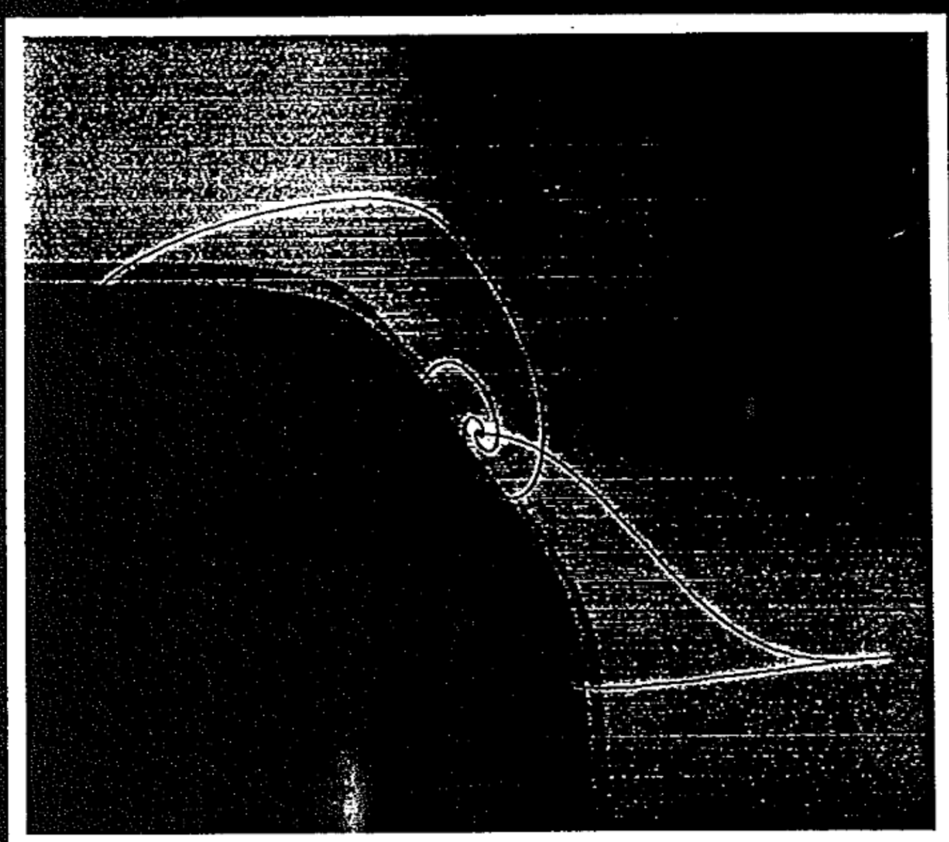
Proceedings

Eighth International Conference

IPMU

Information Processing and Management of Uncertainty in Knowledge-based Systems



Madrid
July 3-7, 2000

Volume II

PROCEEDINGS

EIGHTH INTERNATIONAL CONFERENCE

IPMU

INFORMATION **P**ROCESSING AND
MANAGEMENT OF **U**NCERTAINTY
IN KNOWLEDGE-BASED SYSTEMS

JULY 3–7, 2000
Volume II

CONSEJO SUPERIOR DE INVESTIGACIONES CIENTÍFICAS
MADRID

Cover:

The cover image represents a colour plot of the transfer characteristic of a fuzzy controller. Black line is the trajectory of the controlled plant from an initial point to the equilibrium, followed by a perturbation and a new return to equilibrium.

ISBN Vol 2: 84-95479-04-4 • ISBN Obra Completa: 84-95479-02-8
Depósito legal: M-23536-2000
Printed by: Gráficas 82 S.L.

To order this book, mail to ipmu@mat.upm.es

TABLE OF CONTENTS
Volume II

Aggregation Operators II

Fuzzy Control I

Modelling Linguistic Preferences II

Philosophy of Uncertainty and Vagueness II

Decision Making II

Fundamentals

Fuzzy Control II

Information Retrieval

Aggregation Operators III

Environmental and Biological Models

Information Engineering

Preference Modelling and Decision Making II

Fuzzy Data Bases

Fuzzy Integrals and Applications

Neural Nets

Time

Synonymy from a computational and philosophical perspective.

A. Sobrino Cerdeiriña
Dpt. of Logic and Moral Philosophy
Universidad de Santiago de Compostela. Spain
lflgalex@usc.es

S. Fernández Lanza
Dpt. of Logic and Moral Philosophy
Universidad de Santiago de Compostela. Spain
sflanza@usc.es

Abstract

This paper goes over certain philosophical characteristics of synonymy and certain proposals of computationally dealing with it. In the frame of natural language processing, the present work is a still preliminary attempt at the automation of a dictionary of synonyms.

Keywords: Synonymy, thesaurus, automation of synonymy.

A synonym for 'synonym' has not been found, for them it would appear as if I did not understand the word 'synonym'.

M.G. White

1 Preliminaries

Interest in synonymy, as a relationship of the similarity of meaning, is age old. Aristotle approached this concept 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*". With his description he kindled interest in this subject and influenced the objective of grouping together as synonyms words whose meaning coincided, yet which, nevertheless, showed certain differences.

This tradition was carried on by the Romans. Seleucus wrote a treatise *On the difference in the synonyms* and Ammonius presented a type of dictionary *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) synonymy is a matter of similarity, as is shown by this word's Greek prefix συν –or its Latin equivalent *cog* (*cognominatus*).

Although there have been throughout the ages attempts at defining or characterising synonymy, its systematic study as a lexical relationship is somewhat more recent. The first attempt was made by Gabriel Girard at the start of the Eighteenth Century. In [5] 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*. (Girard, 1749, p. VIII sig)

At present, it is commonly said that two words are synonyms if they have the same or approximately the same meaning. In natural language there are hardly any examples of words that have exactly the same meaning. Some of the rare examples could be *automobile* and *car* or *plane* and *aeroplane*. When this occurs, due to the principle of economy which governs the use of language, one of the synonyms tends to fall into disuse and to disappear, as has occurred with *automobile* and *aeroplane* in favour of *car* and *plane*. Examples of words that have approximately the same meaning abound, as can be seen by consulting any thesaurus.

Thesauri show synonymy as a lexical and approximate characteristic and its study may be relevant in different areas of Artificial Intelligence, such as the development of flexible automatic learning environments (for example in computer systems that detect *delete*, *erase* or *cut* as synonyms), or in the design of expert system communication interfaces in which the richness of vocabulary is valued.

The imprecise characteristics of synonymy in natural language will be studied in this work in terms of the automatisation of a dictionary of synonyms and section 4 is devoted to show this. Prior to that, in section 3, we will present some attempts to automate synonymy and in section 2 we approach an adecuate treatment of meaning for handling synonymy.

2 Synonymy and Meaning

A common definition of synonymy is that two expressions are synonyms if they have the same, or almost the same meaning. Given that the word 'meaning' is a part of the definiens of 'synonymy', it would seem appropriate to analyse it.

The notion of meaning has a long tradition in the Philosophy of Language, and far from there being universal conformity, it has always been a controversial concept. In this work we will analyse some approaches to it, and we will choose the one that, to us, seems most suitable for the computational treatment of synonymy.

The analytical tradition of the Philosophy of Language has been characterised by the study of meaning with the tools of formal logic, and with a view to those concepts that are relevant from a logical point of view, such as analyticity or salva veritate interchangeability. Seen from this perspective, the analysis of the meaning of a proposition requires its formalisation or regimentation, which goes hand in hand with its crispness.

From the point of view that is habitually held in the Philosophy of Language, meaning has fundamentally been dealt with in terms of referential meaning; i.e., as a) a situation or state of affairs, b) mental content, c) a succession of letters or phones. The latter is more characteristics of formal languages than of natural ones.

The first two options, a) and b), pose serious problems. As has been mooted in the Philosophy of Science, not even observational predicates have a single reference in terms of events or matter of fact. Mental content, in turn, varies from individual to individual, and from one moment to another; they are not easily replicated. It may seem that this inconvenience could be avoided by having recourse to their neuronal correlations, so that the referential meaning of the sentence *I have got a headache* may be given in terms of the substitute sentence, *The fibres F in my brain are in a state of stimulation over a threshold u*. However, in this case we would be giving the meaning of the substitutive sentence of the original one, and not the committed sentence. Furthermore, for the substitutive sentence, the correlation of events or states of affairs corresponding to them could also be erroneous.

This poses problems for classical logic when it attempts to clarify or exemplify important notions for its theory, such as defining the salva veritate analyticity or interchangeability in terms of synonymy, with sentences from ordinary language. Thus, it is common to consider that *Michael is slim if and only if Michael is lean* or *2 is a number if and only if 2 is a figure* are examples of analytic sentences. Analyticity follows on from the synonymy of *slim* and *lean* as properties that are attributed to Michael, or from *number* and *figure* as characteristics of 2. If *slim* and *lean* are synonyms, the substitution of one for the other in *Michael is slim if and only if Michael is lean* should leave its meaning (truth value) unaltered –salva veritate interchangeability. From this, the following sentences would result: *Michael is slim if and only if Michael is slim* and *Michael is lean if and only if Michael is lean.* Nevertheless, although these sentences clearly seem to be analytical, the former may be somewhat doubtful. These doubts are less pronounced if the predicates are more precise, as occurs with *number* and *figure*. However, if the aim is to exemplify logical notions with ordinary language, phrases such as *Michael is slim if and only if Michael is lean* will have to be admitted.

With ordinary language, the primary aim must be to attend to its most noteworthy characteristics, such as imprecision –more so than precisificating it (make it precise through regimentation) to adequately illustrate logical notions. But at the same time that we may think it desirable to consider language exactly how it comes about, we believe that formal languages can be of considerable benefit to the study of natural language.

Synonymy frequently appears in ordinary language in terms of similarity of meaning, as is shown by dictionaries of synonyms (thesauri). Dictionaries reflect the linguistic usage of an ideal speaker of a language. Hence, they illustrate a competence-based attitude, not one based on performance, and they are useful tools for testing the use that a linguistic community makes of its vocabulary. Taking this source into account, the meaning that will be investigated in this work is the one related with the use that is made of linguistic expressions in ordinary language and the formalization suggested is related with the automatic management of a dictionary of synonyms

3 Some attempts at the automation of synonymy

In this section we briefly describe three environments which benefit from the mechanisation of synonymy. All of them have to do, directly or indirectly, with the automation of a dictionary or synonyms.

3.1 In Computer-assisted learning environments.

In flexible computer-user communication it is desirable not only that the computer should recognise those words or messages that have predefined ones as being acceptable, rather others that may be equivalent in meaning. This comes about if in a learning aid program, the learner gives the following reply to a question: '*the response is...*', instead of '*the solution is...*', if this were the predefined phrase. In this case, it is a sign of computer robustness that the computer does halt the dialogue, and the interaction with the subject carries on; this happens because it recognises that *response* and *solution* are lexical items that have the same or similar meaning.

With this purpose, E.Foxley and M. Gwei [3] attempted to design a program for the automatic generation of synonyms, using as a reference Roget's Thesaurus, which allows input to concepts or ideas (which will nevertheless have to be verbalised as words) and offers the list of terms that express the concept as output. Its aim is to provide a sort of algorithm for finding the term which best represents an idea or a meaning. It can be summarised into the following steps (examples are included in brackets):

1. To look in the index of the Thesaurus for the word that we consider to be the one that is the most similar to the concept. This word will appear associated to one or more other words (which appear in italics), followed by a number and a grammatical category (e.g. *n. noun*). [The concept that we wish to express belongs to the area of sport, to someone who participates in it, and the word chosen is *sport*. The input item *sport*, has the following entries: *misfit 25 n, athletics 162 n,..., good man 937 n.*].
2. To compare the meaning of the source work with the meaning of the related words that belong to the same grammatical category, and select the most suitable one. [From these entries, *athletics 162 n* us, selected as being the most suitable.
3. To locate the word which best expresses the concept.[It turns out to be *gymnastics*, which was the word that was searched for].
4. If the word or words that are selected in 3 are satisfactory, to stop the search; the paragraph would contain the suitable word and its synonyms. Otherwise, return to step 2. [The list of synonyms would be: *athleticism, gymnastics, feats of strength, callisthenics*].

The automatic generation of synonyms includes a measurement of degree, and the manipulation of prefixes and suffixes is considered as the principal heuristics for the derivation of acceptable words and synonymous sentences. For example, if one wishes to find a lexical item that is related to the concept of *code*, it is quite probable that the class of words that are affixal variants of this word (such as *decode, encode, subcode*) will include the word that is being searched for.

The decision as to which word or paragraph is the most suitable for expressing the meaning or concept is made by using measures of confidence (which measure the certainty that the selected word expresses the meaning) and of relevance (which determine the order of the selected word or paragraph amongst the possible paragraphs that are considered).

The measures of confidence are made to depend on the degree of prefix/suffix manipulation that is used in order to derive the term from the morphological root. If $Error_{af}$ is the percentage of error that is associated to a certain category of affixes, the value of a term that is formed with the affix is $(Value(t)_{af}) = 100 - Error(af)$. If the term occurs in a phrase, the value assigned to the sentence will be half the value that is associated to the term on its own.

Relevance is determined as being relative to a paragraph p and is defined as:

$$Relevance_p = \frac{Score_p}{Max} \times Conf_p$$

where:

- $Conf_p$ is the confidence in the paragraph, and

- $Score_p$ is the sum of the degrees that are obtained in the path that leads the input to the paragraph in which the selected word is located:

$$Score_j = \sum_{i=1}^{N_p} Val_i$$

- *Max* is the greatest high score, relative to an individual paragraph, which is obtained in the search for the appropriate word.

This automation of synonyms should help in the selection of words when we have an idea of what we want to say, but one cannot find the suitable lexical item. The system must be able to diagnose whether the user employs the word incorrectly, and in this case, determine what he/she wished to communicate, and then suggest which terms are close to expressing this information, without losing the root meaning of the word employed by the user. Thus the aim is:

- To facilitate the automatic unsderstanding of the meaning of words.
- To make applications for coversational interfaces, with a computer dictionanry which assists in the search for synonyms in a more efficient manner than could be done with a manual dictionary.

3.2 In data classification.

B. Cases [1] proposes the automation of a dictionary of synonyms by using a self-modifying Q-Diam automaton as a model. The self-modification process leads to the avoidance of the paradoxes that may come about in a manual dictionary if an inverse cross-search is carried out, i.e., from the source word to the synonyms and vice versa. The paradoxes emerge when in a reversible chaining a word and its negation are reached. His aim is to obtain an automatic dictionary in which a word and its negation cannot belong to the same class and, on identifying precision without contradictiousness, where two synonymous words can replace each other in any context without any change in meaning.

To this end she uses Q-analysis, which is a branch of the theory of systems which provides the geometric interpretation of a relation r that is represented in a hypergraph:

$r \subset X \times Y,\ X = \{x_1, x_2, \ldots, x_n\},\ Y = \{y_1, y_2, \ldots, y_n\}$,

Briefly, a hypergraph K is defined as the pair (Y,E), where Y is the set of vertices or descriptors and E = $\{P_i\colon 1 \le i \le n\}$ is the set of edges. . $P_i = P(x_i) = \{y_j\colon x_i r y_j\}$ is the description of the item x_i. The notions of dimension (number of edges), q-neighbourhood (q is the number of edges that two polyhedrons share), and q-connection between polyhedrons (two polyhedrons are q-connected if they share a number of edges that is greater or equal to q) are also considered. The network (wood) that is obtained by ordering the connections in terms of the q-neighbourhood is called the q-analysis of the hypergraph K.

A dictionary of synonyms must consist of lexical entries and a list of words that have the same meaning:

$Sin(A, A_i) \Leftrightarrow A\colon A_1,\ldots,A_i,\ldots,A_m$ *is a lexical entry or* $A = A_i$.

In order to obtain its automation, the Q-analysis methodology is followed. Thus the synonyms of the term A are represented by the polyhedron P(A) = $\langle A, A_1,\ldots,A_m\rangle$. The aim of this being twofold:

- To reduce the number of dictionary entries, making different terms which in a manual dictionary are not classes of equivalence equivalent, since, according to the author, in these dictionaries it can be seen that synonymy is reflexive, but neither symmetrical nor transitive.
- To avoid contradictions that may come about in the reversible chaining. To this end she proposes to carry out a forwards and backwards search (inverse direction). She calls the latter one *look-up*^{R} and defines as follows:

$$\text{look-up}^R(A) = \{B\colon \sin^*(B, A)\}$$

It can then be verified that certain paradoxes come about, e.g., if it is interpreted that *violet* and *yellow* are opposite colours (in fact they are complementary on the chromatic scale), it turns out that in a Spanish dictionary of synonyms *blanco = amarillento, blanco = cárdeno*, but *amarillento = no(cárdeno)*. In order to resolve these cases, a tetra-valued logic is chosen, which formalises the so-called *Diamond* paradoxes. The logical values are ordered in a diamond form I≤T=F≤J, where T and F are Boolean values, I is the initial value and J is the paradoxical value, solutions to the self-negating equation X=not(X). The binary operators '+' y 'x' are defined as *max* and *min*, in order to add or multiply values.

In a reversible chain, the calculation of a group of synonyms to another is carried out by defining a Q-diam automaton:

$M = (K, D, f_q, R)$

where,

- K is a hypergraph and each polyhedron $P(x_i)$ is one cell (an automaton),
- D = $\langle T, F, I, J\rangle$ is the set of states for the cells,

- f_q is the transition function of states $f_q(P, P') \in D$ (q is a parameter for connectivity),
- R is a rule constructor, acting on the 0-neighbourhood of a cell in order to produce a self-referential equation(N_q is the neighbourhood q for the cell P).

$$R(N_0(P(x)) = X = 'd(X, X_1,\ldots,X_n)'.$$

Given a lexical input A_0: $A_1,\ldots,A_m$ with $a_0=P(A_0)$, R is selected so that $R(N_0(a_0))$ is the equation (Σ refers to the diamond + operator):

$$A_0 = \sum_{i=0}^{m} A_i \times f_q(a_i, a_0)$$

If $\dim(a_0 \cap a_i) \geq q$, $f_q(a_i, a_0) = J$; otherwise $f_q(a_i, a_0) = I$. It should be noted that $A_i \times J = A_i$ y $A_i \times I = I$.

The states are interpreted as follows:

-$I \in D$ represents the ground colour

-T and F represent the colours of the dictionary which are reverse chained, and can be taken as signals wich pass from one cell to another.

- J is the colour for the paradox: two input items made up of words with opposite meanings (in the example above, *amarillo* and *cárdeno*), which cross in the same lexical entry (*blanco*).

The aim is to offer a program which organises itself in order to extract a precise and non-contradictory relation of synonymy. This is carried out by a type of genetic algorithm that modifies the structure of the Q-diam automaton as it computes, bearing in mind the relative eccentricity *(ecc(P, P_i))* of one cell with respect to another one and the mean eccentricity *(ecc_S(P))* of one cell with respect to the set of cells that are connected by a q-neighbourhood. The idea is to modify a cell P making *(ecc_S(P))* decrease and to integrate the values so that a word and its negation do not appear in the same class.

From this process, which is not described in detail, the result is groups of lexical entries: entries that are affected by T (resp. F) are similar to those that are affected by the very disturbance in their neighbourhood. Entries that are affected by J change, so that they avoid being disrupted by T and by F at the same time, in order that they should not reach contradictory values. Finally classes of word equivalence are obtained which do not contain terms and their negations. Thus, they are interchangeable in any context.

3.3 In information retrieval on the Internet.

Green [6] presents an automatic hypertext generation system which is based on the *lexical chaining* technique, which justifies the setting up of links between the words of a text that have some sort of similartity in meaning. The usefulness of synonymy is to be found in that it is possible to extend this relation to words that, although not being identical, mean approximately the same, increasing the possibility of establishing links that in another case would have gone unnoticed or remained hidden.

In this work the lexical chains are taken from the WordNet database. WordNet shows three types of relations between words: a) Extra-strong, if the same word is repeated; b) Strong, if the words are in the same WordNet synset (class of synonymous words) or if they are connected by a simple antonym or by the relations *is-a* or *includes*; c) Regular, if there is an admissible path in the WordNet graph between the synsets that contain the words under consideration.

By using these relations it is possible to determine the opportuneness of establishing intra- or inter-document links.

- *Intra-documental links.*

Given that it is usually admitted that the structure of a document is related to the structure of links that are admitted by its vocabulary, is it plausible to put forward that the vocabulary that belongs to the same paragraph will have a certain degree of uniformity. In order to determine how relevant a link is for a paragraph of the document, the fraction of words (punctuation signs are excluded) of the paragraph that are present in this link is found. This fraction is called the paragraph chain density, and consequently, for each paragraph there is a density vector, which deals with the number or links that are allowed for each word.

The similarity of two or more paragraphs in a document is computed, by calculating the proximity chain density vectors that represent them, by using a similarity measurement. The resultant is a symmetrical matrix k×k, where k is the number of paragraphs, and from this matrix it is possible to find the mean and standard deviation of the paragraph similarities that are found. If the two paragraphs are more similar than a given threshold, then links should be established between them. The threshold is described in terms of a value p, which represents the degree of standard deviation from the average similarity for a given document (if a simple threshold is defined, almost all the paragraphs will tend to be linked).

- *Inter-documental links*

Here the aim is to have hypertext links that respond to the premise that two documents dealing with the same theme tend to use similar words (though not necessarily the same ones). By using the lexical chaining that is permitted by WordNet the links can be extended, as has already been pointed out, to semantic relations that are more appropriate than the mere repetition of words.

The process for establishing inter-documental links is made up in the following manner: Each document in the database is represented by two vectors. Each element of the vector includes information on a WordNet synset.

- First vector (called *member*): This contains a weight based on the number of occurrences of a given synset in the words that make up the sentences of a document.
- Second vector (called *linked synset vectors*): This contains a weight based on the number of occurrences of a given synset – which is not present in the first vector- which is related (e.g. by antonymy) in the words that make up the sentences of a document. (The weighing of the synset is based not only on its frequency in the document, but also on the number of times that it appears in WordNet).

Given the vectors, the relation between the two documents D1 and D2 can be computed, by calculating three similarities:

1. The similarity of the member vectors of D1 and of D2.
2. The similarity of the member vector of D1 and the linked vector of D2.
3. The similarity of the linked vector of D1 and the member vector of D2.

The first similarity is the most important, since it captures the repetition of terms and the relation between documents that is given by synonymy. The second and third similarities perceive more distant relations between WordNet synsets. The sum of these three similarities indicates between which documents it is possible to establish links.

These three attempts at the automation of synonymy are commendable, but they are not without problems. The first deals with the automatic generation of a dictionary of synonyms based fundamentally on the manipulation of affixal categories, however the extension of this process or others of a similar nature to other types of vocabulary would seem to be difficult. The second attempt is a very imaginative, but artificial approach, since it attempts to eliminate the meaning similarity of words that are synonyms, in this way drawing up a dictionary of perfect synonyms. The third and final case that is described uses a type of dictionary of synonyms (WordNet) in order to automate Internet searches, however there is no mention as to how this dictionary may be mechanised.

In section 3 we propose a still preliminary attempt at the mechanisation of a dictionary of synonyms using logic programming techniques [3].

4 A Natural Language Processing-Based Approach to Synonymy

In the natural language processing (NLP) tradition [3], we will describe a Prolog program that manages the degree of synonymy between the words of a dictionary of synonyms. The examples are taken from [8]. Measurement is realised meaning by meaning. Thus, given two words A and B, the program sets the meaning of A, verifying that B belongs to the list of synonyms of A in that meaning, it calculates the degree of synonymy that A possesses with all the meanings of B and selects the meaning of B whose degree of synonymy is the greatest.

A dictionary of synonyms can be considered as a database of words (entries) and lists of words (synonyms) grouped together by meanings. We will use the following predicate:

syn_dic(Word,List_of_synonyms, Meaning)

to represent this information. The first argument is the dictionary entry, the second is the list of its synonyms and the third, the meanings corresponding to the group of synonyms. Let us consider an example that includes various input items from [8]:

```
syn_dic(fruit,[fruit,crop,harvest,produce,p
roduct,yield],1).
syn_dic(fruit,[fruit,advantage,benefit,cons
equence,effect,outcome,profit,result,return
,reward],2).
syn_dic(crop,[crop,fruit,gathering,harvest,
produce,reaping,'season's_growth',vintage,y
ield],1).
syn_dic(crop,[crop,clip,curtail,cut,lop,mow
,pare,prune,reduce,shear,shorten,snip,top,t
rim],2).
...
```

The general procedure for calculating the degree of synonymy between two words is the following. Given two words A and B:

1. Take the list of synonyms of A in the first meaning, which we will call ListA.

2. Check that B belongs to ListA.

2.1. If it does not belong to ListA, carry out step 1 with the following meaning of A. If all the meanings of A have been checked and the result is still negative, then A and B are not synonyms.

2.2. If it belongs to ListA, go to 3.

3. Take the list of synonyms of B in the first meaning (called ListB) and calculate the degree using one of the usual similarity measures: jaccard coeficient, dice coeficient, cosine coeficient, mutual similarity coeficient or overlap coefficient.

4. Repeat step 3 with all the meanings of B.

5. Calculate the maximum of the degrees obtained in 4 and its corresponding meaning. The result is the degree of synonymy between A and B.

The predicate *synonym* calculates the degree of synonymy between two words:

synonym(A, MA, B, MB, DS, TH, SM)

that is, the word A in the meaning MA is a synonym of the word B in the meaning MB with a degree DS for the threshold TH and for the similarity measure SM. It runs in this way:

1. Any word is a grade 1 synonym of itself if it belongs to the dictionary and its meaning does not vary, for any threshold and similarity measure.

2. Two different words are DS_MAX degree synonyms, given a threshold and a similarity measure, if:

- The meaning of the first one being set, the second one belongs to the list of the former's synonyms.
- List1 is a list of all the degrees that can be obtained by varying all the meanings of the second word for that threshold and similarity measure.
- List2 is the list of all the meanings that correspond respectively to the degrees calculated in the previous point.
- DS_MAX is the maximum degree of List1 and MB is the meaning that corresponds to this degree.

The code (to the Jaccard Coefficient) is as follows:

```
synonym(A,X,A,Y,1.0,_,jaccard_coefficient):-
    syn_dic(A,Z,X),
    X=Y.
synonym(A,MeanA,B,MeanB,DS_MAX,TH,jaccard_c
oefficient):-
    comparable(A,B,MeanA),
    findall(GRS,syn_meaning(A,MeanA,B,Mean,
    GRS,TH,jaccard_coefficient),List1),
    findall(Mean,syn_meaning(A,MeanA,B,Mean
    ,GRS,TH,jaccard_coefficient),List2),
    max(List2,MeanB,List1,DS_MAX).
```

The predicate *comparable* enables us to verify whether a word is in the list of synonyms of another one, once the specific meaning of the latter has been set. Thus:

comparable(A, B, MA)

can be read as: word A is comparable with another B in the particular meaning MA of A. Two words are comparable, one of the meanings of the former being set, if:

- X is the list of synonyms of the former for the meaning MA.
- the second is one of the elements of X.

```
comparable(A,B,MA):-
    syn_dic(A,X,MA),
    member(B,X).
```

`member` as is habitual in Prolog.

The maximum of the degrees with the corresponding meaning is calculated by means of the predicate:

max(List_of_meanings,Meaning,List_of_degrees, Greatest_degree)

where the first argument is a list of meanings, the second, the meaning corresponding to the greatest degree, the third is a list of degrees, and the fourth the greatest of these degrees. The predicate runs in this way:

1. The maximum of a list with one single element is this element, both for meanings and degrees.

2. If the lists have more that one element, check if the first element in the list of the degrees is greater than or equal to the second. In the affirmative case, eliminate the second of these in both lists.

3. If the first is not greater than or equal to the second, eliminate the first one in both lists.

4. Repeat this process until only one element remains in both lists.

```
max([B],B,[A],A).
max([D,E|F],Y,[A,B|C],X):-
    A >= B,
    max([D|F],Y,[A|C],X),
    !.
max([D,E|F],Y,[A,B|C],X):-
    max([E|F],Y,[B|C],X).
```

The predicate *syn_meaning* calculates the degree of synonymy for the meanings of two specific words:

syn_meaning(A, MA, B, MB, DS, TH, SM)

(only for the *Jaccard coefficient* is explained in detail). A word A in the meaning MA is a synonym of B in the meaning MB to a degree DS for the threshold TH if:

• It is not the case that A coincides with B (this prevents the compiler from calculating the degree of synonymy for two equal words within the same meaning, as the result would be trivially 1).
• X is the list of synonyms of A for the meaning MA.
• Y is the list of synonyms of B for the meaning MB.
• U is the union of X and Y.
• I is the intersection of X and Y.
• NU is the cardinality of the union U.
• NI is the cardinality of the intersection I.
• The degree of synonymy DS is NI/NU.
• DS is greater than or equal to the threshold TH.

```
syn_meaning(A,MA,B,MB,DS,TH,jaccard_coeffic
ient):-
        not(A=B),
        syn_dic(A,X,MA),
        syn_dic(B,Y,MB),
        union(X,Y,U),
        inter(X,Y,I),
        card(U,NU),
        card(I,NI),
        DS is NI/NU,
        DS >= TH.
```

(`union`, `intersection` and `cardinal` as are habitual in Prolog).

It is possible now ask what the degree of synonymy is between two words belonging to the dictionary, such as *fruit* and *produce*, with the threshold 0 an for the Jaccard coefficient. The goal is:

```
synonym(fruit,Meaning_fruit,produce,Meaning
_produce,Degree,0,jaccard_coeficient).
```

And the program answers:

```
Meaning_fruit = 1
Meaning_produce = 7
Degree = 0.625
```

5 Final remarks

This paper is an initial attempt at automating a dictionary of synonyms using Prolog. The study of the nature of the measurements of similarity and their adaptation to the treatment of dictionary synonymy, as well as the justification of the setting of the threshold, are important themes that will need to be tackled in the future.

The automation of a dictionary of synonyms may be relevant in the design of expert system communication interfaces in which great importance is attached to the qualification and flexibility that is offered by lexical synonymy.

Acknowledgements
This work has been supported by the Xunta de Galicia project PGIDT99PXI10502B, Galicia, Spain.

References

[1] B. Cases (1996). 'From Synonymy to self-modifyng automata: Q-Diam Language', in Dassow, J. et al. (eds.), *Developments in Language Theory II. At the Crossroads of Mathematics, Computer Science and Biology*. World Scientific, Singapore. pp. 454-459.
[2] R. Carnap (1955). 'Meaning and Synonymy in Natural Languages', *Philosophical Studies*, 7, 33-47.
[3] M. A. Covington (1994) *Natural Language Processing for Prolog Programmers*, Prentice-Hall.
[4] E. Foxley &, G. M., Gwei (1989). 'Synonymy and Contextual Disambiguation of Words', *International Journal of Lexicography*, Vol. 2, n° 2, 111-131.
[5] G. Girard (1749). *Synonymes françois et leur différents significations et le choix qu'il faut faire pour parler avec justesse*. Paris, Veuve d'Houry. 9 ed.
[6] S. J. Green (1998). 'Automated link generation: Can we do better than term repetition?, *Computer Networks and ISDN Systems* 30 (1998), 75-84.
[7] B. Mates (1952). 'Synonymyty', in L. Linsky (ed.), *Semantic and the Philosophy of Language*. The University of Illinois Press at Urbana
[8] W. T. McLeod, (ed.) (1989). *The New Collins Dictionary and Thesaurus in One Volume*, Collins.
[9] K. Spark Jones (1986). *Synonymy and Semantic Classification*, Edinburgh U. P.