A Method for Measuring Semantic Similarity of Documents

Andreia Dal Ponte Novelli
Dept. of Computer Science
Aeronautic Technological Institute
Dept. of Informatics
Federal Institute of Education, Science and Technology of Sao Paulo
Sao Paulo, Brazil

Jose Maria Parente de Oliveira
Dept. of Computer Science
Aeronautic Technological Institute
Sao Paulo, Brazil

ABSTRACT
With the documents increasing amount available in local or Web repositories, the comparison methods have to analyze large documents sets with different types and terminologies to obtain a response with minimum documents and with as much useful content to the user. For large documents sets where each document can contain many pages, it is impossible to compute the similarity using the entire document, to require creating solutions to analyze a few meaningful terms, in summary form. This article presents TextSSimily, a method that compares documents semantically considering only short text for comparison (text summary), using semantics to improve the set of responses and summaries to improve time to obtain results for large sets of documents.

General Terms
Semantic Retrieval, Documents Comparison.

Keywords
Semantic Similarity, Comparison by Similarity, Short Text Comparison.

1. INTRODUCTION
The amount of information available to society grows exponentially and makes it necessary the development of ways of improving the information retrieval from sets of documents, getting smaller sets of responses with greater amount of relevant information. One way to reach this purpose is by the employment of search engines that incorporate semantics in comparison and make use of similarity.

For a set of documents, a search may have a document containing many pages as input or a small set of terms provided by the user. Most of the times users end up providing a set of terms that are nothing more than a small text. The idea of the proposed method is to use summaries of documents and queries to make the comparisons. These summaries are also small texts that have a limited number of terms depending on the chosen method for the summaries generation. In a short text there is little context, so it is necessary to employ a method that gets good results without having to analyzing the context [1].

Current retrieval methods are still deficient in regard to the responses quality as shown by Fachin in [2] and Breitman in [3], which indicates a need forimprovements in the solutions as suggested in Souza's article [4]. Thus, this paper proposes the comparison of documents by semantic similarity of short texts using well known retrieval techniques and ontologies to deal with synonymy and polysemy problems. The TextSSimily method is simple and seeks results with higher quality and few numbers.

This article is organized as follows: in Section 2, the state of the art is presented; in Section 3, the proposed method is presented; in Section 4, the results achieved in the use of the method, and in Section 5, the conclusions of the article.

2. RELATED WORK
Much has been studied about similarity and semantic comparison of both documents and terms, because an accurate comparison or just syntactic shows few really relevant results in a viable amount that can be analyzed by the user to get what he needs. Therefore, this section presents papers and concepts about similarity metrics, and works on documents comparisons and retrieval.

To calculate similarity one may use one of the several metrics proposed in the literature. Among the main metrics some may be mentioned: vector comparison, strings comparison, time series and frequency values.

Comparison between stringscalculates the similarity based on the cost of doing characters insertion and removal operations in the string. One derived metric from that transforms each string intokens sets and establishes the similarity through operations between sets [5].

The organization and vector comparison presented in details by Baeza-Yates [6] is well known in literature and used primarily for complex objects, as XML documents. In order to obtain the similarity, formulas are used as the Euclidean distance or other distance metrics [7].

Frequency values Compares common terms from documents, measuring the frequency of them. Those frequencies determine the similarity. One metric very employed and similar to this one is the IDF (Inverse Document Frequency), where the frequency of unusual values is used to define the similarity [8].

The field of information retrieval has numerous studies on comparing and retrieval of documents. The first studies used vector analysis, and more currently, they rely on other more elaborate forms of comparison using theories of artificial intelligence, linguistics, ontologies and semantic mathematical modeling as in the case of latent semantic.

Among the techniques that are base in artificial intelligence to compare terms and documents, surface matching may be mentioned, based on the frequency metric values, and it can be used for both long and short texts. A summary of main works using artificial intelligence is presented by Manning and Schutze in [9]. Some studies employ the concept text co-

occurrence, as presented in the studies [10;11;12]. Sahami and Heilman in [13] expand the representation of compared sets using results from Web queries, and the similarity considering this expanded representation, improving the obtained similarity reducing errors. Another work, which uses variation in the set, is presented by Metzler in [1], that employs the concept of kl divergence to measure the similarity of words expanded setobtained from several Web pages consulted. Metzler also makes a comparison of the main methods of similarity focusing on small texts.

The comparison of document terms using ontologies, is done by identifying of correlation between ontology concepts. This correlation is measured by the comparison between properties, level of generality or specificity, and its relationship to other concepts. The calculation of similarity between ontologies done through the semantic matching by mapping of meaning among the concepts is presented by Giunchiglia and Shvaiko[14]. Using the measure of Jaccard, Brank *et al* in [15] describe the similarity based on common ancestors analysis. The work of Isaac et al in [16] presents an approach based on co-occurrence statistical measures that aim for the mapping between ontologies analyzing the instances.

The paper of Novelli and Oliveira in [17] presents an approach of documents comparison that analyzes the semantic similarity of complete documents, based on the structure, content and the descriptive ontology of the document to obtain the similarity.

There are still studies that use ontologies to assist the process of comparison, mainly for being able to deal better with synonymy and polysemy upon knowledge structuration through the definition of concepts and their relationships. Among the works that can highlight are those of Varelas et al [18] that describes how to use the Wordnet ontology to calculate the semantic similarity, and the work of Thiagarajan et al [19] that presents a set of semantic similarity metrics, based on ontologies and takes into account, in the calculations, the relationships between terms (entities).

The latent semantic approach is relatively recent in documents comparison and retrieval, and there are many studies in literature. The main features and operation of the method are presented in the studies [20;21;11]. This approach uses for the documents comparison selection of terms from each one. The chosen terms are usually the most important or the ones that characterize the document. Since it is a mathematical modeling, many variations of the model have been proposed to improve the performance and accuracy of the values obtained. One of these variations is the weight function employment done by Foronda in [22] and studies that make use of probabilistic latent semantic analysis as the study of Mendoça in [23].

Many are the metrics and methods presented in literature. However, when it comes to information retrieval, it is necessary to balance between quality results, performance and number of relevant results obtained. Even after much research developed, the field of information retrieval has not yet achieved a method that minimizes satisfactorily the amount of results maximizing the relevant content obtained, when taken into account the method performance parameter.

3. TEXTSSIMILY METHOD

The method of document comparison that measures the semantic similarity of documents by the comparison of the terms from summaries (short texts) generated from documents and queries. For the comparison between the terms, firstly, syntactic similarity is measured, and this value is considered as accepted if it is above a minimum value set by the user that calibrates the method. The syntactic similarity calculation uses the classic model of information retrieval with the vector organization presented by Baeza-Yates and Ribeiro-Neto in [6] and the metric known as LEdit presented in the study of Sankoff and Kruskal in [24]. For values not accepted, semantic similarity is calculated by comparing the terms taking into account their semantic correlations, using for that, executing comparisons between ontology concepts.

This method keeps a good performance, because not all terms of summaries need to be compared semantically, since the most common terms, either the ones that are just gender bending or just plurality bending, already have in most of the times good similarity values when only compared syntactically. Thus, only a few terms end up being analyzed semantically providing better results.

The method can work with different ontologies depending on the type of document or compared context, for example, it can identify documents that are of the article type from the medicine field.

Considering the documents language, summaries must be analyzed so that they are grouped together and compared with those from the same language easing comparisons and improving results. So, the summaries go through a simple process of categorization based on the identified language for each one. To identify the language several tools and algorithms already defined in the literature may be used.

The method is performed in three phases: documents preparation, syntactic comparison of the terms and semantic comparison of certain terms from the documents. These phases are detailed in the following sections and briefly presented in Fig 1.

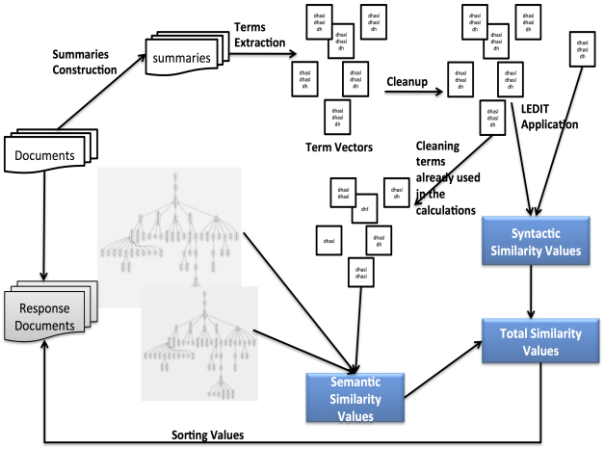

Fig 1: Method overview.

3.1 Documents Preparation

In this first step, documents are prepared for comparison. First, a summary is obtained from each document(s) from the considered repository(ies). These summaries development follows algorithms from literature. The summaries can be stored in files or databases, for some time, since the preparation of them takes a reasonable computational time

that can be suppressed by keeping them for use in other queries. After the summaries development, it is possible do the analysis for the languages definition, and keeping this information stored next to the summary.

The queries may be either a small set of terms or an entire document containing many pages. Thus, for the queries, summaries are also obtained; however, they should not be stored.

The summaries are read and the terms that will be used for similarity comparison are extracted, that is, they are transformed into string vectors containing all terms of each one of summaries, as shown in Fig 2. At this time, the removal of repeated terms in the vectors is performed.

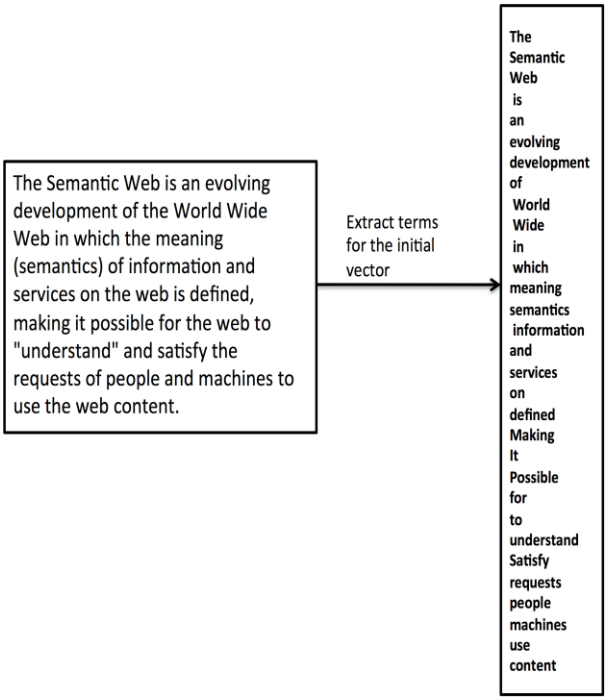

Fig 2: Document Summary and first vector.

Thereafter, the terms in uppercase letters are transformed into ones in lowercase letters, and the remaining terms that have become equal in the vector are removed. A cleanup algorithm is run on the resulting vector, eliminating the low relevance terms for comparison. This algorithm is specific for the language in which the documents are written. A resulting vector example from such cleanup is shown in Fig 3.

At the end of this phase, the vectors are ready to be compared, containing only one set of relevant terms for the comparison process.

3.2 Syntactic Similarity

This phase is responsible for calculating the syntactic similarity between terms of the vectors prepared in the previous step.

Similarity is calculated by comparing each term from the query vector with each one of the terms from vectors that represent the documents. As comparison result, equal and similar terms are found. Therefore, the first step in this phase is to find equal terms in the vectors, considering its similarity with value one, and then remove these terms of considered vectors in calculation.

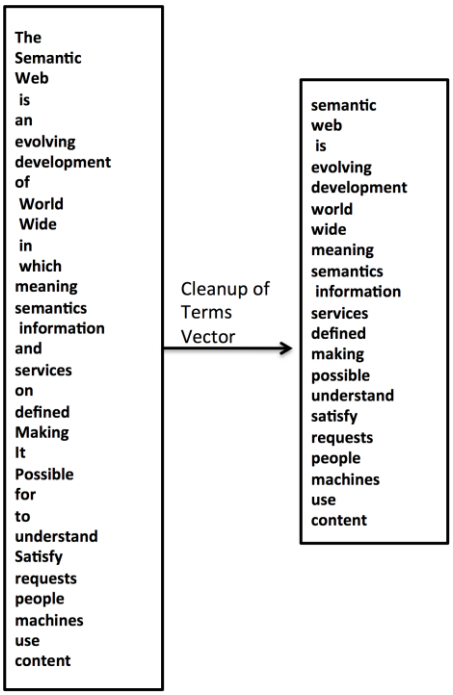

Fig 3: Vector Terms after cleanup algorithm application.

For the comparison of remaining terms is used a LEdit metric which calculates how many letters must be added, changed or deleted for a term to become another one, for example, in the terms "word" and "words" it is necessary to add a letter s and, it is returned, as the metric result, the value 1 and as similarity, 0.8. The value of similarity using this metric is anumber between 0 and 1, calculated by the formula below, in which A and B are two terms:

$$\mathbf{Similarity} = \mathbf{Max}\left(\mathbf{0}, \left(\frac{|\,\mathbf{MAX}(|\mathbf{A}|, |\mathbf{B}|) - \mathbf{Ledit}(\mathbf{A}, \mathbf{B})|}{(\mathbf{MAX}(|\mathbf{A}|, |\mathbf{B}|))}\right)\right)$$

For each one of the query terms, the comparison is not done sequentially considering a single term from each summaries vector, that is, for each t term from the query vector, all the other terms of document vector are compared to t, so that the best calculated result is considered when computing the syntactic similarity. After the values are calculated, the highest value obtained will be analyzed to verify whether it reached the minimum limit that was set for the method. In case this limit is reached, this term with highest similarity to t is taken from the document vector along with t from the query vector. An example of this calculation execution is shown in Fig 4. This way of comparison differs from the others that mostly use the idea of matching between vectors or even the query image projection in the summary vectors, obtaining, thus, better result values for each term by analyzing all possibilities.

The calibration of the minimum limiting value is done by user and it can be different for each language or context.

3.3 Semantic Similarity

In this phase, it is compared the existing semantics between the terms, considering their correlations, as synonyms, antonyms and plurality.

For each context or language, a different ontology should be used, since in this method the language detection context is not automatic and must be input by the user. Each one of these ontologies may be organized differently or to have only

portion of concepts. Therefore, more than one ontology may be required when of making the analysis.

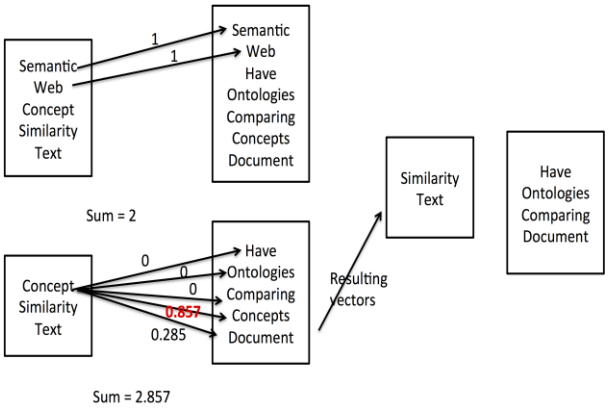

Fig 4: Two steps of syntactic similarity calculation.

The semantic similarity is firstly calculated by obtaining the correlations of the considered term in the ontology(ies). The semantic correlation analysis uses concepts previously proposed in the literature as in the studies [14;18;19].

These correlations are used for comparison with the terms from the document vector. When a correlation is found in the vector, it is analyzed and, depending on the correlation, a similarity is attributed to it, for example, if it is a synonym term, the similarity is considered to be one.

Obtaining the semantic correlation takes a considerable amount of time. Therefore, some results can be kept for a period of time, and they are for either terms that frequently occur or for recently searched terms, because some terms that tend to occur repeatedly and subsequently in more than one document when they deal with very similar issues. In this context, the searches are first done locally in "cache" and if the term is found, it is not necessary to perform other searches, improving the execution performance of the method.

3.4 Total Similarity

The final similarity computation is given by the sum of the values obtained from the syntactic and semantic similarities. In this paper, it was chosen not to weigh the values, still the semantic similarity has greater significance in the final result.

The sum is divided by the maximum size of the compared terms vectors.

Considering v1 and v2 two term vectors, the final similarity formula for the two vectors is given by:

$$\mathbf{Total\_Similarity} = \frac{\sum \mathbf{simSyntetic} + \sum \mathbf{simSemantic}}{\mathbf{Max}(|\mathbf{v1}|, |\mathbf{v2}|)}$$

At the end of the method execution, there is a vector of similarities of summaries ordered in decreasing order. For a search engine, the results presented to the user would be documents regarding the first summaries of the similarities vector, because these are the ones that hold the largest amount of relevant content based on the query provided by the user. The amount of documents that are part of the response also depends on of the method calibration set by the user. Thus, in some cases, few documents meet the request, however for some situations in which the obtained similarity values for of documents are very low, around 0.5, the user may choose to read more documents to obtain the contents he is seeking.

4. EXPERIMENTAL RESULTS

In this section it is presented some experimental results for the method shown in section 3.

Two kinds of experiments were done using a local repository containing a set of twenty-seven thousand documents in html format describing international movies.

The used ontology was WorldNet, because the text of documents is general and the repository documents are written in English. The analyzed correlations in the experiments were possible because of WorldNet ontology.The possible correlations ofWordnet are shown in Fig 5.

Semantic Relationship	Syntactic Category	Examples
Synonym (similar)	N, Aj, V, Av	Go up, ascend Sad, unhappy Fast, quick
Antonym (opposite)	Aj, Av (S,V)	Wet, dry High, low
Hyponym (subordinated)	N	Apple tree, tree Tree, plant
Hypernym (superordinate)	N	Tree, Apple Tree Plant, Tree
Meronym (part-of)	N	Ship, fleet Sleeve, shirt
Connection/Consequence	V	Drive, get ride Divorce, marry

Legend: N = noun, Aj = adjective, V = verb, Av = adverb

Fig5: Semantic relations of Wordnet [25].

For getting the summaries it was used an extractive method based on TF-IDF (Term Frequency-Inverse Document Frequency).

The first type of experiment was designed to validate the basic functionality of the method. A set of users developed some queries that were used to find the movies that came closest to what was requested by users. After the results, the same users were used to validate whether the results were really satisfactory.

Considering information retrieval systems, there are several measures that may be used, being accuracy and coverage the most common. Precision is the proportion of retrieved documents that are relevant to a certain query regarding the total documents retrieved. Coverage is the ratio between the number of retrieved documents that are relevant to a query and the total documents in the collection that are relevant to the query [6].

Subsequently, based on the most common measures for a system recovery, a second type experiment was conducted to verify the method coverage and accuracy. Initially, it was used all repository documents and some queries were generated from document repository fragments. From these queries, summaries were created. Afterwards, the experiment was performed with random removal of ten documents from the repository and the creation of their summaries. These summaries were used for comparison with the remaining documents.

The experiments, their results and the result analysis of are presented in the following sections.

4.1 Method Functionality

This experiment was done with a group of ten users attending university. For each user it was request query that was done from something they wanted to know about movies or about a particular film. This query should have at most a paragraph or a hundred words. The best queries were selected and used as input for the method, and comparisons used the entire

repository. To obtain answers, the method was calibrated to return only the first ten of the documents with the highest values of similarity, since these similarities were higher than 0.5, that is, the movies closer to what was searched, so that users could validate whether the result was really satisfactory. The results obtained in the experiment and are shown in Table 1.

Table 1. Users queries and acceptance of these results.

Some Relevant Query Terms	Average values of similarity of the top ten documents	Average Percentage of User Acceptance to the set of answers
Fatal accidents, problems or fights over inheritance, intrigue, back to life.	0583	70.58
Race cars, street racing and the death of people racing.	0.657	80
Separated love, sensitive person, people returning from the afterlife.	0.796	90.18
Death, person that haunts the other one, groups of people.	0.821	90
Security, police, people training, crimes, arrests.	0.997	100

The objective of this experiment was reached because, according to users, the searched content in their queries was answered with acceptance over seventy percent for a set of ten documents as response. It was demonstrated that for lower values of similarity, a small amount of documents couldn't achieve completely what the user is looking for. It was further observed that for very general terms, the achieved similarities values gets smaller and the contents cannot be recovered in few documents, requiring the user to do a more detailed query or that a much larger number of documents to be considered.

4.2 Accuracy and Coverage

In the first experiment ten queries were created from random fragments obtained documents from the repository. Using each of the ten queries, comparisons were made using the proposed method, between the query and all documents in the repository.

For this experiment results were considered the five highest values of similarities. The values considered relevant were all similarity values equal or higher than 0.7. The formula used to measure the accuracy was P@n, which measures the relevance of the first n documents in an ordered list:

$$p@n = \frac{r}{n}$$

in which, n is the number of returned documents, r is the number of documents considered relevant and returned to the n position of the ordered list.

This same experiment was performed using only one method of syntactic comparison with the application of the LEdit metric.The table 2 shows the values of accuracy and coverage of the proposed method compared with the values of the syntactic method, using queries generated randomly.

The accuracy when only the syntactic method is used is lower than the results when using the semantic method. Thus, by using a semantic method, it reaches the highest quality of obtained content in a small number of documents returned in the response.

Table 2. Proposed method Results compared with the syntactic method results.

Query	Accuracy of the Proposed Method	Coverage of the Proposed Method	Accuracy syntactic method
1	0.4	1	0
2	1	0.83	0.2
3	0.8	1	0.8
4	1	0.83	0.8
5	0.6	1	0.4
6	0.8	1	0.8
7	0.4	1	0.2
8	0.6	1	0.2
9	1	0.71	0.8
10	0.8	1	0.4
Average	0.74	0.93	0.46

It was also found that the recovery of five documents in response has a good coverage by considering similarity values equal or higher than 0.70. From this value, it is possible to conclude that the method accomplishes its objective of obtaining a small number of documents in the response and getting a considerable amount of content per retrieved document.

Another experiment was conducted using full documents as input, taken from the same set of repository documents. In this experiment, it was explored the fact that the input query text has a better quality and therefore lead to a better summary containing more significant terms that would facilitate the achievement of a better set answer. For this experiment the five highest values of similarities were considered as results. The values considered relevant were all the similarity values equal or higher than 0.9. Table 3 presents accuracy and coverage valuesof the proposed method for the experiment.

Table 3. The proposed method results for queries using complete documents.

Query	Accuracy	Coverage
1	0.80	1
2	1	0.65
3	1	0.65
4	1	0.5
5	0.80	1
6	1	0.5
7	1	0.20
8	1	0.5
9	1	0.35
10	1	0.20
Average	0.96	0.55

The data in Table 3 shows that the results are more accurate when using a more significant amount of terms since the summaries are obtained from complete documents. However, it is more difficult to obtain a small number of responses, because documents can be much more analyzed and a larger quantity of them will be part of the response. For this experiment, the largest amount of documents that would meet the queries according to the criteria of having a similarity

higher than 0.90, would be 24 documents. This number, although large, is still small compared to the amount of documents present in the repository.

5. CONLUSIONS

This paper presented a semantic method for documents comparison, focusing on the use of short texts in the form of summaries to facilitate comparison and improve results. The method uses various techniques from literature and ontologies to obtain better results when it is compared large sets of documents that have any number of pages. Experimental results show that even dealing with short texts the method appears to have good accuracy and coverage. For query texts of any formulation, accuracy results are around seventy percent, and for queries from complete documents, the accuracy increases, getting around ninety-six percent. The results also show that the method cover is also good getting results higher than fifty percent, thus achieving the goal to minimize the amounts of results obtained and maximize the amount of useful content to the user.

6. REFERENCES

[1] D. Metzler et al, "Similarity Measures for Short Segments of Text" *Advances in Information Retrieval*, vol. 44, pp. 16-27, 2007.

[2] G. R. B. Fachin, "RecuperaçãoInteligente de Informação e Ontologias: um levantamentonaárea de Ciência da Informação," *BIBLOS*, vol. 23, no. 1, 2009.

[3] K. Breitman, *Web Semantica: A Internet do Futuro*.: LTC, 2006.

[4] R. R. Souza, "Sistemas de Recuperação de Informação e Macanismos de Buscana Web: Panorama atual e Tendências," *Perspectiva em Ciência da Informação*, vol. 11, no. 2, pp. 161-173, 2006.

[5] G. A. Navarro, "A Guided tour to Aproximate String Matching," *ACM Computing Surveys*, vol. 33, pp. 31-88, 2001.

[6] R. Baeza-Yates and B. Ribeiro-Neto, *Modern Informarion Retrieval*. New York: Addison-Wesley, 1999.

[7] J. C. P. Carvalho and A. S. Silva, "Finding Similar Indenties among Objects from Multiple mWev Sources.," *WIDM*, pp. 90-93, 2003.

[8] M. Weis and F. Naumann, "Detecting Duplicate Objects in XML Documents," *IQIS*, pp. 10-19, 2004.

[9] C. D. Manning and H. Schutze, *Foundations of Statitical Natural Language Processing*.: The MIT Press, 1999.

[10] L. Fitzpatrick and M. Dent, "Automatic Feedback Using Past Queries: Social Searching," *SIGIR*, pp. 306-313, 1997.

[11] T. Landauer et al, "An Introduction Latent Samantic Analysis," *Discourse Processes*, pp. 259-284, 1998.

[12] |P. D. Turney et al, "Mining the Web for Synonyms: PMI-IR versus LSA on TOEFL," *ECML 01*, 2001.

[13] M. Sahami and T. Heilman, "A Web_Based Kernel Function for Measuring the Similarity of Short Snippet," *WWW 06*, pp. 2-9, 2006.

[14] F. Giunchiglia and P. Shavaiko, "Semantic Matching," *The Knowledge Engineering Review Journal*, vol. 18, pp. 265-280, 2004.

[15] J. Brank et al, "Automatic Evaluation of Ontologies," *Natural Language Processing and Text Mining*, pp. 193-219, 2007.

[16] A. Isaac et al, "An Epirical Study of Instance- based Ontology Matching," *6th International Semantic Web Conference and 2nd Asian Semantic Web Conference*, pp. 253-266, 2007.

[17] A. D. P. Novelli and J. M. P. Oliveira, "ESimilyOnto: Um MétoroeficienteparaObtenção da Similaridade entre Documentos da Web Semântica," *Sinergia*, pp. 89-99, 2008.

[18] G. Varelas et al, "Semantic Smimilarity Methods in WordNet and their Application to Information Retrieval on de Web," *7th ACM International Workshop on Web Informatrion and data Management*, pp. 10-16, 2005.

[19] R. Thiagarajan et al, "Computing Semantic Similarity Using Ontologies," *International Semantic Web Conference*, 2008.

[20] M. W. Berry et al, "Using Linear Algebra for Intelligent Information Retrieval," *SIAM Review*, vol. 37, no. 4, pp. 573-595, 1995.

[21] S. Deerwester et al, "Indexing by Latent Semantic Analysis," *Journal fo The American Society for Information Science*, vol. 40, pp. 391-407, 1990.

[22] D. A. H. Foronda, "EstudoExploratório da IndexaçãoSemânticaLatente e das Funções Peso," *Dissertação de Mestrado*, 2005.

[23] D. S. Mendoça, "AnáliseProbabilistica de SemânticaLatenteAplicada a Sistemas de Recomendação," *Dissertação de Mestrado*, 2008.

[24] D. Sankoff and J. Kruskal, *Time Warps, String Edits, and Macromolecules: The Theory and practice of Sequence Comparison*. Nova York: Addison-Wesley, 1983.

[25]K. Breitman, M. A. Casanova e W. Truszkowski, Semantic Web: Concepts, Technologies and Applications, Srpinger, 2007.