Automatic Detection of Semantic Primitives with Bio-inspired, Multi-Objective, Weighting Algorithms

Obdulia Pichardo-Lagunas

Instituto Politécnico Nacional, UPIITA, Av. IPN, s/n, 07320, Mexico City, Mexico, opichardola@ipn.mx

Grigori Sidorov

Instituto Politécnico Nacional, Centro de Investigación en Computación, Av. Juan de Dios Batiz, s/n, 07320, Mexico City, Mexico, sidorov@cic.ipn.mx

Alexander Gelbukh

Instituto Politécnico Nacional, Centro de Investigación en Computación, Av. Juan de Dios Batiz, s/n, 07320, Mexico City, Mexico, gelbukh@cic.ipn.mx

Nareli Cruz-Cortés

Instituto Politécnico Nacional, Centro de Investigación en Computación, Av. Juan de Dios Batiz, s/n, 07320, Mexico City, Mexico, nareli@cic.ipn.mx

Alicia Martínez-Rebollar

Centro Nacional de Desarrollo Tecnológico en Cómputo (CENIDET), Interior Internada Palmira, s/n, Palmira, 62490, Cuernavaca, Mexico, amartinez@cenidet.edu.mx

Abstract: This paper proposes the usage of computational techniques that allow for automatic analysis of the vocabulary contained in an explanatory dictionary. It is proposed for the extraction of a set of words, called semantic primitives, which are considered those allowing the creation of a system used to establish definitions in dictionaries. The proposed approach is based on the representation of a dictionary as a directed graph and the combination of a multi-objective differential evolution algorithm with the PageRank weighting algorithm. The differential evolution algorithm extracted a set of primitives that fulfill two objectives: minimize the set size and maximize its degree of representation (PageRank), allowing the creation of a computational dictionary without cycles in its definitions. We experimented with a RAE dictionary of Spanish. Our results present improvement over other algorithms that are representative of the state-of-the-art.

Keywords: lexicography; computational lexicography; semantic primitives; defining vocabulary; explanatory dictionary; multiobjective bioinspired algorithms; differential evolution; weighting algorithms; PageRank.

1 Introduction

Traditional explanatory dictionaries are aimed at human readers. However, if an explanatory dictionary is to be used by computers, some important differences must be considered. Dictionaries for computers are important mainly because a large number of problems related to Computational Linguistics (CL) need to be addressed. Some of those problems are automatic translations and the generation of abstracts and the alignment of texts, among many others. In all these tasks we deal with *semantics*, therefore, it is quite beneficial to use vocabularies containing pre-coded information about deep relations among words and not only the isolated words.

The automatic construction of dictionaries for computer use is usually done starting from a traditional explanatory dictionary. However, traditional dictionaries have a major problem, that is, the existence of *cycles* in their definitions. Actually, in every traditional dictionary, the existence of cycles in the definitions is unavoidable, since the words are explained by cross references to another words reached in one or more steps. For example, we can define *treaty* as *pact*, *pact* as *agreement* and *agreement* as *treaty*, thus, returning to the first word in a two-step cycle. The idea behind dictionaries for humans is that their cycles should be as large as possible, then, it is probable that the person knows at least one of the words in the cycle. In this sense, the longer the paths, the better for humans. On the other hand, the dictionaries for computers cannot have cycles, because computers are not able to process them. So, when designing dictionaries for computers the main problem faced is how to break every cycle.

A dictionary can be represented as a directed graph. That is, for a determined entry the out arrows correspond to words in its definition. We will discuss formal representation of this idea in Section 2.

It is obvious that definitions contained in any dictionary are created using other words, but not all words are considered as the same category, actually there are special sets, i. e., words are considered either defining vocabulary or semantic primitives.

A defining vocabulary is a set of words with which the definitions are created in a dictionary. For example, the Longman vocabulary [9] (that was created and revised by human lexicographers), has about 3,000 words. If we consider a representation of the dictionary as a directed graph, then the words of the

Longman defining vocabulary would lie one step of distance from the dictionary entries, i.e., they are the words used in the definitions.

Conversely, the semantic primitives, named by Wierzbicka (1980, 1996) [15], [16] is the set of words characterized by lack of definition, i.e., the out arrows were removed from the graph, then, they guarantee that the graph has no cycles. Obviously, from each entry we reach only a small set of semantic primitives, not all of them.

This work aims to automatically extract a set of words considered semantic primitives, to create a dictionary without cycles for various CL tasks. In general, we consider that the smallest set of semantic primitives is the best one.

In the development of this work, we based on the following approaches: the hypothesis of the existence of a natural semantic meta-language [15]; the representation of the dictionary as a directed graph [11]; and the usage of an evolutionary algorithm for detecting semantic primitives [10]. We complement our previous works [10, 17] with the design of a multi-objective function for the algorithm Differential Evolution, and the usage of weights assigned by the PageRank algorithm [8] for semantic primitives identification.

The paper is organized as follows. Related works are discussed in Section 2. The proposed method is explained in Section 3. The validation of the experiments with the multi-objective function and the PageRank algorithm are shown in Section 4. Conclusions and future work are presented at the end of the paper.

2 Theoretical Framework

2.1 Related Work

The hypothesis proposed by Anna Wierzbicka [15], [16] claims the existence of a natural semantic meta-language (NSM), which is a vocabulary used to complete the lexicon of any language. Wierzbicka proposed a number of 60 words to be considered as primitives, they represent an irreducible semantic nucleus and that are used (with an additional set of rules) to generate new definitions. The core of this meta-language (the 60 words) is considered universal. Accordingly, the meaning of any expression can be specified through a reductive paraphrase. That is, any complex definition can be described using simpler terms than the original.

Apresjan (1995) [4] supports the idea of using restricted vocabularies for the development of lexicon, but he states that it cannot be as small as mentioned by Wierzbicka.

The Longman dictionary of contemporary English (LDOCE) [9] takes up the concept proposed by Apresjan and uses what is called a defining vocabulary.

In this dictionary, all the definitions are constructed using exclusively the restricted vocabulary. The size of this vocabulary is about 3,000 words in its latest version.

Kozima and Furugori (1993) [5] created a semantic network for LDOCE, in which each word of the dictionary is represented by a node, creating a closed system in which all words are defined by the same dictionary. The authors came to the following conclusion: "If there is a defining vocabulary, it corresponds to the dense part of the network, whereas words that are not defining are not linked to each other, therefore they are found on the periphery. This experiment was the first attempt for automatic vocabulary construction.

Rivera-Loza et al. (2003) [11] and Pichardo-Lagunas (2012) [10] returned to this problem using the Anaya dictionary and RAE dictionary for Spanish as cases of study. In both cases, the dictionary was represented as a directed graph, where each node represents a word. The graph was created by inserting word by word avoiding the existence of cycles in the system of definitions. For each iteration, if a word closed a cycle, then, it was considered as semantic primitive. Note that the order in which the words are added to the graph is important, i.e., different input permutations will generate different output sets. Since we look for the smallest set of primitives, our final goal is to find the input permutation that reduces the number of words considered as such.

In Rivera-Loza et al. [11], the entry words order was given by two methods: randomly and frequencies by random voting. The method of random frequencies obtained the best result with a total of 2,246 semantic primitives.

Pichardo-Lagunas et al. [10] proposed the use of heuristic methods, specifically the algorithm differential evolution (DE) adapted for handling permutations. The DE algorithm generated different permutations for constructing the graph and obtaining a total of 2,169 primitives.

There are some theoretical works related to semantic primitives for the English language, however, for the best of our knowledge, no other work for the automatic semantic primitives' detection is known to date.

2.2 Graph Theory Concepts

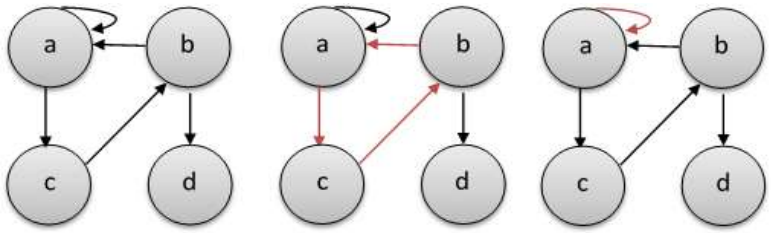
A directed graph G is a tuple $G = (V, F)$, where $V \neq \emptyset$, whose elements are called vertices, $F \subseteq V \times V$. The elements of F are called directed edges, see Figure 1(a).

A directed path in G is a finite sequence of vertices of G denoting:

$$V1, V2, \ldots, Vn.$$

Definitions:

- A directed path T is closed if and only if $V1 = Vn$,
- A closed directed path is a directed cycle or cycle,
- A semantic primitive is the vertex V that closes the directed path.



(a) (b) (c)

Figure 1
(a) Directed graph, (b) Cycle in a directed graph, (c) Loop in a directed graph

The above cases are shown in Figure 1(a), (b) and (c). Figure 1(a) represents an example of a directed graph. Figure 1(b) contains a cycle "$a \rightarrow c \rightarrow b \rightarrow a$". Figure 1(c) contains another cycle "$a \rightarrow a$" (the loop).

Let $G = (V, F)$ be a directed graph, then $G' = (V', F')$ is a subgraph of G if $V' \neq \emptyset$ and $F' \subset F$, where $\forall$ edge of F' is incident to the vertices of V'.

2.3 Multi-Objective Optimization

Multi-objective optimization attempts to find a solution vector that simultaneously optimizes more than one objective function. These functions typically are in conflict to each other, which means that improvement in one function makes worse the performance of the other ones. Multi-objective optimization can be mathematically defined as:

Find the vector $\vec{x}^*$ that optimizes the target function vector

$$f_{1(\vec{x})}, f_{2(\vec{x})}, \dots f_{k(\vec{x})}$$

subjected to m inequality constraints

$$g_i = (\vec{x}) \leq 0; i = 1, \dots, m,$$

and p equality constraints

$$h_i(\vec{x}) = 0; i = 1, \dots, p.$$

2.3.1 Pareto Optimum

The Pareto Optimum is a set of solutions that reaches a compromise among the different objective functions. A formal definition is as follows:

A decision vector of variables $\vec{x}^* \in F$ (where F is the feasible area) is Pareto optimal if there is no other $\vec{x} \in F$ such that: $f_i(\vec{x}) \leq f_i(\vec{x}^*)$ for all $i = 1,.., k$ and $f_j(\vec{x}) \leq f_j(\vec{x}^*)$ for at least one j.

In other words, "Pareto optimum is that vector of variables, in which the solutions of the problem cannot be improved in one objective function without worsening any of the others" (Abbass, 2002) [2].

The Pareto optimum provides a set of solutions called Pareto Optimal Set.

2.3.2 Pareto Dominance

The term Pareto Dominance can be defined as follows:

$A\ vector\ \vec{u} = (u_1, \ldots, u_k)\ dominates\ another\ vector\ \vec{v} = (v_1, \ldots, v_k)$
$if\ and\ only\ if\ it\ dominates\ anoter\ \vec{u}\ which\ is\ partially\ smaller\ than\ \vec{v}.$

For example, when comparing two different solutions A and B, there are three possible situations:

- A dominates B,
- A is dominated by B,
- A and B are not dominated to each other.

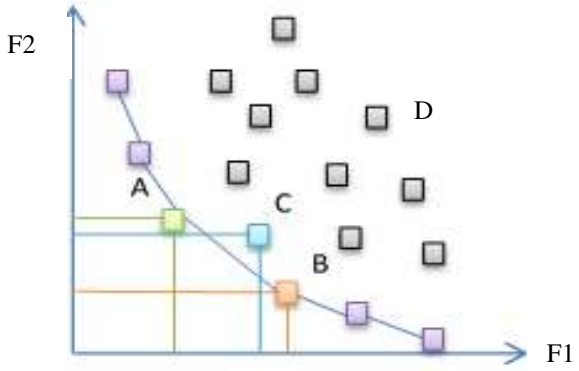


Figure 2

Objective function space illustration for two objectives F1 and F2

See for example in Figure 2, an illustration of two-objective functions F1 and F2. The solution B dominates to all the solutions represented by grey squares because B has smaller values for F1 and F2 than all of them. Further, A and B are not dominated to each other because B has smaller value for F2 but A has a smaller value in F1. The point D is dominated by A because is smaller in F1 and F2 than D.

2.3.3 Pareto Front

The solutions whose vectors are not dominated and are also in the Pareto optimal set are called the Pareto front. The formal definition is as follows:

For a given multi-objective problem $\vec{f}(x)$ and a set of Pareto optimal P^*, the Pareto front (FP^*) is:

$$FP^* := \left\{\vec{f} = [f_1(x), \ldots, f_k(x)] \mid x \in P^*\right\}.$$

2.4 Differential Evolution Algorithm

The differential evolution (DE) is a population-based evolutionary algorithm, developed for optimization in continuous spaces [13].

The general DE idea is as follows: The initial population is a set of real numbers randomly generated and stored in a vector. Then, three individuals are selected to play the role of parents. One of the candidates is the main father and is altered with information taken from the other two parents. If the resulting value (solution) from the previous operation is better than the current individual, then it is replaced. Otherwise, the parent is retained. The process is repeated until a determined criterion is reached.

As mentioned earlier, the DE algorithm was designed to work with potential solutions represented by real numbers. In the problem that is being addressed, we look for solutions with representation of permutations, so we used an adaptation that allowed us to convert the representation of permutations into real numbers [14]. Next, there is an example:

The vector solutions are a permutation of the integers from 1 to 5. Given two vectors X_{r1} y X_{r2} as follows:

$$X_{r1} = \begin{bmatrix} 1 \\ 3 \\ 4 \\ 5 \\ 2 \end{bmatrix}, \qquad X_{r2} = \begin{bmatrix} 1 \\ 4 \\ 3 \\ 5 \\ 2 \end{bmatrix}.$$

Then we transform them into $X_{r1,f}$ y $X_{r2,f}$. The subscript *f* denotes a floating point representation vector. This way now the vectors are real numbers and the algorithm DE can be directly applied as in its original version.

$$X_{r1,f} = \frac{X_{r1}}{5} = \begin{bmatrix} 0.2 \\ 0.6 \\ 0.8 \\ 1 \\ 0.4 \end{bmatrix}, \qquad X_{r2,f} = \frac{X_{r2}}{5} = \begin{bmatrix} 0.2 \\ 0.8 \\ 0.6 \\ 1 \\ 0.4 \end{bmatrix}.$$

Continuing with the general DE process, a third vector is randomly selected:

$$X_{r3} = \begin{bmatrix} 5 \\ 2 \\ 1 \\ 4 \\ 3 \end{bmatrix} \rightarrow X_{r3,f} \begin{bmatrix} 1 \\ 0.4 \\ 0.2 \\ 0.8 \\ 0.6 \end{bmatrix}.$$

Then the mutation can be as:

$$v_f = X_{r3,f} + F\left(X_{r1,f} - X_{r2,f}\right)^{F=0.85} = \begin{bmatrix} 1 \\ 0.4 \\ 0.2 \\ 0.8 \\ 0.6 \end{bmatrix} + 0.85 \begin{bmatrix} 0 \\ -0.2 \\ 0.2 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0.23 \\ 0.37 \\ 0.8 \\ 0.6 \end{bmatrix}.$$

The resulting vector must be transformed back into integers:

$$v_f = \begin{bmatrix} 1 \\ 0.23 \\ 0.37 \\ 0.8 \\ 0.6 \end{bmatrix} \rightarrow v = \begin{bmatrix} 5 \\ 1 \\ 2 \\ 4 \\ 3 \end{bmatrix},$$

which is an adequate representation of our problem.

Due to the characteristics of the problem to be solved, a multi-objective ED algorithm was implemented, specifically, the Pareto Differential Evolution (PDE) [1], which is a modification of the original ED and whose algorithm is presented below.

Let G denotes a generation, P a population of size M,
and $\vec{x}_{G=k}^{j}$ the j^{th} the individual of
dimensions N in population P in generation k,
and CR denotes the croosover probability
input *$N, M \geq 4, F \in (0,1+), CR \in [0,1]$, and initial:*
bounds: lower (x_i), upper (x_i), $i = 1, \dots N$
initialize $P_{G=0} = \{\vec{x}_{G=0}^{1}, \dots, \vec{x}_{G=0}^{M}\}$ as
For each *individual $j \in P_{G=0}$*
$x_{i,G=0}^{j} = Gaussian\ (0.5, 0.15), i = 1, \dots N$
Repair $\vec{x}_{i,G=k}^{j}$ if any variable is outside its boundaries
end for each

$evaluate\ P_{G=0}$
$\boldsymbol{k = 1}$
$\boldsymbol{while}$ $the\ stopping\ criterion\ is\ not\ satisfied$ $\boldsymbol{do}$
$remove\ all\ dominated\ solutions\ in\ P_{G=k-1}$
$\boldsymbol{if}$ $the\ number\ of\ non\ dominated\ solutions\ in\ P_{G=k-1} > \alpha$
$\boldsymbol{then}$ $apply\ the\ rule\ of\ neighbord\ rule$
$\boldsymbol{end\ if}$
$\boldsymbol{for}\ \boldsymbol{j = 0}$ $to\ the\ number\ of\ non\ dominated\ solutions\ in\ P_{G=k-1}$
$\vec{x}^{j}_{G=k} \leftarrow \vec{x}^{j}_{G=k-1}$
$\boldsymbol{end\ for}$
$\boldsymbol{while\ j \leq M}$
$randomly\ select\ \boldsymbol{r_1, r_2, r_3} \in (\boldsymbol{1}, \dots, \boldsymbol{\alpha}), of\ the$
$non\ denominated\ solutions\ of\ \boldsymbol{P_{G=k-1}}, where\ \boldsymbol{r_1 \neq r_2 \neq r_3}$
$randomly\ select\ \boldsymbol{i_{rand}} \in (\boldsymbol{1}, \dots, \boldsymbol{N})$
$\boldsymbol{for\ all\ i \leq N}, \vec{x}'_{i,G=k} =$

$$\begin{cases} x^{r_3}_{i,G=k-1} + \boldsymbol{Gaussian}\ (\boldsymbol{0}, \boldsymbol{1}) \times (x^{r_1}_{i,G=k-1} - x^{r_2}_{G=k-1} \\ x^{j}_{i,G=k-1} \quad \boldsymbol{otherwise} \end{cases}$$

$\boldsymbol{end\ forall}$
$Repair\ \vec{x}^{j}_{G=k-1}\ if\ each\ variable\ is\ outside\ its\ boundaries$
$\boldsymbol{if}\ \vec{x}' = \boldsymbol{dominates}\ \vec{x}^{r_3}_{G=k-1}\boldsymbol{then}$
$\vec{x}^{j}_{G=k} \leftarrow \vec{x}'$
$\boldsymbol{j = j + 1}$
$\boldsymbol{end\ if}$
$\boldsymbol{k = k - 1}$
$\boldsymbol{end\ while}$
$\boldsymbol{return\ non\ dominated\ solutions}$

The next considerations were applied to the multi-objective algorithm:

1. The initial population is generated with a Gaussian distribution N (0.5, 0.15).
2. The parameter F is generated with a Gaussian distribution N (0, 1).
3. Reproduction is performed only with non-dominated solutions at each generation.
4. Limits on the variables are preserved by changing its sign, if it is less than 0, or subtracting 1 if it is greater than 1, until the variable is within the allowed limits.
5. A generated individual is placed in the population if he dominates his father.

The multi-objective DE algorithm is summarized as follows. An initial population is generated, all dominated solutions are removed from the population and the rest are used for reproduction. Three parents are randomly selected to generate a child. The offspring is placed in the population if it dominates the main father, otherwise he is forgotten. This process is repeated until the population is complete [12].

2.5 PageRank Algorithm

The PageRank algorithm was proposed by Larry Page and Sergey Brin (1998) [8] and is used to assign a numerical value that corresponds to the relevance of the different web pages that can be indexed by the search engines.

The PageRank algorithm is based on a democratic system that uses the link system as an indicator of the relevance of a particular webpage. Google interprets the links between pages as votes considering also the relevance of the page that contains the league. That is, the votes of a relevant page are more important than those of a page with less relevance. The algorithm at the beginning assigns random values and then iterates until no changes are produced.

The PageRank algorithm is described as follows:

$$PR(A) = (1-d) + d\sum_{i=1}^{n}\frac{PR(i)}{C(i)},$$

where PR(A) is the PageRank of page A, d is damping factor having a value between 0 and 1 (usually, 0.85), PR(i) are the PageRank values that have each page i that has links to A (incoming links), C(i) is the total number of outgoing links of the page i (whether or not to A).

3 Proposed Method for Automatic Detection of Semantic Primitives

The approach of this research is divided in three stages: (1) preprocessing, which consists of the dictionary debugging and the construction of the graph, (2) execution of the PageRank algorithm, that serves to weight the nodes that belong to the graph and (3) application of the algorithm of Differential Evolution, that determines different input permutations to obtain the set of semantic primitives and the construction of the dictionary without cycles.

3.1 Preprocessing and PageRank Algorithm

For the experiments, the dictionary of the Royal Spanish Academy (RAE for its acronym in Spanish), edition of 2007 was used. The RAE has a total of 152,370

entries. As it is common in computational linguistics, we ignore stopwords (like prepositions, conjunctions, etc.), i.e., we only used the content words: verbs, adverbs, nouns and adjectives. To identify the content words, the dictionary was tagged using the Freeling tool [7].

The description of words that have more than one meaning were grouped into a bag of words, that is, even if a word has more than one meaning, in the graph it was represented only once. We plan to take into account word senses in our future work.

Words that were contained in their own definition (that is, those that make loops or cycles) were detected and were considered as semantic primitives.

Words not used in other definitions were not added to the graph because they would not have the possibility of closing some cycle and therefore have no opportunity to be considered as semantic primitives. This process was done iteratively because each time a set of words were deleted some other words were no longer used in the set of definitions.

Once the list of dictionary entries was generated, a number was assigned to each of them, which functions as an index. The index identifies the word and allows the evolutionary algorithm to work only with numbers and not with the string of characters. With the indexes already assigned, the adjacency list representing the dictionary was generated as a graph.

Preprocessing was the same as that used in Rivera-Loza et al. [11] and Pichardo-Lagunas et al. [10]. Summarizing, it includes the following steps:

1. Delete additional information (like the origin, for example, “from Latin *Ab*”).
2. Remove prefixes and suffixes from dictionary entries.
3. Delete entries contained in your own definition.
4. Remove entries that are not used in the rest of the definitions.
5. Tag dictionary words using Freeling.
6. Remove stopwords from entries and definitions.
7. Remove entries that are not used in definitions.

The adjacency list generated after the preprocessing was used as the input in the PageRank algorithm. The algorithm calculated the weighting of each node according to the relations that it maintains with the rest of the nodes. The information provided by the PageRank algorithm serves to evaluate the second objective of the PDE function that seeks to maximize the sum of the weighting associated to each of the nodes of the extracted set.

3.2 Pareto Differential Evolution (PDE)

In the context of the problem, it is necessary to construct the graph *G'*, which is a sub graph of *G,* where *G* is the preprocessed dictionary. The sub graph *G'* is constructed by inserting node after node verifying that it keeps without cycles. Thus, we try different input permutations.

The Pareto Differential Evolution looks for a permutation σ that is an input permutation for the graph G', which is constructed by inserting node by node (according to σ). With each insertion in *G'* it is verified that no cycle is generated between the definitions of the graph, if so, the vertex is not inserted and it is considered a semantic primitive.

To apply the algorithm of differential evolution it is required:

- Representation of possible solutions (permutations),
- Creation of an initial population of possible solutions (random values),
- Definition of the evaluation function (fitness function),
- Other parameters, such as:
    1. Population size,
    2. Probability of crossover,
    3. Probability of mutation,
    4. Maximum number of generations.

The PDE algorithm requires as input a list of indices of words:

$$x_{i=0}^{m} = \{(I1, j_1), (I2, j_2) \dots (In, j_n)\},$$

where *n* is the total of entries in the dictionary, *m* is the total of vectors in the population, *j* is the weight associated to the node, and the vector x^m is a permutation.

3.3 Fitness Function

The fitness of an individual is measured according to the Pareto dominance criterion (see 2.3.1), which is determined by evaluating the objective function in each set. The objective function for our problem is defined as follows:

$$\begin{cases} Minimize|P|, where\ p = \{x | x \in V \wedge x \notin G'\}, \\ Maximize\ S, where\ s = \left\{ \sum_{i=1}^{n} PR(p_i) \right\}. \end{cases}$$

The first objective function seeks to minimize P where p is a set of nodes belonging to V (which is the set of words of G) and which do not belong to G'. Where G is the complete graph and G' is the graph constructed without cycles. At the same time, the second objective function seeks to maximize S, where s is the sum of the weights obtained by PageRank associated with the p_i that represents the individual.

4 Experiments and Results

We conducted our experiments using the most influential RAE dictionary of Spanish. The RAE dictionary has 152,370 words with definitions. After the preprocessing tasks, this number is reduced to 77,300. The generated list was used as input for the execution of PageRank algorithm with three different parameters. In the first case, we used 70 iterations and the damping factor of 0.75. The second run used 50 iterations and 0.8 as the damping factor. The third case applied 100 iterations and the damping factor of 0.85, which are the parameters specified by Page and Brin (1996) [8].

Although the values generated by the algorithm varied in each case, the average difference remained within the range of ±3.0%, so we used the list generated by the third configuration.

The vector with the indices of identification and the weighting associated to each node served as the input for the algorithm of Pareto Differential Evolution. The proposed algorithm obtains given a directed graph G, a defining subset P, where $P \subseteq V$ and each p from P is considered semantic primitive, since any cycle in the graph G contains a vertex to P.

The first purpose of the objective function is to minimize the set of semantic primitives, seeking to maintain the G' graph with as few words as possible. The other objective is to maximize the PageRank value of P.

The PDE algorithm was executed 30 times [6]. In each of the executions different configuration parameters were used for the algorithm.

The configuration of parameters that obtained the best results was:

- 500 individuals,
- 300 generations,
- Probability of crossover: 0.2.

In each iteration, a set of non-dominated solutions of different sizes was obtained, but as proposed in Santana-Quintero (2004) [12] the final size of the set was reduced to 50 using the neighborhood distance function.

It was obtained a set of 50 non-dominated solutions. The one that is identified as the best solution is that obtained the least number of semantic primitives.

Table 1
Runs with best results

	Number of individuals	Number of generations	Probability of crossover	Number of primitives
Run 17	300	500	0.16	2,234
Run 23	500	500	0.1	2,252
Run 24	300	500	0.2	2,228
Run 30	500	300	0.2	**2,148**

The iteration with the best results obtained a total of 2,148 primitives with a sum PageRank value of 1,776.52. The smallest set was selected because the essential objective of this research is to find the set with the least number of words. The system also generated sets of words that obtained higher PageRank values, but for sets with greater size. These sets should be subjected to analysis in later works.

Table 2
Values of Page Rank for some runs

	Number of primitives	Sum of PageRank values
Run 17	2,234	1,762.179
Run 23	2,252	1,770.031
Run 24	2,228	1,785.185
Run 30	**2,148**	1,776.522

To carry out the validation between the obtained set and the complete vocabulary, we used an automatic translation of Longman vocabulary from LDOCE. A coincidence of the word from our set with this vocabulary means that at least one of the meanings of the translations of the English word coincides with at least one of the meanings of a word of the generated set in Spanish. The absolute coincidences with LDOCE are calculated by dividing the number of primitives that are ate same time present in LDOCE by the size of LDOCE. This measure shows which part of LDOCE is covered by the obtained set of primitives. The relative coincidences are calculated by dividing the number of primitives in LDOCE by the size of the obtained set of primitives. This measure shows which part of the set of primitive belongs to LDOCE.

Table 3
Comparison of matches with LDOCE vocabulary

	Number of primitives	Relative coincidences with LDOCE	Absolute coincidences with LDOCE
Pichardo-Lagunas *et al.*	2,169	1,594 (56.05%)	73.87%
Rivera-Loza *et al.*	2,246	1,487 (52.15%)	66.20%
ED Pareto	2,148	1,719 (80.02%)	72.95%

Considering the work done by Rivera-Loza et al. [11] and Pichardo-Lagunas et al. [10], a comparison was made between them and the results obtained by the multi-objective function presented in this work. Pichardo-Lagunas et al. obtained a 73.87% coincidence with the vocabulary Longman and the set obtained by PDE reached 72.95%, with a difference of 1%. As compared to Rivera-Loza et al.'s work, an improvement of 6.75% was obtained. It should be noticed that the relative coincidences with LDOCE of the proposed method augmented about 25%.

Conclusions

For the experiments carried out in other works, such as, Rivera-Loza et al. and Pichardo-Lagunas, the number of obtained primitives shows a certain level of stability.

The Pareto Differential Evolution algorithm (PDE) was applied, which improved the results obtained by previous works by 1.02% with respect to the size of the obtained set and the relative coincidences with LDOCE augmented to about 25%. Thus, including the importance of words (nodes) as an evaluation parameter (application of the PageRank algorithm), the sets of the primitives tend to decrease their size and the nodes of the obtained sets have relations of major importance within the graph.

Acknowledgements

This work was partially supported by the Mexican Government (CONACYT project 240844, SNI, COFAA-IPN, SIP-IPN 20161947, 20161958, 20162204, 20162064).

References

[1] Abbass, H. & Sarker, R. (2002). The Pareto Differential Evolution Algorithm. International Journal on Artificial Intelligence Tools, 11(4):531−552

[2] Abbass, H. (2002). The Self-Adaptive Pareto Differential Evolution Algorithm. Congress on Evolutionary Computation CEC'2002. Volume 1, 831−836, Piscataway, New Jersey.

[3] Abbass, H., Sarker, R. & Newton, C. (2001). PDE: A Pareto-frontier Differential Evolution Approach for Multi-objective Optimization Problems. Proceedings of the Congress on Evolutionary Computation, Vol. 2, New Jersey, 971−978.

[4] Apresjan, J. (1995). Selected works (in Russian). Moscow.

[5] Kozima, H. & Furogori, T. (1993). Similarity between words computed by spreading activation on an English dictionary. Proceedings of the 6th conference of the European chapter of ACL, 232−239.

[6] Levine, D., Berenson, M. & Krehbiel, T. (2006). Estadística para administración. Pearson Education, México.

[7] Padró, L., Collado, M., Reese, S., Lloberes, M. & Castellón, I. (2010). Freeling 2.1: Five years of open-source language processing tools. Proceedings of the 7th Language Resources and Evaluation Conference, La Valleta, Malta.

[8] Page, L. & Brin, S. (1998). The anatomy of large-scale hypertextual Web search engine. Computer Networks and ISDN Systems, 30(1-7), 107−117.

[9] Pearson Education (1991). Longman Dictionary of Contemporary English. London.

[10] Pichardo-Lagunas, O. (2012). Detección automática de primitivas semánticas con algoritmos bioinspirados. Tesis de doctorado, CIC-IPN, México.

[11] Rivera-Loza, G., Gelbukh, A. & Sidorov, G. (2003). Selección automática de primitivas semánticas para un diccionario explicativo del idioma español. Tesis de maestría, CIC-IPN, México.

[12] Santana-Quintero, L. (2004). Un algoritmo basado en evolución diferencial para resolver problemas multiobjetivo. Tesis de maestría, CINVESTAV-IPN, México.

[13] Storn, R. & Price, K. (1995). Differential evolution – a simple and efficient adaptative scheme for global optimization over continuous spaces. Technical Report TR-95-12, International Computer Science, Berkeley, California.

[14] Storn, R., Price, K. & Lampinen, J. (2005). Differential Evolution. A practical Approach to Global Optimization. Springer.

[15] Wierzbicka, A. (1980). Lingua Mentalis: The semantics of natural language. New York: Academic Press. xi, 368.

[16] Wierzbicka, A. (1996). Semantics: Primes and Universals. Oxford University. Oxford.

[17] Pichardo-Lagunas, O., Sidorov, G., Cruz-Cortés, N. & Gelbukh, A. (2014). Detección automática de primitivas semánticas en diccionarios explicativos con algoritmos bioinspirados [Automatic detection of semantic primitives in dictionaries using bio-inspired algorithms]. Onomazein, 29:104−117.