Access to a Large Dictionary of Spanish Synonyms: A Tool for Fuzzy Information Retrieval*

Alejandro Sobrino-Cerdeiriña[1],
Santiago Fernández-Lanza[1], and
Jorge Graña-Gil[2]

[1] Departamento de Lógica y Filosofía Moral, Univ. de Santiago de Compostela, Campus Sur s/n, 15782 - Santiago de Compostela, Spain
{lfgalex, sflanza}@usc.es
[2] Departamento de Computación, Universidad de La Coruña, Campus de Elviña s/n, 15071 - La Coruña, Spain
grana@udc.es

Summary. We start by analyzing the role of imprecision in information retrieval in the Web, some theoretical contributions for managing this problem and its presence in search engines, with special emphasis on the use of thesaurus in order to increase the number and relevance of the documents retrieved. We then present a general architecture for implementing large dictionaries in natural language processing applications which is able to store a considerable amount of data relating to the words contained in these dictionaries. In this modelling, efficient access to this information is guaranteed by the use of minimal deterministic acyclic finite-state automata. In addition, we implement a Spanish dictionary of synonyms and illustrate how our general model helps to transform the original dictionary into a computational framework capable of representing semantic relations between words. This process allows us to define synonymy as a gradual relation, which makes the final tool more suitable for word sense disambiguation tasks or for information retrieval applications than other traditional approaches. Moreover, our electronic dictionary, called Fdsa, will be freely available very soon for stand-alone use.

1 Introduction

Nowadays, when there is more and more digital information available, to ask questions and recover their corresponding relevant answers is of great importance. In this task, Web searchers play a fundamental role. Until now, the

* This work was partially supported by the projects TIN2004-07246-C03-02 of the Spanish Ministry of Education and Science, and PGIDTI02PXIB30501PR of the Autonomous Government of Galicia (Spain).

criteria preferred by search engines were of syntactic nature: a document was recovered as a relevant answer when the words used by the engine to index that document fitted with the words introduced by the user in the search box.

However, a search method based on simple syntactic matchings is not enough, and it should include some kind of weighting for the terms to match. Simple matching of words does not ensure relevance of documents respect to queries, since relevance is of a semantic nature. Moreover, relevance is hardly absolute, but a gradual subject. In fact, a document is seldom completely informative or completely devoid of any interest. Usually, documents inform of a theme in a partial way, i.e. to a certain degree. Weighting of words according to their field (in semi-structured texts) or to their position in the document (title, abstract, etc.) has been the most commonly applied method for automatically inferring relevance, as shown in a great number of proposed models for information retrieval, e.g. the classic vectorial model [13] or the TF×IDF weights [9]. Based on the vectorial space model, the relevance score of a document is the sum of weights of the query terms that appear in the document, normalized by the Euclidean vector length of the document. The weight of a term is a function of the word's occurrence frequency (also known as TF, the term frequency) and the number of documents containing the word in collection (also known as IDF, the inverse document frequency).

Nevertheless, weighting of terms is not enough in many cases for an appropriate retrieval. If we need, for instance, information about `powerful cars`, there will be relevant pages about `fast cars`, and this relevance is not given by any weighting of the word `powerful`, but is given by an identification of `powerful` and `fast` as synonymous terms in this context. This intuitive idea arises from our linguistic knowledge, which allows us to consider as relevant pages those including terms explicitly present in the query, and also those including terms semantically related with the query through the synonymy process. In practice, the application of this idea implies expanding the query over a word with its *synset* or set of synonymous terms semantically related.

The use of associative methods for recovering information is not a recent proposal in the field of information retrieval. Query expansion through synonymy is a traditional resort for improving performances, and, since similarity of meanings is a gradual subject, fuzzy logic plays an important role in this task. In fact, there exist various fuzzy thesauri [11, 12] in which, if two words are related, this link presents a degree which points out the strength of their semantic association. This degree is determined by means of statistical data and by validation processes performed by users. Fuzzy thesauri have been used for recovering textual information with good results, even though they show three main problems:

- With the use of this kind of thesauri, the coverage is increased, but this often diminishes precision.

- These thesauri do not distinguish meanings or contexts, and therefore they cause wrong recoveries, increase the latency of the search, and make the list of results redundant or useless.
- For queries with sentences or complex terms, these thesauri do not integrate an appropriate semantic to combine the corresponding simple terms.

In this work, we propose an alternative solution for avoiding these problems. This solution is given by our electronic dictionary of synonyms, which presents the following features:

- Our dictionary mechanizes a Spanish dictionary of synonyms considering, for each word, its possible meanings and, for each meaning, its synset or list of synonymous terms. To consider different meanings of a word is crucial for avoiding erroneous searches. In some way, these meanings associated to the words provide a (partial) ontology, because a meaning indicates the contextual use of a word, and separates this use from the other possible uses. Moreover, our dictionary assigns a degree of synonymy to all the words of the synset. This degree can be calculated by using different definitions of similarity. The use of these different criteria for measuring similarity is important because vagueness of words is plural and can require different models (fuzzy logics) for its formalization, particularly in the case of searches involving sentences or complex terms. Currently, in our dictionary, this functionality must be used manually. In the future, it will be convenient to integrate an automatic method to select a specific similarity measure, depending on the different situations in which vagueness appears. Nevertheless, for this last task, it would be necessary to have a very wide ontology available.
- Our dictionary is also efficient from a computational point of view. With regard to this aspect, to handle an electronic dictionary for the synonymous words of a given language, considering their corresponding meanings, is in fact a complex task and involves a great computational cost. The last part of this work attend to this problem by designing a method to build an acyclic deterministic automaton, which allows us to complete the printed dictionary and to access its contents rapidly.

Since synonymy is a linguistic feature modeled with a gradual logic, such as fuzzy logic [8], it should be a resort used by the well-known searchers. The query `fuzzy search engines` in Google provides, among others, the following results:

- Northern Light (`www.northernlight.com`). It is said that it has a fuzzy *and* because it uses singulars and plurals in the search, giving wider results than a precise searcher (as is described in its technical manual).
- Discount Engine (`www.discountdomainsuk.com/glossary/3/499/0`). It defines fuzzy search as a search that succeeds when matching words that are partially or wrongly spelled. The same definition can be found in

`www.search-marketing.info/search-glossary/engine-terms/fuzzy-search.htm`.

- NETSCAPE (`www.netscape.com`). It uses the fuzzy operator `~`, which must be typed after some symbols. Its mission is to force the system to guess what comes after it: for instance, with `acetil~` the searcher should provide all the pages including terms that are compatible with this prefix, even those related with `acetilsalicilic`, the complete name of the term perhaps partially forgotten.

An overview to the rest of URLs provided by this query shows that the word *fuzzy* is applied, in most cases, when the searcher has algorithms to correct wrongly typed strings (GOOGLE itself has this functionality) and, in any case, when the searcher handles linguistic resources such as synonymy.

To correct wrongly spelled words is obviously very useful. However, this approximate matching is only fuzzy in a formal aspect, but not from a semantic point of view. A search is said to be genuinely approximate when it is handled by considering the meaning of the terms involved in it, i.e. by using their representation in the index of the search engine, and the relations established between them attending to their meanings (and, for instance, to their synonyms), and not to the way in which they are written. Our dictionary can be an appropriate tool for performing fuzzy searches in the Web because it seems plausible that:

- To distinguish meanings will reduce the number of pages provided as relevant answers by the searcher.
- To have an efficient implementation does not penalize the latency of the searching proccess.
- To implement different similarity measures will permit progress to be made in the selection of the appropriate operators to combine the terms of the queries.

We are sure that a fuzzy semantic search is a necessary step on the way to obtaining more robust information retrieval systems. However, its implementation in a commercial search engine and an evaluation of the results is a challenge that we are obliged to undertake in the future. For the moment, we restrict the objective of this work to building an electronic dictionary of synonyms for Spanish.

In Sect. 2, we include a brief discussion on synonymy. Section 3 gives our way to treat synonymy and specifies how to calculate the degree of synonymy between two entries of the dictionary. Section 4 describes our general model of electronic dictionary and allows us to understand the role of the finite-state automata in this context. In Sect. 5, we describe the Spanish dictionary of synonyms [2] and detail all the transformations performed on it with the help of our automata-based architecture for dictionaries. As we have seen, our main aim is to integrate this dictionary in natural language processing applications of a more general nature, as a tool able to provide greater precision in the

analysis of synonymy relations. Nevertheless, our electronic dictionary, called FDSA[3], will be available in the very near future for stand-alone use. In Sect. 6, we present its main features and functionalities. Finally, Section 7 presents our conclusions after analysing the data contained in this new dictionary.

2 A short historical introduction to synonymy

Synonymy, or the relationship of similarity of meaning, has long been a subject of interest. The first recorded mention of the concept was made by the Ancient Greek philosopher Prodicus of Keos (465 - 399? B. C.), and Aristotle refers to it in his *Topics* (I 7:103a6-32):

> "... from the outset one should clearly state, with regard to that which is identical, in how many ways it can be said."

This description led to continued interest in the subject of synonymy, and influenced the aim of grouping together as synonyms those words whose meanings, although coinciding, showed certain differences.

The Romans took up this tradition, as is shown by Seleucus' treatise *On the difference between synonyms* and a rudimentary dictionary by Ammonius, *On similar and different expressions*. Both Greeks and Romans alluded to two fundamental traits of synonymy:

1. it is a characteristic of the meaning of words, and deals with the plurality of signifiers of a single reference;
2. it is a relationship of the similarity of meanings.

But the first attempt at a systematic study of synonymy as a lexical relationship was made by the Frenchman Gabriel Girard at the beginning of the 18th century. In his work *La Justesse de la Langue Française, ou les Differents Significations de Mots qui Passent pour Synonymies* (1718), he stated that:

> "In order to obtain propriety, one does not have to be demanding with words; one does not have to imagine at all that so-called synonyms are so with all the rigorousness of perfect resemblance; since this only consists of a principal idea which they all enunciate, rather each one is made different in its own way by an accessorial idea which gives it its own singular character. The similarity bought about by the principal idea thus makes the words synonyms; the difference that stems from the particular idea, which accompanies the general one, means that they are not perfectly so, and that they can be distinguished in the same manner as the different shades of the same colour." [6, pp. VIII ff.]

[3] Fuzzy Dictionary of Synonyms and Antonyms.

Girard clearly considers synonymy more as an approximate relation than a matter of perfect resemblance. This conception still prevails today where a common definition of synonymy is that two expressions are synonyms if they have the same, or approximately the same, meaning. In natural language there are few examples of words that have exactly the same meaning, nevertheless in dictionaries of synonyms there are many examples of words that have approximately the same meaning. The imprecise characteristics of word synonymy in natural language will be studied in this work in terms of the automatisation of a dictionary of synonyms.

As we have seen, this definition of synonymy lies in the concept of meaning. Although it may seem appropriate to analyze this concept, which has been the subject of long-standing controversy in the fields of philosophy of language and linguistics, such an analysis lies outside the scope of the present work, in which we only try to look for a computational way to represent the meaning of a word. Our proposal is to consider the set of words that a dictionary of synonyms gives for an entry as a computational way to represent the meaning of that entry. This is not to say that the meaning of a word is the set of synonym words that a dictionary of synonyms associate with it. Our approach is strictly empirical and practical, but could be helpful for those that analyse meaning from a theoretical point of view in order to test their theories.

This empirical point of view is not free of problems. There is always an excesive dependence on the particular published dictionary that we use, and dictionaries are man-made tools. In consequence, dictionaries may contain mistakes, may not be complete, or may give a slanted view of synonymy. Moreover it is usual to find relations other than that of synonymy between the words appearing in dictionaries of synonyms. This difficulty was perceived by John Lyons when he stated that, strictly speaking, the relation that holds between the words in dictionaries of synonyms is quasi-synonymy more than synonymy (see [10]). We assume these risks on behalf of the practical applications and utility of our resulting tool: Fdsa. Some of these applications and utilities will be shown in Sect. 6 and 7.

3 A computational view of synonymy

In the previous section, we have discussed three main ideas. Firstly, we have seen that it is usual to conceive synonymy as a relation between two expressions with identical or similar meaning. Secondly, we were also able to infer that the controversy of understanding synonymy as a precise question or as an approximate question, i.e. as a question of identity or as a question of similarity, has always been present since the beginnings of the study of this semantic relation. And finally, in order to provide a method to apply synonymy in practice, we have stated that, in this work, synonymy is understood as a gradual relation between words.

In order to calculate the degree of synonymy, we use measures of similarity applied on the sets of synonyms provided by a dictionary of synonyms for each of its entries. In the examples shown in this work, we will use as our measure of similarity *Jaccard's coefficient*, which is defined as follows. Given two sets X and Y, their *similarity* is measured as:

$$sm(X,Y) = \frac{|X \cap Y|}{|X \cup Y|}$$

This similarity measure yields values ranging between 0 (the words are not synonymous at all) and 1 (the words are completely synonymous).

On the other hand, let us consider a word w with M possible meanings m_i, where $1 \leq i \leq M$, and another word w' with M' possible meanings m_j, where $1 \leq j \leq M'$. By $dc(w, m_i)$, we will represent the function that gives us the set of synonyms provided by the dictionary for every entry w in the concrete meaning m_i. Then, the degree of synonymy of w and w' in the meaning m_i of w is calculated as follows [3, 4]:

$$dg(w, m_i, w') = \max_{1 \leq j \leq M'} sm[dc(w, m_i), dc(w', m_j)]$$

Furthermore, by calculating

$$k = \arg \max_{1 \leq j \leq M'} sm[dc(w, m_i), dc(w', m_j)]$$

we obtain in m_k the meaning of w' closest to the meaning m_i of w.

Let us consider this example[4], extracted from the dictionary we will use in this work:

$$w = \texttt{abandonado} \qquad m_i = m_2 \qquad w' = \texttt{sucio}$$

$$dc(w, m_2) = \{\texttt{abandonado}, \texttt{desaseado}, \texttt{desaliñado}, \texttt{sucio}\}$$

Case $m_j = m_1$:

$$\begin{aligned} dc(w', m_1) &= \{\texttt{sucio}, \texttt{impuro}, \texttt{sórdido}\} \\ dc(w, m_2) \cap dc(w', m_1) &= \{\texttt{sucio}\} \\ dc(w, m_2) \cup dc(w', m_1) &= \{\texttt{abandonado}, \texttt{desaseado}, \texttt{desaliñado}, \texttt{sucio}, \\ &\quad \texttt{impuro}, \texttt{sórdido}\} \\ sm[dc(w, m_2), dc(w', m_1)] &= \frac{|dc(w, m_2) \cap dc(w', m_1)|}{|dc(w, m_2) \cup dc(w', m_1)|} = \frac{1}{6} = 0.16666667 \end{aligned}$$

[4] The Spanish words involved in this example (and one of their corresponding translations into English) are: `abandonado` (*slovenly*), `sucio` (*dirty*), `desaseado` (*untidy*), `desaliñado` (*down-at-heel*), `impuro` (*impure*), `sórdido` (*squalid*), `inmundo` (*foul*), `puerco` (*nasty*), `cochino` (*filthy*), `obsceno` (*obscene*) and `deshonesto` (*indecent*).

Case $m_j = m_2$:

$$\begin{aligned} dc(w', m_2) &= \{\texttt{sucio}, \texttt{inmundo}, \texttt{puerco}, \texttt{cochino}, \texttt{desaseado}\} \\ dc(w, m_2) \cap dc(w', m_2) &= \{\texttt{sucio}, \texttt{desaseado}\} \\ dc(w, m_2) \cup dc(w', m_2) &= \{\texttt{abandonado}, \texttt{desaseado}, \texttt{desaliñado}, \texttt{sucio}, \\ &\quad \texttt{inmundo}, \texttt{puerco}, \texttt{cochino}\} \\ sm[dc(w, m_2), dc(w', m_2)] &= \frac{|dc(w, m_2) \cap dc(w', m_2)|}{|dc(w, m_2) \cup dc(w', m_2)|} = \frac{2}{7} = 0.28571429 \end{aligned}$$

Case $m_j = m_3$:

$$\begin{aligned} dc(w', m_3) &= \{\texttt{sucio}, \texttt{obsceno}, \texttt{deshonesto}\} \\ dc(w, m_2) \cap dc(w', m_3) &= \{\texttt{sucio}\} \\ dc(w, m_2) \cup dc(w', m_3) &= \{\texttt{abandonado}, \texttt{desaseado}, \texttt{desaliñado}, \texttt{sucio}, \\ &\quad \texttt{obsceno}, \texttt{deshonesto}\} \\ sm[dc(w, m_2), dc(w', m_3)] &= \frac{|dc(w, m_2) \cap dc(w', m_3)|}{|dc(w, m_2) \cup dc(w', m_3)|} = \frac{1}{6} = 0.16666667 \end{aligned}$$

Finally, we have:

$$dg(w, m_2, w') = \max_{1 \leq j \leq 3} sm[dc(w, m_2), dc(w', m_j)] = 0.28571429$$

$$k = \arg \max_{1 \leq j \leq 3} sm[dc(w, m_2), dc(w', m_j)] = 2$$

That is, the degree of synonymy of the second meaning of the word `abandonado` with respect to `sucio` is 0.28571429 and the meaning of `sucio` that is more similar to `abandonado` is m_2.

The conception of synonymy as a gradual relation implies a distancing from the idea that considers it as a relation of perfect equivalence. This is coherent with the behaviour of synonymy in the printed dictionary, since it is possible to find cases in which the reflexive, symmetrical and transitive properties do not hold:

- The reflexive relation is usually omitted in dictionaries in order to reduce the size of the corresponding implementations, since it is obvious that any word is a synonym of itself in each one of its individual meanings.
- The lack of symmetry can be due to several factors. In certain cases, the relation between two words can not be considered as one of synonymy. This is the case of the words `granito` (*granite*) and `piedra` (*stone*), where the relation is a hyponymy. This phenomenon also occurs with some expressions: for instance, the expression `ser uña y carne` (*to be inseparable* or, in literal translation, *to be nail and flesh*) and the word `uña` (*nail*) appear as synonyms. In other cases, symmetry is not present because a word can have a synonym which is not an entry in the dictionary. One reason for this is that the lemmas of the words are not used when these words are

provided as synonyms. Another possible reason is an omission by the lexicographer who compiled the dictionary, but, in general, all these problems support the claim of Lyons when he talks of quasi-synonymy to define the relation between words appearing in dictionaries of synonyms ([10]).

- Finally, if synonymy has been understood as similarity of meanings, it is reasonable that transitivity does not always hold.

The dictionary used also includes antonyms of each entry. The main problem with antonyms in most of the Spanish published dictionaries of synonyms and antonyms is that the sets of antonyms for an entry are frequently incomplete. For example, the first meaning of the word `abandonado` has a set of associated antonyms formed by the words `diligente` (*diligent*) and `amparado` (*protected*). Neither word appears as a dictionary entry, but only as a synonym of other entries. Most synonyms of `diligente` and `amparado` are antonyms of `abandonado` and must be included in the set of antonyms of this entry. In other words, a synonym of an antonym of `abandonado` is an antonym of this word. The inclusion of new antonyms in the sets of antonyms under this criterion can be performed automatically by FDSA. It is known that synonymy and antonymy are distinct semantic relations that do not work in exactly the same way. It is not the purpose of this work to deal with the computational treatment of antonymy but our proposal for synonymy could be helpful for the aforementioned lack of antonyms in published dictionaries. Once again, it is necessary to state that our approach is fundamentally guided by applied and practical criteria and the results with regard to antonymy were positive.

The use of finite-state automata to implement dictionaries efficiently is a well-established technique [1]. The main reasons for compressing a very large dictionary of words into a finite-state automaton are that its representation of the set of words is compact, and that the process of looking up a word in the dictionary is proportional to the length of the word, and therefore very fast [7]. Of particular interest for natural language processing applications are minimal acyclic finite-state automata, which recognize finite sets of words, and which can be constructed in various ways [15, 5]. The aim of the present work was to build a general architecture to handle a large Spanish dictionary of synonyms [2].

In the following sections, we will describe a general architecture that uses minimal deterministic acyclic finite-state automata in order to implement large dictionaries of synonyms, and how this general architecture has allowed us to modify an initial dictionary with the purpose of letting the relations between the entries and the expressions provided as answers satisfy the reflexive and symmetrical properties, but not the transitive one.

4 General architecture of an electronic dictionary of synonyms

Words in a dictionary of synonyms are manually inserted by linguists. Therefore, our first view of a dictionary is simply a text file, with the following line format:

```
word meaning homograph synonym
```

Words with several meanings, homographs or synonyms use a different line for each possible relation. With no loss of generality, these relations could be alphabetically ordered. Then, in the case of Blecua's dictionary, the point at which the word `concesión` (*concession*) appears could have this aspect:

```
concesión 1 1 gracia      (grace)
concesión 1 1 licencia    (licence)
concesión 1 1 permiso     (permission)
concesión 1 1 privilegio  (privilege)
concesión 2 1 epítrope    (a figure of speech)
```

For a later discussion, we say that the initial version of the dictionary had $M = 27,029$ different words, with $R = 87,762$ possible synonymy relations. This last number is precisely the number of lines in the text file. The first relation of `concesión` appears in line $25,312$, but the word takes the position $6,419$ in the set of the M different words ordered lexicographically.

Of course, this is not an operative version for a dictionary. It is therefore necessary to provide a compiled version to compact this large amount of data, and also to guarantee an efficient access to it with the help of automata. The compiled version is shown in Fig. 1, and its main elements are:

- The `Word_to_Index` function changes a word into its relative position in the set of different words (e.g. `concesión` into $6,419$).
- In a *mapping* array of size $M+1$, this number is changed into the absolute position of the word (e.g. $6,419$ into $25,312$). This new number is used to access the rest of arrays, all of them of size R. The lexicographical ordering guarantees that the relations of a given word are adjacent, but we need to know how many they are. For this, it is enough to subtract the absolute position of the word from the value of the next cell (e.g. $25,317-25,312 = 5$ relations).
- The arrays *m1* and *h1* store numbers which represent the meanings and homographs, respectively, of a given word. The arrays *m2* and *h2* have the same purpose for each of its synonyms.
- The array *w2* is devoted to synonyms and also stores numbers. A synonym is a word that also has to appear in the dictionary. The number obtained by the `Word_to_Index` function for this word is the number stored here, since it is more compact than the synonym itself. The original synonym can be recovered by the `Index_to_Word` function.

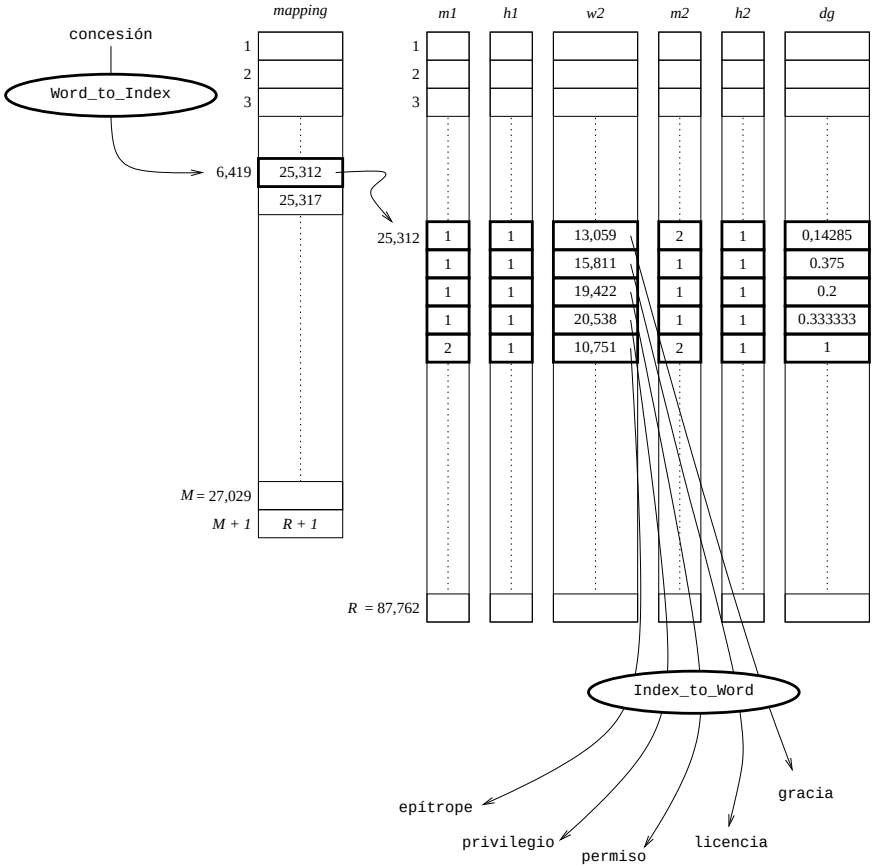

Fig. 1. Compact modeling of an electronic dictionary of synonyms

- The array *dg* directly stores the degrees of every possible synonymy relation. In this case, no reduction is possible.

Note that the arrays *m2*, *h2* and *dg* store data that are not present in the original version of the dictionary. This new information was easily calculated from the rest of arrays with the formulas explained in Sect. 3, once the dictionary had been compiled into this general model and those initial data could be efficiently accessed. The specific transformations performed on the initial dictionary are detailed in Sect. 5.

This is the most compact architecture for storing all the information of the words present in a dictionary, when this information involves specific features of each word, such as the degree of a synonymy relation. Furthermore, this architecture is very flexible: it is easy to incorporate new arrays for other additional data (such as part-of-speech tags), or to remove unused arrays (thus saving the corresponding space). To complete this model, we only need

the implementation of the functions `Word_to_Index` and `Index_to_Word`. Both functions operate over a special type of automata, the numbered minimal acyclic finite-state automata described in [5], allowing us to efficiently perform perfect hashing between numbers and words.

5 Improving the dictionary

The implementation of the dictionary of synonyms [2] was carried out in several steps, some of which required manual processes whereas others could be made automatically. The initial version of the dictionary had 21,098 entries; however it also included 5,931 expressions that appear as synonyms of others but were not entries by themselves (from now, no-entries). This version had 87,762 pairs of synonyms and our first goal was to fill the information corresponding to the *m2*, *h2* and *dg* arrays described in Sect. 4. With respect to *dg* and *m2*, this could be done mechanically by using the formulas of Sect. 3. The automatic detection of homographs *h2* was carried out by including all the homographs of the second word in the calculation of the degree, but only 300 entries of dictionary proved to have homographs. From these initial data we made further modifications related to the properties that the synonymy relation satisfies in the dictionary:

- Symmetry: One of the reasons why symmetry does not hold is the existence of 11,596 pairs involving no-entries. This means that there exist pairs of the form (entry, no-entry) but not the converse pairs. The next improvement was to add as entries all no-entry expressions. In order to do so, we had to associate a set of synonyms to each no-entry. This set was made up of all entries in which the no-entry expression appears as a synonym. In this way, the number of entries and the number of pairs were increased to 27,029 and 99,358 respectively. At this moment, all the expressions involved in the dictionary appear as entries. Nevertheless, the synonymy relation is still non-symmetric. Since we use a measure of similarity, in this case Jaccard's coefficient, two meanings of two different entries will be non zero synonyms (i.e. will be synonyms) if their associated sets of synonyms have some element in common. Following this criterion, if an entry x has synonyms in common with another y given two respective meanings of them, y will have the same synonyms in common with x for those meanings. We have improved the dictionary again by adding to each set of synonyms X the new entries that had meanings whose associated sets of synonyms had elements in common with X. This second step does not further modify the number of entries of the dictionary, but the number of pairs is modified increasing to 621,265. We obtained a symmetric relation of synonymy and we transformed the initial dictionary into a richer one.
- Reflexivity: The final improvement was to incorporate the reflexive pairs in the synonymy relation by adding for each entry of the dictionary the

entry word itself in all the corresponding sets for each meaning of it. This is useful in order to avoid some problems in the calculation of the degrees of synonymy. For instance, when a word x appears as a unique synonym of another y and y as a unique synonym of x for two specific meanings of them, the corresponding sets of these meanings have no elements in common. In this case, the degree of synonymy is 0; therefore x and y will not be considered as synonyms, which is not very intuitive. By adding the reflexive case in the sets of synonyms we avoid this problem. For example, let us consider the Spanish words `carrete` and `bobina`[5]. The only meaning of the word `carrete` had as its set of associated synonyms {`bobina`} and the only meaning of the word `bobina` had as its set of associated synonyms {`carrete`}. If we calculate the degree of synonymy between both words using the similarity measure that we have presented in Sect. 3, we can see that the corresponding sets of synonyms are not similar at all, resulting in a degree of synonymy equal to 0. But if we include the reflexive case in the set of synonyms, we will have the associated set {`carrete`, `bobina`} for both words, which results in a degree of synonymy equal to 1 (i.e. the maximum degree). This second option is more coherent with our intuitions about the synonymy of `carrete` and `bobina`. After this modification the number of pairs increases to 655,583 and the relation is now reflexive and symmetric.

- Transitivity: Since the criterion followed indicates that two entries are synonyms if their corresponding sets of synonyms have elements in common, it is reasonable to think that the synonymy relation is not necessarily a transitive one. This is because, in general, from the fact that a set of synonyms X has elements in common with Y and Y has elements in common with Z it can not be inferred that X has elements in common with Z. Although there exist some dictionaries of synonyms whose synonymy relation is transitive, the dictionary we have used includes a considerable number of examples showing the non-existence of this property.

With regard to the time figures involved in this final configuration of Blecua's dictionary, the time needed to build the automaton (27,029 words, 27,049 states and 49,239 transitions) is 0.63 seconds, in a Pentium Centrino M715 1.5 GHz. under Linux operating system. A further 1.65 seconds are needed to incorporate the information regarding meanings, homographs, synonyms and degrees, thus giving us a total compilation time of 2.28 seconds. Finally, it should be noted that the recognition speed of our automata is around 180,000 words per second. This figure makes it possible to access the information very rapidly, thus proving the suitability of our general architecture for both the process of improvement and the use of this Spanish dictionary of synonyms.

[5] Both words can be translated into English as *bobbin*, *reel* or *spool*.

6 Stand-alone use of FDSA

FDSA (Fuzzy Dictionary of Synonyms and Antonyms) is the generic name of our electronic dictionary of synonyms, although we usually reserve this name for its stand-alone version. As we have seen, we propose a general architecture for this kind of linguistic resource, it being possible to use this model to implement an electronic dictionary of synonyms and antonyms for any language. In this work, we build a dictionary for Spanish from all the information that appears in Blecua's printed dictionary of synonyms [2].

Since FDSA is able to calculate the degree of synonymy between two entries, it presents some advantages with respect to the printed dictionary, amongst which are:

- It provides, using automatic procedures, the meaning of synonyms and antonyms that it gives as answers.
- It provides, by automatic procedures too, the homograph of synonyms and antonyms when these have various homographs.
- It orders the synonyms by the degree of synonymy with respect to the entry.
- It includes in the answer words that do not appear in the dictionary as synonyms but which could be synonyms because they have a non null degree of synonymy.
- It provides more antonyms than the printed dictionary using the criterion that a synonym of an antonym of the entry may be an antonym of the entry.
- It allows the user to reduce the number of answers by the use of thresholds.

Moreover, FDSA offers the user the possibility of implementing all the improvement processes described in this work. The main components of this software are: the electronic dictionaries, the algorithms that calculate the degrees of synonymy and antonymy, and the graphical user interface.

The electronic dictionaries. Initially, they include all the information contained in the printed dictionary. This information is stored in three different electronic dictionaries, `Syn`, `Ant` and `Inf`:

- `Syn` contains all the information about synonyms. To each entry we can associate one or more homographs, to each homograph one or more meanings, and to each meaning a set of synonyms.
- `Ant` contains all the information about antonyms. Its structure is similar to `Syn` but now the sets associated to each meaning are sets of antonyms. In other words, the information is classified in the same way for synonyms and antonyms. Of course, this is not to say that synonymy and antonymy have the same structure.
- `Inf` contains notes on style and usage, such as information about inflexion suffixes, grammatical issues, technical terms, dialectalisms, loanwords,

pragmatic issues, diachronic issues, etc. The structure is similar to `Syn` and `Ant` but now the sets associated to each meaning consist of this additional information.

Modifications can be made to these initial versions of the electronic dictionaries by applying various improvement techniques carried out by the algorithms that calculate the degrees of synonymy and antonymy.

The algorithms that calculate the degrees of synonymy and antonymy. These algorithms have been used to apply the improvements described in this work, thus adding new information to the electronic dictionaries, such as:

- degrees of synonymy and antonymy,
- meanings and homographs of the synonymous and antonymous words,
- no-entries,
- reflexive cases,
- additional synonyms (with respect to the previous version of the dictionary).

Each of these improvements can be incorporated in an independent step, thus providing different versions of the electronic dictionaries. Moreover, the last of the improvements cited above (additional synonyms) can be implemented repeatedly, as a recurrent process. In each iteration, we will also obtain a different version of the electronic dictionaries. However, care must be taken when exercising this option, since it may lead to distortion of our starting point, i.e. the representation of the meaning of a word using the sets of synonyms with which it is associated in the printed dictionary of synonyms. This is particularly critical in the case of polysemic and imprecise words.

The graphical user interface. The graphical user interface of FDSA has three components (see Fig. 2): a main window and two dialog boxes (one of them for synonyms and the other for antonyms). In the dialog for synonyms, a user can introduce a Spanish word and will obtain, among other things, the synonyms that the printed dictionary gives, the words that do not appear in the printed dictionary as synonyms but could be synonyms because they have a non null degree of synonymy, the corresponding degree of synonymy, the verbalization of the degree indicating whether the synonymy is low, medium, or high, and the information about inflection suffixes, grammatical issues, technical terms, etc. Moreover, the user can list the synonyms ordered by degree of synonymy, can fix a threshold that reduces the number of answers, and can select one of various similarity measures.

The dialog for antonyms has the same structure but now the semantic relation between entries and answers is antonymy.

The tool as described above is of interest to the general user, but FDSA also includes a module for advanced users such as computational linguists, lexicographers or natural language processing researchers. This module is named Statistics and improvements (see Fig. 3) and is useful for:

- Obtaining statistical information about the dictionary, for example, number and list of entries, number and list of no-entries, number and list of pairs that hold symmetry, number and list of pairs that do not hold symmetry, etc.
- Improving automatically the dictionaries by adding no-entries or new synonyms using the procedures and criteria described in Sect. 5.
- Dumping all the information stored in the dictionaries to a text file in order to be reused by other tools.

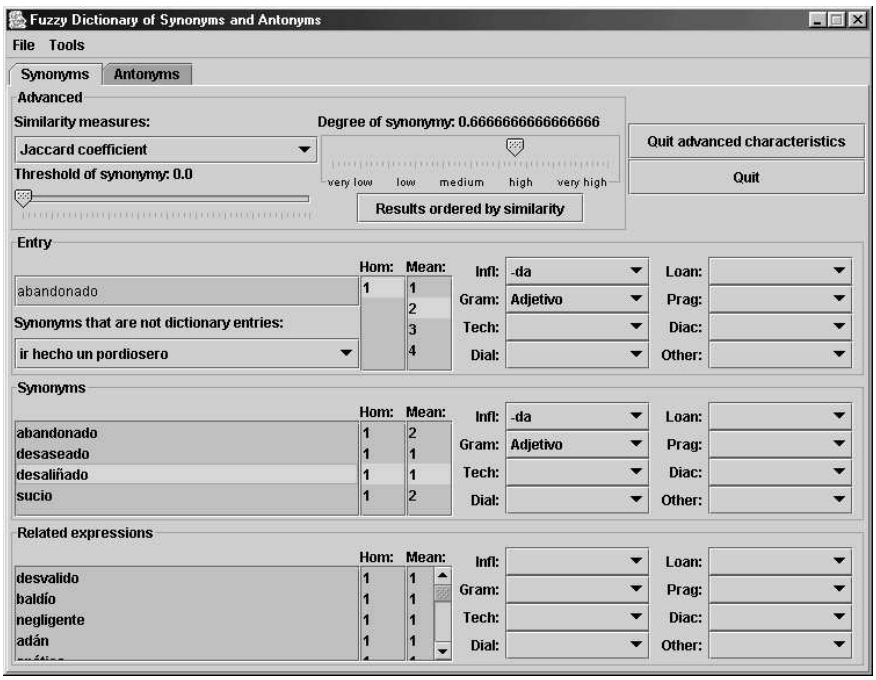

Fig. 2. The graphical user interface of FDSA

7 Conclusions

We have presented a contribution for handling suitably large sets of words in the natural language processing domain. This contribution has been to design a general architecture for dictionaries which is able to store large amounts of data related to the words contained in them. We have shown that it is the most compact representation when we need to deal with very specific information about these words such as degrees of synonymy.

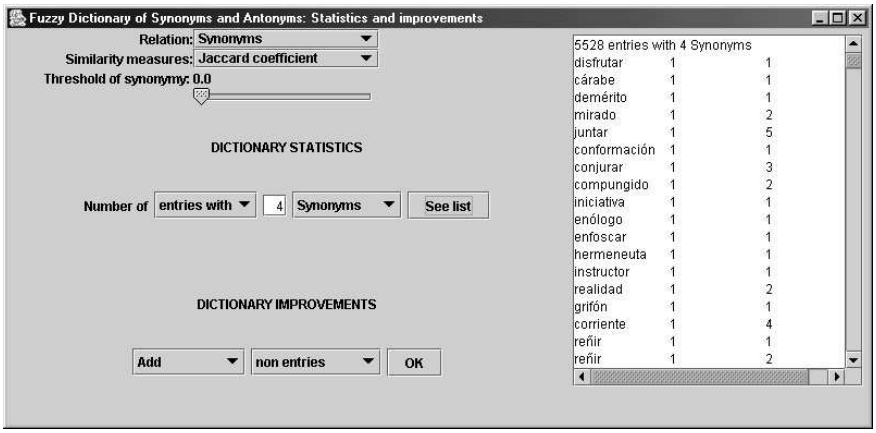

Fig. 3. The graphical user interface of the module Statistics and improvements

We have described how our general model has helped to implement and transform a Spanish dictionary of synonyms into a computational framework able to represent relations of synonymy between words. This framework, characterized by the conception of synonymy as a gradual relation, could be useful in order to improve the efficiency of some natural language processing tasks such as word sense disambiguation or query expansion in information retrieval systems.

With respect to this last task, one of the main problems of using synonymy to increase recall is the loss of precision. The information about degrees of synonymy, not present in other classical approaches such as the Spanish version of EuroWordNet [14] but included in the improved version of the Spanish dictionary of synonyms described in this work, makes it possible to use thresholds to control this loss of precision. If we consider synonymy as an approximate relation between words, we can obtain a greater or smaller number of answers depending on the user specifications of precision. In some cases a high degree of synonymy of the answers with respect to the entries could be necessary, but in other cases we do not need to be so strict with this requirement. Furthermore, Spanish EuroWordNet does not detect the meaning of the words that it gives as an answer.

Therefore, these features of our dictionary lead to conjecture that its use will increase recall without diminishing too much precision and latency in a fuzzy information retrieval system which is still at the experimental stage.

References

1. Aho, A., Sethi, R., Ullman, J. *Compilers: Principles, Techniques and Tools.* Addison-Wesley, Reading, MA, 1985.

2. Blecua-Perdices, J. *Diccionario Avanzado de Sinónimos y Antónimos de la Lengua Española*. Bibliograf, 1997.
3. Fernández-Lanza, S., Sobrino-Cerdeiriña, A. Hacia un Tratamiento Computacional de la Sinonimia. *Revista de la Sociedad Española para el Procesamiento del Lenguaje Natural*, 26:89–95, 2000.
4. Fernández Lanza, S. *Una Contribución al Procesamiento Automático de la Sinonimia Utilizando Prolog*. Ph. D. Thesis, University of Santiago de Compostela, 2001.
5. Graña-Gil, J., Barcala-Rodríguez, F. M., Alonso-Pardo, M. Compilation Methods of Minimal Acyclic Finite-State Automata for Large Dictionaries. *Lecture Notes in Computer Science*, 2494:135–148, 2001. Springer-Verlag.
6. Girard, G. *Synonymes François et leur Différents Significations et le Choix qu'il Faut Faire pour Parler avec Justesse*. Veuve d'Houry, Paris, 9th edition, 1749.
7. Hopcroft, J., Ullman, J. *Introduction to Automata Theory, Languages and Computation*. Addison-Wesley, Reading, MA, 1979.
8. López de Mántaras, R., Trillas, E. Towards a Measure of the Degree of Synonymy. In Sánchez, E. (ed.), *Fuzzy Information, Knowledge Representation and Decision Analysis*, Pergamon Press, 1984.
9. Lee, D.L., Chuang, H., Seamons, K.E. Document Ranking and the Vector-Space Model. *IEEE Software*, 14(2):67–75, 1997.
10. Lyons, J. *Linguistic Semantics. An Introduction.* Cambridge University Press, Cambridge, 1995.
11. Miyamoto, S. Information Retrieval based on Fuzzy Associations. *Fuzzy Sets and Systems*, 38(2):191–205, 1990.
12. Neuwirth, E., Reisinger, L. Dissimilarity and Distance Coefficients in Automation-Supported Thesauri. *Information Systems*, 7(1):47–52, 1982.
13. Salton, G., McGill, M. *Introduction to Modern Information Retrieval.* McGraw-Hill, New York, 1983.
14. Vossen, P. *EuroWordNet. A Multilingual Database with Lexical Semantic Networks.* Kluwer Academic Publishers, Dordrecht, The Netherlands, 1998.
15. Watson, B. *A Taxonomy of Finite Automata Construction Algorithms*. Computing Science Note 93/43, 1993, Eindhoven University of Technology, The Netherlands.