

Antonymy and Conceptual Vectors

Didier Schwab, Mathieu Lafourcade and Violaine Prince

LIRMM

Laboratoire d'informatique, de Robotique
et de Microélectronique de Montpellier
MONTPELLIER - FRANCE.

{schwab,lafourca,prince}@lirmm.fr

<http://www.lirmm.fr/~{schwab,lafourca,prince}>

Abstract

For meaning representations in NLP, we focus our attention on thematic aspects and conceptual vectors. The learning strategy of conceptual vectors relies on a morphosyntactic analysis of human usage dictionary definitions linked to vector propagation. This analysis currently doesn't take into account negation phenomena. This work aims at studying the antonymy aspects of negation, in the larger goal of its integration into the thematic analysis. We present a model based on the idea of symmetry compatible with conceptual vectors. Then, we define antonymy functions which allows the construction of an antonymous vector and the enumeration of its potentially antinomic lexical items. Finally, we introduce a measure which evaluates how a given word is an acceptable antonym for a term.

1 Introduction

Research in meaning representation in NLP is an important problem still addressed through several approaches. The NLP team at LIRMM currently works on thematic and lexical disambiguation text analysis (Laf01). Therefore 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 to expressions. The conceptual vectors learning system automatically defines or revises its vectors according to the following procedure. It takes, as an input, definitions in natural language contained in electronic dictionaries for human usage. These definitions are then fed to a morphosyntactic parser that provides tagging and analysis trees. Trees are then used as an input to a procedure that computes vectors using tree geometry and syntactic functions. Thus,

a kernel of manually indexed terms is necessary for bootstrapping the analysis. The transversal relationships¹, such as synonymy (LP01), antonymy and hyperonymy, that are more or less explicitly mentioned in definitions can be used as a way to globally increase the coherence of vectors. In this paper, we describe a vectorial function of antonymy. This can help to improve the learning system by dealing with negation and antonym tags, as they are often present in definition texts. The antonymy function can also help to find an opposite theme to be used in all generative text applications: opposite ideas research, paraphrase (by negation of the antonym), summary, etc.

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 (SM83) and for meaning representation by the LSI model (DDL⁺90) from latent semantic analysis (LSA) studies in psycholinguistics. In computational linguistics, (Cha90) 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 concepts, it is possible to build vectors (conceptual vectors) and to associate them to lexical items². The hypothesis³ that considers a set of concepts as a generator to language has been long described in (Rog52). Polysemic words combine different vectors corresponding

¹well known as lexical functions (MCP95)

²Lexical 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*.

³that we call thesaurus hypothesis.

to different meanings. This vector approach is based on 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 (Lar92) where 873 concepts are identified). To be consistent with the thesaurus hypothesis, we consider that this set constitutes a generator family for the words and their meanings. This family is probably not free (no proper vectorial base) and as such, any word would project its meaning on it according to the following principle. Let be \mathcal{C} 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 $\langle door \rangle$ could be projected on the following concepts (the *CONCEPT*[intensity] are ordered by decreasing values): $V(\langle door \rangle) = (OPENING[0.8], BARRIER[0.7], LIMIT[0.65], PROXIMITY[0.6], EXTERIOR[0.4], INTERIOR[0.39], \dots$

In practice, the larger \mathcal{C} is, the finer the meaning descriptions are. In return, the computing is less easy: for dense vectors⁴, the enumeration of activated concepts is long and difficult to evaluate. We prefer to select the thematically closest terms, i.e., the *neighbourhood*. For instance, the closest terms ordered by increasing distance to $\langle door \rangle$ are: $\mathcal{V}(\langle door \rangle) = \langle portal \rangle, \langle portiere \rangle, \langle opening \rangle, \langle gate \rangle, \langle barrier \rangle, \dots$

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 (Mor99). 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 $D_A(A, B) = \arccos(Sim(A, B))$. 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 $D_A(A, B) \leq \frac{\pi}{4}$

⁴Dense vectors are those which have very few null coordinates. In practice, by construction, all vectors are dense.

(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).

$$D_A(V(\langle tit \rangle), V(\langle tit \rangle)) = 0 \quad (0)$$

$$D_A(V(\langle tit \rangle), V(\langle bird \rangle)) = 0.55 \quad (31)$$

$$D_A(V(\langle tit \rangle), V(\langle sparrow \rangle)) = 0.35 \quad (20)$$

$$D_A(V(\langle tit \rangle), V(\langle train \rangle)) = 1.28 \quad (73)$$

$$D_A(V(\langle tit \rangle), V(\langle insect \rangle)) = 0.57 \quad (32)$$

The first one has a straightforward interpretation, as a $\langle tit \rangle$ cannot be closer to anything else than itself. The second and the third are not very surprising since a $\langle tit \rangle$ is a kind of $\langle sparrow \rangle$ which is a kind of $\langle bird \rangle$. A $\langle tit \rangle$ has not much in common with a $\langle train \rangle$, which explains a large angle between them. One can wonder why there is 32 degrees angle between $\langle tit \rangle$ and $\langle insect \rangle$, which makes them rather close. If we scrutinise the definition of $\langle tit \rangle$ from which its vector is computed (*Insectivorous passerine bird with colorful feather.*) perhaps the interpretation of these values seems clearer. In effect, the thematic is by no way an ontological 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 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 between 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 71000 items for French and more than 288000 vectors, in which 2000 items are concerned by antonymy. These items are either defined through negative sentences, or because antonyms are directly in the dictionary. Example of a negative definition:

‘*non-existence*’: property of what does not exist. Example of a definition stating antonym: ‘*love*’: antonyms: ‘*disgust*’, ‘*aversion*’.

3 Definition and Characterisation of Antonymy

We propose 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 kinds 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. We can distinguish different sorts of media: (i) a property that shows scalar values (*hot* and *cold* which are symmetrical values of temperature), (ii) the true-false relevance or application of a property (e.g. *existence/non-existence*) (iii) cultural symmetry or opposition (e.g. *sun/moon*). From the point of view of lexical functions, if we compare synonymy and antonymy, we can say that synonymy is the research 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 types of symmetry by relying on (Lyo77), (Pal76) and (Mue97). Each symmetry type characterises one particular type of antonymy. In this paper, for the sake of clarity and precision, we expose only the complementary antonymy. The same method is used for the other types of antonymy, only the list of antonymous concepts are different.

3.1 Complementary Antonymy

The complementary antonyms are couples like *event/unevent*, *presence/absence*.

he is present \Rightarrow he is not absent
 he is absent \Rightarrow he is not present
 he is not absent \Rightarrow he is present
 he is not present \Rightarrow he is absent

In logical terms, we would have:

$$\begin{array}{ll} \forall x \ P(x) \Rightarrow \neg Q(x) & \forall x \ \neg P(x) \Rightarrow Q(x) \\ \forall x \ Q(x) \Rightarrow \neg P(x) & \forall x \ \neg Q(x) \Rightarrow P(x) \end{array}$$

This corresponds to the exclusive disjunction relation. In this frame, the assertion of one

of the terms implies the negation of the other. 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 combinations).

4 Antonymy Functions

4.1 Principles and Definitions.

The aim of our work is to create a function that would improve the learning system by simulating antonymy. In the following, we will be mainly interested in antonym generation, which gives a good satisfaction clue for these functions. We present a function which, for a given lexical item, gives the n closest antonyms as the neighbourhood function \mathcal{V} provides the n closest items of a vector. In order to know which particular meaning of the word we want to oppose, we have to assess by what context meaning has to be constrained. However, context is not always sufficient to give a symmetry axis for antonymy. Let us consider the item ‘*father*’. In the ‘*family*’ context, it can be opposite to ‘*mother*’ or to ‘*children*’ being therefore ambiguous because ‘*mother*’ and ‘*children*’ are by no way similar items. It should be useful, when context cannot be used as a symmetry axis, to refine the context with a conceptual vector which is considered as the referent. In our example, we should take as referent ‘*filiation*’, and thus the antonym would be ‘*children*’ or the specialised similar terms (e.g. ‘*sons*’, ‘*daughters*’) ‘*marriage*’ or ‘*masculine*’ and thus the antonym would be ‘*mother*’.

The function $AntiLex_S$ returns the n closest antonyms of the word A in the context defined by C and in reference to the word R.

$$AntiLex_S(A, C, R, n)$$

$$AntiLex_R(A, C, n) = AntiLex_S(A, C, C, n)$$

$$AntiLex_B(A, R, n) = AntiLex_S(A, R, R, n)$$

$$AntiLex_A(A, n) = AntiLex_S(A, A, A, n)$$

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 determine a symmetry axis rather than a context. In practice, we have $AntiLex_B = AntiLex_R$. The last function is

the *absolute antonymy function*. For polysemic words, its usage is delicate because only one word defines at the same time three things: the word we oppose, the context and the referent. This increases the probability to get unsatisfactory results. However, given usage habits, we should admit that, practically, this function will be the most used. It's sequence process is presented in picture 1. We note $Anti(A,C)$ the

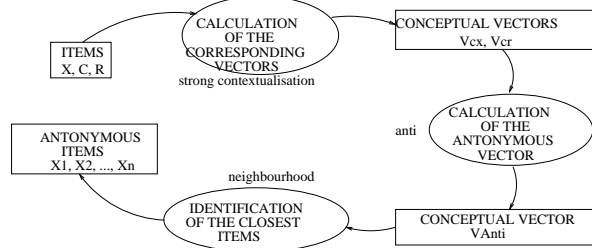


Figure 1: run of the functions *AntiLex*

antonymy function at the vector level. Here, A is the vector we want to oppose and C the context vector.

Items without antonyms: 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. When symmetry is at stake, then fixed points or plans are always present. We consider the case of these objects, and in general, non opposable terms, as belonging to the fixed space of the symmetry. This allows to redirect the question of antonymy to the opposable properties of the concerned object. For instance, if we want to compute the antonym of a ‘*motorcycle*’, which is a *ROAD TRANSPORT*, its opposable properties being *NOISY* and *FAST*, we consider its category (i.e. *ROAD TRANSPORT*) as a fixed point, and we will look for a road transport (*SILENCIOUS* and *SLOW*), something like a ‘*bicycle*’ or an ‘*electric car*’. With this method, thanks to the fixed points of symmetry, opposed “ideas” or antonyms, not obvious to the reader, could be discovered.

4.2 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)*.

4.2.1 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:

$$\begin{aligned}
 AntiC(EXISTENCE, V) &= V(NON-EXISTENCE) \\
 AntiC(NON-EXISTENCE, V) &= V(EXISTENCE) \\
 AntiC(AGITATION, V) &= V(INERTIA) \oplus V(REST) \\
 AntiC(PLAY, V) &= V(PLAY) \\
 \forall V \\
 AntiC(ORDER, V(order) \oplus V(disorder)) &= \\
 V(DISORDER) \\
 AntiC(ORDER, V(classification) \oplus V(order)) &= \\
 V(CLASSIFICATION)
 \end{aligned}$$

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 antonyms *PRESERVATION*, *CONSTRUCTION*, *REPARATION* or *PROTECTION*. So, we have defined for each concept, one conceptual vector which allows the selection of the best antonym according to the situation. For example, 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.

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

4.3.1 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 will see in 4.4.1 that this may cause real problems because of sense selection. We should stress now on the construction of the antonym vector from two conceptual vectors: V_{item} , for

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

the item we want to oppose and the other, V_c , for the context (referent).

4.3.2 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:

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

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

We crafted the definition of the weight P after several experiments. We noticed that the function couldn't be symmetric (we cannot reasonably have $Anti_R(V(\langle hot \rangle), V(\langle temperature \rangle)) = Anti_R(V(\langle temperature \rangle), V(\langle hot \rangle))$). 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 conceptual⁶ the vector is, the more important this power should be. That is why the power is the variation coefficient⁷ which is a good clue for "conceptuality". To finish, we introduce this function \max because an idea presents 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 $\langle cold \rangle$ in the $\langle temperature \rangle$ context, the weight of $\langle cold \rangle$ has to be important even if it is not present in $\langle temperature \rangle$.

4.4 Lexical Items and Vectors: Problem and Solutions

The goal of the functions $AntiLex$ is to return antonym of a lexical item. They are defined with the $Anti$ function. So, we have to use tools which allow the passage between lexical items and vectors. This transition is difficult because of polysemy, i.e. how to choose the right relation between an item and a vector. In other words, how to choose the good meaning of the word.

4.4.1 Transition lexical items \rightarrow Conceptual Vectors

As said before, antonymy is relative to a context. In some cases, this context cannot be sufficient to select a symmetry axis for antonymy.

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

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

To catch the searched meaning of the item and, if it is different from the context, to catch the selection of the meaning of the referent, we use the strong contextualisation method. It computes, for a given item, a vector. In this vector, some meanings are favoured against others according to the context. Like this, the context vector is also contextualised.

This contextualisation shows the problem caused by the absolute antonymy function $Anti_{\alpha_R}$. In this case, the method will compute the vector of the word *item* in the context *item*. This is not a problem if *item* has only one definition because, in this case, the strong contextualisation has no effect. Otherwise, the returned conceptual vector will stress on the main idea it contains which one is not necessary the appropriate one.

4.4.2 Transition Conceptual Vectors \rightarrow Lexical Items

This transition is easier. We just have to compute the neighbourhood of the antonym vector V_{ant} to obtain the items which are in thematic antonymy with V_{item} . With this method, we have, for instance:

$$\mathcal{V}(Anti_{c_R}(\text{death}, \langle death \rangle \& \langle life \rangle)) = (\text{LIFE } 0.4) (\langle killer \rangle 0.449) (\langle murderer \rangle 0.467) (\langle blood sucker \rangle 0.471) (\langle strige \rangle 0.471) (\langle to die \rangle 0.484) (\langle to live \rangle 0.486)$$

$$\mathcal{V}(Anti_{c_R}(\text{life}, \langle death \rangle \& \langle life \rangle)) = (\langle death \rangle 0.336) (\text{DEATH } 0.357) (\langle murdered \rangle 0.367) (\langle killer \rangle 0.377) (\text{C3:AGE OF LIFE } 0.481) (\langle tyrannicide \rangle 0.516) (\langle to kill \rangle 0.579) (\langle dead \rangle 0.582)$$

$$\mathcal{V}(AntiC_{c_A}(\text{LIFE})) = (\text{DEATH } 0.034) (\langle death \rangle 0.427) (\text{C3:AGE OF LIFE } 0.551) (\langle killer \rangle 0.568) (\langle mureded \rangle 0.588) (\langle tyrannicide \rangle 0.699) (\text{C2:HUMAN } 0.737) (\langle to kill \rangle 0.748) (\langle dead \rangle 0.77)$$

It is not important to contextualise the concept *LIFE* because we can consider that, for every context, the opposite vector is the same. In complementary antonymy, the closest item is *DEATH*. This result looks satisfactory. We can see that the distance between the antonymy vector and *DEATH* is not null. It is because our method is not and cannot be an exact method. The goal of our function is to build the best (closest) antonymy vector it is possible to have. The construction of the generative vectors is the second explanation. Generative vectors are interdependent. Their construction is based on an ontology. To take care of this fact, we don't have

boolean vectors, with which, we would have exactly the same vector. The more polysemic the term is, the farthest the closest item is, as we can see it in the first two examples.

We cannot consider, even if the potential of antonymy measure is correct, the closest lexical item from V_{anti} as the antonym. We have to consider morphological features. Simply speaking, if the antonym of a verb is wanted, the result would be better if a verb is caught.

4.5 Antonymy Evaluation Measure

Besides computing an antonym vector, it seems relevant to assess whether 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 $Anti_{cR}(A, C)$ and $Anti_{cR}(B, C)$. Thus, we have:

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

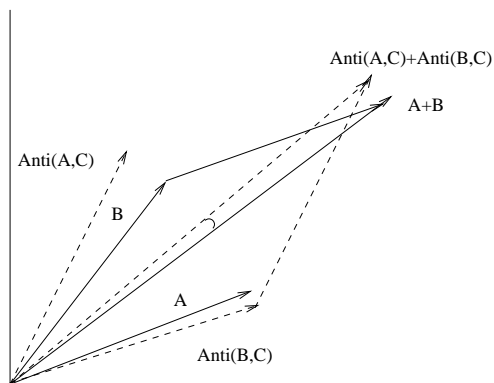


Figure 2: 2D geometric representation of the antonymy evaluation measure $Manti_{Eval}$

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, notwithstanding 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 would be at a distance of about $\frac{\pi}{2}$. Two lexical items at $\frac{\pi}{2}$ have not much in common⁸. 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 .

4.5.1 Examples

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

$Manti_{Eval}(EXISTENCE, NON-EXISTENCE)$	= 0.03
$Manti_{EvalC}(\text{'existence'}, \text{'non-existence'})$	= 0.44
$Manti_{EvalC}(EXISTENCE, CAR)$	= 1.45
$Manti_{EvalC}(\text{'existence'}, \text{'car'})$	= 1.06
$Manti_{EvalC}(CAR, CAR)$	= 0.006
$Manti_{EvalC}(\text{'car'}, \text{'car'})$	= 0.407

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 $D_A(\text{existence}, \text{car}) = 1.464$). We could consider of using this function to look in a conceptual lexicon for the best antonyms. However, the computation cost (around a minute on a P4 at 1.3 GHz) would be prohibitive.

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

5 Action on learning and method evaluation

The function is now used in the learning process. We can use the evaluation measure to show the increase of coherence between terms:

$Manti_{EvalC}$	new	old
⟨existence⟩, ⟨non-existence⟩	0.33	0.44
⟨existence⟩, ⟨car⟩	1.1	1.06
⟨car⟩, ⟨car⟩	0.3	0,407

There is no change in concepts because they are not learned. In the opposite, the antonymy evaluation measure is better on items. The exemple shows that ⟨existence⟩ and ⟨non-existence⟩ have been largely modified. Now, the two items are stronger antonyms than before and the vector basis is more coherent. Of course, we can test these results on the 71000 lexical items which have been modified more or less directly by the antonymy function. We have run the test on about 10% of the concerned items and found an improvement of the angular distance through $Manti_{EvalC}$ ranking to 0.1 radian.

6 Conclusion

This paper has presented a model of antonymy using the formalism of conceptual vectors. Our aim was to be able: (1) to spot antonymy if it was not given in definition and thus provide an antonym as a result, (2) to use antonyms (discovered or given) to control or to ensure the coherence of an item vector, build by learning, which could be corrupted. In NLP, antonymy is a pivotal aspect, its major applications are thematic analysis of texts, construction of large lexical databases and word sense disambiguation. We grounded our research on a computable linguistic theory being tractable with vectors for computational sake. This preliminary work on antonymy has also been conducted under the spotlight of symmetry, and allowed us to express antonymy in terms of conceptual vectors. These functions allow, from a vector and some contextual information, to compute an antonym vector. Some extensions have also been proposed so that these functions may be defined and usable from lexical items. A measure has been identified to assess the level of antonymy between two items. The antonym vector construction is necessary for the selection of opposed lexical items

in text generation. It also determines opposite ideas in some negation cases in analysis.

Many improvements are still possible, the first of them being revision of the VAC lists. These lists have been manually constructed by a reduced group of persons and should widely be validated and expanded especially by linguists. We are currently working on possible improvements of results through learning on a corpora.

References

- Jacques Chauché. Détermination sémantique en analyse structurale : une expérience basée sur une définition de distance. *TAL Information*, 1990.
- Scott C. Deerwester, Susan T. Dumais, Thomas K. Landauer, George W. Furnas, and Richard A. Harshman. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41(6):391–407, 1990.
- Mathieu Lafourcade. Lexical sorting and lexical transfer by conceptual vectors. In *Proceeding of the First International Workshop on MultiMedia Annotation*, Tokyo, January 2001.
- Larousse. *Thésaurus Larousse - des idées aux mots, des mots aux idées*. Larousse, 1992.
- Mathieu Lafourcade and Violaine Prince. Synonymies et vecteurs conceptuels. In *actes de TALN'2001*, Tours, France, July 2001.
- John Lyons. *Semantics*. Cambridge University Press, 1977.
- Igor Mel'čuk, André Clas, and Alain Polguère. *Introduction à la lexicologie explicative et combinatoire*. Duculot, 1995.
- Emmanuel Morin. *Extraction de liens sémantiques entre termes à partir de corpus techniques*. PhD thesis, Université de Nantes, 1999.
- Victoria Lynn Muehleisen. *Antonymy and semantic range in english*. PhD thesis, Northwestern university, 1997.
- F.R. Palmer. *Semantics : a new introduction*. Cambridge University Press, 1976.
- P. Roget. *Roget's Thesaurus of English Words and Phrases*. Longman, London, 1852.
- Gerard Salton and Michael McGill. *Introduction to Modern Information Retrieval*. McGrawHill, 1983.