(B, C, E) \geq Syn_R(A, C, E)$
   This is an inheritance of triangular inequality of D_A. It represents a *pseudo-transitivity* for relative synonymy. It indicates that the distance between A and C/E is at worst equal to the sum of the measures of synonymy between A and B/E on one hand, and B and C/E on the other hand.

$b \backslash a$	sort	classify	choose	hierarchize	dispatch	file	distribute
sort	0	0.269	0.363	0.322	0.288	0.228	0.239
classify	2.183	0	0.474	0.316	0.273	0.211	0.23
choose	2.401	2.17	0	0.485	0.434	0.451	0.425
hierarchize	2.382	2.374	2.314	0	0.313	0.272	0.27
dispatch	2.334	2.303	2.282	2.483	0	0.244	0.201
file	2.505	2.481	2.313	2.648	2.535	0	0.185
distribute	2.476	2.388	2.364	2.637	2.53	2.761	0

Table 2: Contextualized (up) and anti-contextualized (down) angular distances between possible synonyms of ‹*(to) sort (out)*›

4. $Syn_R(A, B, 0) = D_A(A \oplus \vec{\mathbf{0}}, B \oplus \vec{\mathbf{0}}) = D_A(A, B)$
   The null vector $\vec{\mathbf{0}}$ makes relative synonymy collapse with angular distance.
5. $Syn_R(A, B, C) \leq D_A(A, B)$
   By inheritance of the closeness of D_A, whatever the point of view is, relative synonymy can but bring A and B closer to each other.

4.2 Examples

In table 3, we have, in the upper part, the reminding of the angular distance values **(a)** $D_A(X, Y)$ and in the lower part, the values of synonymy of every item measured relatively to the context (i.e. (to) sort (out)) **(b)** $Syn_R(X, Y, sort)$. We see here the effect of polysemy. For instance, we have:

Syn_R(*(to) file, (to) classify, (to) sort (out)*)

weighing 0.283, which indicates a strong relative synonymy of ‹*(to) file*› and ‹*(to) classify*› with respect to ‹*(to) sort (out)*›, whereas the corresponding angular distance (0,409) did not hint at so strongly. And reciprocally, Syn_R(*(to) choose, (to) classify, (to) sort (out)*) weighs 0.636. This tends to show that ‹*(to) choose*› and ‹*(to) classify*› are not synonymous with respect to ‹*(to) sort (out)*›, although they are both possible synonyms of ‹*(to) sort (out)*›. Relative synonymy looks like a good clue for polysemy : ‹*(to) choose*› and ‹*(to) hierarchize*› belong, in most of their features, to two distinct semantic "areas".

5 Conclusion

The study we have conducted about synonymy in lexical knowledge sources has shown that: 1) In a global modelling, such as conceptual vectors space, where the input is words invoking ideas and not concepts combining into words, synonymy has properties that could be expressed in terms of measure. 2) To make a synonymy measure get close to the good mathematical properties of equivalence or quasi-equivalence relationships that we hope to obtain, we have been driven to formalize a particular type of synonymy, called *relative synonymy.*

The latter enhances grouping of terms, quasi-equivalent, with respect to a given topic (called a context). In a broader work, we also defined another type of synonymy, called *subjective synonymy*, whose goal is to track down some properties that could be discriminant in addressing hyperonymy. Whereas the latter appears obvious when given through human expertise in domain ontologies, in the way concept → word, it is much more difficult to assert in the opposite direction, word → concept. Relative synonymy and, we hope, other lexical functions such as antonymy, will help us building a fundamental functional structure that will enrich both knowledge sources and thematic text analysis.

References

Chauché J. *Détermination sémantique en analyse structurelle : une expérience basée sur une définition de distance.* TA Information, 1990, vol 31/1, p 17-24.

Deerwester S. and S. Dumais, T. Landauer, G. Furnas, R. Harshman, *Indexing by latent semantic analysis.* In Journal of the AmericanSociety of Information science, 1990, 416(6), pp 391-407.

Fischer, W. L. *Äquivalenz und Toleranz Struk-*

*b**a*	sort	classify	choose	hierarchize	dispatch	file	distribute
sort	0.0	0.517	0.662	0.611	0.551	0.441	0.462
classify	**0.402**	0.0	0.829	0.6	0.523	0.409	0.444
choose	**0.5**	**0.623**	0.0	0.848	0.77	0.796	0.758
hierarchize	**0.478**	**0.43**	**0.636**	0.0	0.595	0.523	0.519
dispatch	**0.435**	**0.365**	**0.575**	**0.435**	0.0	0.471	0.391
file	**0.369**	**0.283**	**0.607**	**0.385**	**0.344**	0.0	0.36
distribute	**0.376**	**0.309**	**0.57**	**0.383**	**0.272**	**0.268**	0.0

Table 3: Angular distance (up) and Relative synomy (down) between '*(to) sort (out)*' and some other verbs

turen in der Linguistik zur Theory der Synonyma. Max Hüber Verlag, München, 1973.

Gwei G.M. *The Roget Environment (Synonym Generation).* Internal report, Computer Science Group, Nottingham University, Dec. 1984.

Gwei G.M. and E. Foxley *A Flexible Synonym Interface with application examples in CAL and Help Environments.* The Computer Journal 30 6; 1987, pp 551-557.

Hearst, M. A. *Automated discovery of Wordnet relations.* In C Fellbaum ed. "Wordnet An Electronic Lexical Database". MIT Press, Cambridge, MA, 1998, pp 131-151.

Lafourcade M. and E. Sandford, *Analyse et désambiguisation lexicale par vecteurs sémantiques.* In Proc. of TALN'99 (Cargèse, July 1999). pp 351-356.

Lafourcade M. *Lexical sorting and lexical transfer by conceptual vectors.* In Proc. of The First International Workshop on MultiMedia Annotation (MMA'2001) (Tokyo, January 2001) 6 p.

Lafourcade M. and V. Prince. *Synonymies et vecteurs conceptuels.* In Proceedings of TALN'01 (Tours, July 2001) 1; pp. 233-242.

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

Lesk M.E. *Automatic Sense Disambiguation Using Machine Readable Dictionaries: How to Tell a Pine Cone from an Ice Cream Cone*Proceedings of ACM SIGDOC conference(Toronto, 1986).

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

Milne R. *Resolving Lexical Ambiguity in a Deterministic Parser .* Computational Linguistics 12; 1; 1986.

Ploux S. et B. Victorri *Construction d'espaces sémantiques à l'aide de dictionnaires de synonymes.* In TAL, Vol 39/1, 1998 pp 161-182.

Prince, V. *Notes sur l'évaluation de la réponse dans TEDDI : introduction d'une relation d'équivalence pour la synonymie relative.* Notes et Documents LIMSI 91-20. 1991, CNRS. 22 p.

Resnik P. *Using Information contents to evaluate semantic similarity in a taxonomy.* In Proceedings of IJCAI-95, 1995.

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

Sabah, G. *Différentes notions de synonymies liées à la compréhension du langage.* Actes du colloque de l'Association pour la Recherche Cognitive 1984.

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

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

Salton G. *Term-Weighting Approaches in Automatic Text Retrieval.* McGraw-Hill computer science series. McGraw-Hill,voL 24, 1988.

Schank R. *Identification of Conceptualizations Underlying Natural Language.* Computer Models of Thought and Language W. H. Freeman and Co, San Francisco 1973.

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