

Using Proximity Operators for Document Representation in Web Search

Jesús Serrano -Guerrero, José A. Olivas, Javier de la Mata

Dep. Of Computer Science, University of Castilla La Mancha,
Paseo de la Universidad 4, 13071 Ciudad Real, SPAIN,
{jesus.serrano, JoseAngel.Olivas}@uclm.es, jdla@jccm.es

Abstract

It is presented a different way for representing documents in information retrieval systems. The relation degree between the index term pairs which describe the document is proposed as the main factor to represent a document. This document representation approach is based on two kinds of different relations between index terms: the physical relation and the semantic relation.

Keywords: proximity operators, search engine, semantic relations, document representation.

1. Introduction

Classic models (Boolean, probabilistic and vector space) in information retrieval are based on the document representation by means of a set of index terms. Usually, to calculate the similarity degree between a document and a query, the main factor to keep in mind is the frequency of the index terms of the document. This fact causes the lack of conceptual relevance of the query results. The user query is also limited because the user can only use terms to express it, the query is a “lexical focused” query and not a “conceptually focused” query.

The classic systems only use the terms appeared on the documents as index terms, but there are some different works that use ontologies in the indexing process. For example, the semantic indexing technique [6] proposed by Mihalcea uses WordNet [9] to disambiguate the polisemic words. This method improves the results introducing new related terms in the document representation and in the query expansion stage. Other strategy is to use electronic dictionaries to index the documents [3].

1.1 Lexical queries vs. conceptual queries

To efficiently use search engines, the user has to adapt his queries to the different search engines, and not the search engines have to be adapted to the user necessities. The only way to express specialized queries, is that the user knows the internal mechanism of the search engine and its additional operators such as AND, OR, NEAR, FAR, etc. This is one of the main reasons of the bad results for inexpert users. The user spends too much time to learn how to use the search engine appropriately.

Most of the user queries are “lexical focused” queries, without other semantic intentions. For example, the query:

Fuzzy Concept Set

could be conceptually ambiguous. Search engines do not “detect” and “understand” the correct meaning of the query.

Perhaps the user is asking for a document related to:

fuzzy set and concepts

or

concepts about fuzzy sets

or

The conceptual fuzzy sets theory by T. Takagi

So, search engines are not able to understand the meaning of the query and then it is possible that the results would not be enough relevant for the user.

The user of the system here proposed must express his queries in a complete way, forgetting the classic management of search engines, and allowing the system

to know the semantic properties of his query.

In the following section, the conceptual relations used are described. In the third section, the focus is the description of the proposed document representation using conceptual relations. The fourth section talks about the necessity of the implicit knowledge acquisition using electronic dictionaries. Finally, some conclusions and a proposal of studies to be developed in the future are presented.

2. Conceptual relations

There are a lot of linguistic relations between terms: holonymy, hypernymy, hyponymy, meronymy, etc., but working with all of them in an automatic way could be too complex and hard, so the approximation here proposed uses only two kinds of relations: the physical and the semantic ones.

2.1 Proximity Operators

There are information retrieval systems that add additional Boolean operators to improve the precision of the obtained results, but these operators are only used for expert users, so, in the web community there are a lot of users whose never use this kind of tools. For this reason some IRSs have to implement these operators. There are basic operators implemented directly in famous IRSs, for example, the operator AND in Google. One of the most important operators is the one related to the proximity between terms. There are several approaches to describe this operator, but usually only expert users know about its existence, so this operator is not very used in most of the usual Web Search Engines.

There are a lot of studies about the use of key-words in IR's, and it is well known that usually only 2 or 3 key-words are used for each query. These words are used to model a concept:

Conceptual fuzzy sets
Adolph Hitler
Information retrieval systems

Most of the queries represent a concept, where the more important factor is the proximity between terms, so to improve the precision of the IRS is necessary to take into account the proximity factor. Most of the users don't know the existence of proximity operators or do not use them, but there are studies about the positive influence of proximity and adjacency operators in the precision of the IRS [4].

Some typical habits when using search engines limit the effectiveness of the searching process. The origin of this problem is the search based on key-words. There is

also a relevant lack of experience in the users when using search engines. The low average number of terms used in a query (the average is 2.35) and the low feedback of the query with new terms (only 2.02 queries per session), as well as the low number of documents consulted (the average screens per query is 1.39) often make the problem of finding relevant documents become complicated [8]. The problems appear when using a small number of terms in the query, since the same word can have different meanings. Another disadvantage is the vocabulary problem [2]. Previous research has shown that people tend to use different vocabulary to describe the same concept.

The way of making searches in the Web changes depending on the search engine used, in order to obtain better results. The user must be conscious of the characteristics of the tool used to design the opportune plans and strategies for the search. Common ignorance in the functioning of search engines leads users into making poor decisions regarding search techniques. Most inexperienced users only use the basic characteristics of the search engines, avoiding advanced searches and the use of search operators. In most cases, users are restricted to make new searches by using other semantically related terms.

For this reason (to improve the precision of the IRS) the implementation of an implicit proximity operator to represent the weight vector (which is the internal representation of a document in our IRS) is proposed.

2.2 Semantic relations

The proposed semantic relations between terms are based on two main ideas:

1. In a document, the relation between two words located in the same sentence should be different (bigger) to that between words from two different paragraphs. This idea is modeled by means of a value called "the distance coefficient".
2. And if an idea is repeated in several paragraphs, is more important than an idea that appears only in a paragraph. This idea is modeled by means of a value called "the repetitive coefficient".

2.1.1. The distance coefficient (DC)

It was described the necessity to use implicit proximity operators. The system here proposed, tries to apply this idea by means of the "distance coefficient":

$$DC (t_i, t_j) = \frac{\sum_{occur (t_i, t_j)} \left(\frac{1}{a^{sentence * (paragraph + 1)}} \right)}{num (occur (t_j, t_i))}$$

Explanation of the formula:

* $num(ocurr(t_j, t_i))$ represents the number of pairs $t_i - t_j$ that appear in a document. For example (figure 1 on the left), its value is 3 because there are three different pairs of terms (adventure – sports). It is a term to normalize the Distance Coefficient. The value of Distance Coefficient is in the range [0, 1].

The proximity operator is calculated by means of:

$$a^{sentence * (paragraph + 1)}$$

where:

- ❖ a is a constant (experimental) value, from now on $a = 1,1$.
- ❖ $sentence$ is a variable which value is 0 if t_i and t_j are in the same sentence, 1 if t_i is in a sentence and t_j are in the next sentence, 2 if t_i is in a sentence and t_j is the second next sentence, and so on.
- ❖ $paragraph$ is a variable which value is calculated such as the sentence value above described, 0 if t_i and t_j are in the same paragraph and so on.

For each pair of terms $t_i - t_j$ the proximity between the terms is calculated, so it is being taken into account if the pair $t_i - t_j$ is a whole concept.

$$\sum_{occur(t_i, t_j)} \left(\frac{1}{a^{sentence * (paragraph + 1)}} \right)$$

The repetitive coefficient (RC)

If an idea is repeated in several paragraphs, the semantic relation is stronger because this idea is part of the “topics” of the document. This concept is measured:

$$RC(t_i, t_j) = \frac{num(paragraph - cooccur(t_i, t_j))}{num(totalparagraph)}$$

where

* $num(paragraph-cooccur(t_i, t_j))$: It is the number of paragraphs where exist a co-occurrence of the terms t_i, t_j .

* $num(total paragraph)$: It is the number of paragraphs in the document. It is used to normalize the equation in the range [0, 1].

So, if in a document, the same idea is being repeated in several paragraphs the $RC(t_i, t_j)$ value is maximum. And if a document has several paragraphs talking about

different ideas, the value of the RC is lower. These last ideas are described by the following figure (Fig. 1)

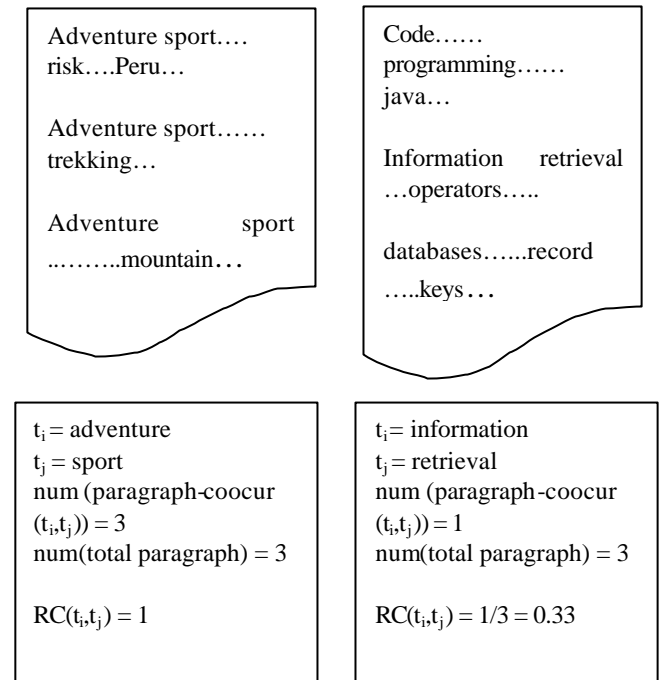


Fig. 1. Calculation of Repetitive Coefficient in two different documents.

The conceptual relations (CR) between terms are based on the “Distance Coefficient” (DC) and the “Repetitive Coefficient” (RC):

$$CR = DC \text{ AND } RC$$

In a simple way, the AND operator is the product operator. Thus, the conceptual relation can be calculated as:

$$CR = \frac{\sum_{occur(t_i, t_j)} \left(\frac{1}{a^{sentence * (paragraph + 1)}} \right)}{num(occur(t_j, t_i))} * \frac{num(paragraph-cooccur(t_i, t_j))}{num(totalparagraph)}$$

Physical Relation

There are a lot of terms very related to the document terms which do not appear in an explicit sense, but they exist and it is necessary to add them to the document representation. These terms are modeled by means of relations such as: inheritance, aggregation, synonymy, etc. There are several sources to obtain this kind of relations, for example, using a thesaurus or by means of the syntactic analysis of the indexed documents. The first one is a relatively simple process with some problems such as the ambiguity, and the second one is very hard

process which needs the use of complex libraries to manage text such as GATE [1].

So, the identification of verbal forms such as: “to be”, “belong to”, “to have”, etc. allows establishing physical relations:

The car has wheels (1)

In the same way, using thesaurus it can be added synonyms, hyponyms, etc. to the document representation. So, there are no differences between the last example and the following sentence using a synonym:

The auto has wheels

Or it can be replaced the term “car” by its hypernym “automobile”.

In this case, the relation weight value is very high because the relation is very intuitive.

There are a lot of thesaurus and electronic dictionaries to use, the most famous is WordNet but there are some of them that given a term specify related terms with different fuzzy degrees. Some examples are OMCSNet¹, the Spanish Synonymy dictionary design by Fernandez [7], or the measures proposed by Garces[5].

The system here proposed uses the second approach in a simple way; it uses only WordNet to retrieve synonymy and hyperonymy relations. The value of the physical relation (PR) between two terms using WordNet is 1. So considering the sentence above (1), the physical distance between “car” and its hypernym “auto” is:

$$PR(car, auto) = 1$$

This coefficient (PR) will be later used for document representation in the phase called “context expansion”.

3. Document Representation

The semantic relation, described above, represents the membership degree of each pair of words ($t_i - t_j$) to a document (D), so it can be written like:

$$CR = \mathbf{m}_{CR} (t_i - t_j, D)$$

The document representation is a binary fuzzy relation R defined on the Cartesian product of B x B of the set of document terms such as:

$$R = \int_{b \times b} \mathbf{m}_R (b, b) / (b, b)$$

where \mathbf{m}_R is the membership function of R given as

$$\mathbf{m}_R : B \times B \rightarrow [0, 1]$$

and B is the set of document terms.

$$B = \{\text{set of document terms}\}$$

Graphically the document is a bidimensional matrix where each cell represents the CR between pairs of terms (Fig. 2):

	Screen	TFT	Pentium	Order
Screen	0	0.7	0.5	0.2
TFT	0.7	0	0.5	0.1
Pentium	0.5	0.5	0	0.7
Order	0.2	0.1	0.7	0

Fig.2. Document A

All the values of the main diagonal are zero, because is not possible to evaluate a term t with itself, i. e., it does not exist a real conceptual distance in the pair t-t.

In the last figure, it can be seen a document representation where

$$Screen \xrightarrow{0.7} TFT \xrightarrow{0.5} Pentium \xrightarrow{0.7} Order \xrightarrow{0.2} Screen$$

It can be observed that the relation between Screen and TFT is very strong because it may exist a paragraph talking about screens. There may be also a paragraph talking about *order* and *processors*, due to the strong relation between *Pentium* and *Order*. The relation between *order* and *screen* is lower (0.2).

4. Context Expansion

For each word of the document are added hipernyms and synonymies. So the document is modelled by means of:

$$A = \{\text{document term set}\}$$

$$B = \{\text{set of synonymies and hypernyms of A}\}$$

The relation R will be defined on the Cartesian product A x B of the set A and B such as:

$$R = \int_{a \times b} \mathbf{m}_R (a, b) / (a, b)$$

where the membership function of R

$$\mathbf{m}_R = A \times B \rightarrow [0, 1]$$

is defined by the electronic dictionary.

¹ <http://web.media.mit.edu/~hugo/conceptnet/>

Considering the document of figure 2, it is possible to be retrieved, using an electronic dictionary, the following relation (Fig. 3):

$$A = \{Pentium, order, tft, screen\}$$

B = {monitor (synonymy of monitor), computer (hypernym of processor y meronym of screen), instruction (synonymy of order)}

	Monitor	Order	Computer
Screen	1	0	0.6
TFT	0	0	0
Processor	0	0	1
Instruction	0	0.8	0

Fig.3. Relation A x B

The new terms allow the system to complete the document representation redefining the relation R and S in the Cartesian product A' x A', where

$$A' = \{processor, instruction, TFT, monitor, screen, computer, order\}$$

And the fuzzy value of the new relations is zero. Thus, the new representation of the relation R and S is:

R =

	Screen	TFT	Processor	Instruction	Monitor	Order	Computer
Screen	0	0.7	0.5	0.2	0	0	0
TFT	0.7	0	0.5	0.1	0	0	0
Processor	0.5	0.5	0	0.7	0	0	0
Instruction	0.2	0.1	0.7	0	0	0	0
Monitor	0	0	0	0	0	0	0
Order	0	0	0	0	0	0	0
Computer	0	0	0	0	0	0	0

S =

	Screen	TFT	Processor	Instruction	Monitor	Order	Computer
Screen	0	0	0	0	1	0	0.6
TFT	0	0	0	0	0	0	0
Processor	0	0	0	0	0	0	1
Instruction	0	0	0	0	0	0.8	0
Monitor	1	0	0	0	0	0	0
Order	0	0	0	0.8	0	0	0
Computer	0.6	0	1	0	0	0	0

and the union (max operator) between both relation is:

$$R \cup S =$$

	Screen	TFT	Processor	Instruction	Monitor	Order	Computer
Screen	0	0.7	0.5	0.2	1	0	0.6
TFT	0.7	0	0.5	0.1	0	0	0
Processor	0.5	0.5	0	0.7	0	0	1
Instruction	0.2	0.1	0.7	0	0	0.8	0
Monitor	1	0	0	0	0	0	0
Order	0	0	0	0.8	0	0	0
Computer	0.6	0	1	0	0	0	0

which represents the original documents with explicit and implicit terms. Now, the system is ready to answer a query in which are used the terms *computer* and *screen*. However, only the direct relation between terms has been added, for example, the relation *monitor* and its synonymy is add:

$$m_{R \circ S}(screen, monitor) = 1$$

but the indirect relations, such as “screen” and “TFT” do not exist:

$$m_{R \circ S}(TFT, monitor) = 0$$

when “monitor” and “screen” are synonym and besides :

$$m_{R \circ S}(TFT, monitor) = 0.7$$

its membership degree between “TFT” and “monitor” is greater than zero.

So, it is necessary to re-calculate the relations between terms using the composition operation (max-min):

$$(R \circ S) \circ (R \circ S) =$$

	Screen	TFT	Processor	Instruction	Monitor	Order	Computer
Screen	0	0.5	0.5	0.5	0.7	0.2	0.5
TFT	0.7	0.7	0.5	0.2	0.7	0.1	0.6
Processor	0.5	0.5	1	0.2	0.5	0.7	0.5
Instruction	0.2	0.5	0.2	0.8	0.2	0	0.7
Monitor	1	0.7	0.5	0.2	1	0	0.6
Order	0	0	0.7	0	0	0.8	0
Computer	0.6	0.6	0.5	0	0.7	0	1

Now, the system has all the relations between its terms well calculated. The new matrix is not symmetric, there are several ways to calculate the relation between terms :

$$Screen \xrightarrow{1} Monitor \xrightarrow{0.7} TFT$$

$$\mu_{R \circ S}(Monitor, TFT) = 0,7$$

$$Screen \xrightarrow{0.7} Monitor \xrightarrow{0.5} TFT$$

$$\mu_{R \circ S}(Monitor, TFT) = 0,5$$

Fig.4. Different membership degrees between “Screen” and “TFT”

Logically, the most interesting value is the maximum, because it shows the greater membership degree between the terms. Now the system is able to answer the query

Monitor TFT

Using the relation

$$\mu_{R \circ S}(screen, TFT) = 0,7$$

4. Conclusions and future work

The work here proposed is only a mathematical approach for document representation. The system proposes an internal implementation of the proximity operators to improve the semantic capabilities of the information retrieval systems. Apart from this system, It will be necessary the implementation of a suitable mathematical method for representing user queries.

The best results obtained until now were in documents with a great quantity of information, but in Internet there are important web pages in which the text is not the most important factor, so, the future works has to try to solve this problem

The fuzzy t-conorms and t-norms used for modeling the process are very basic, so it is necessary to test different measures to improve the system results.

The measures used for calculating the different coefficients (repetitive, distance, etc.) can be improved with new parameters and variables.

Acknowledgments

Partially supported by SCAIWEB PAC06-0059 project, JCCM, Spain.

References

[1] GATE (General Architecture for Text Engineering) <http://www.gate.ac.uk/>

- [2] G. W. Furnas; T. K. Landauer; L. M. Gomez; S. T. Dumais. (1987). The Vocabulary Problem in Human-System Communication. *Communications of the ACM*, volume 30, number. 11, pages 964-97.
- [3] J. Gonzalo, F. Verdejo, I. Chugur, J. Cigarrán, “Indexing with WordNet synsets can improve text retrieval,” Proceedings of the COLING/ACL’98 Workshop on Usage of WordNet for NLP, Montreal, 1998, pp. 38-44.
- [4] MC. McJunkin. Precision and recall in title keyword searches. *Information Technology and Libraries*. 14(3): 161-171. 1995.
- [5] P. Garcés; J. A. Olivas; F. P. Romero,: “Concept-matching IR systems versus Word-matching IR systems: Considering fuzzy interrelations for indexing web pages”. *Journal of the American Society for Information Science and Technology JASIST*, 57 (4): 564-576, 2006.
- [6] R. Mihalcea, D. Moldovan, “Semantic Indexing using WordNet Senses,” Proceedings of ACL Workshop on IR & NLP, Honk Kong, 2000.
- [7] S. Fernandez, “A contribution to the automatic processing of the synonymy using Prolog,” PhD Thesis, University of Santiago de Compostela, Spain, 2001.
- [8] C. Silverstein; M. Henzinger ; H. Marais and M. Moricz. Analysis of a Very Large AltaVista Query Log. *Technical Note 1998-014, Compaq Systems Research Center*, 1998.
- [9] WORDNET. A lexical database for the English language. <http://www.cogsci.princeton.edu/~wn/>