Journal of Computer Science 2 (3): 261-268, 2006
ISSN 1549-3636

A Dynamic Weight Assignment Approach for Index Terms

[1,2]Kamel Eddine Haouam and [1]Farhi Marir
[1]School of Informatics & Multimedia Technology, London Metropolitan University, N7 8DB, U.K
[2]College of Computer and Information Sciences, King Saud University
P.O. Box 51178, Riyadh 11543, Saudi Arabia

Abstract: Currently in the development of Informational Retrieval (IR) systems, a set of keywords represents the semantics of a text document and after assigning weights to the set of keywords, they are used for indexing, searching and retrieval purposes. The current approaches for assigning weights are provided by an IR model that is used by an IR system. The main objective of the weight assignment was to provide ranking feature to an IR system. The retrieval performance of an IR system mainly depends on two parameters: extraction of a good set of keywords from text documents and the use of a good weight assignment approach. Most of currently available weight assignment approaches do not suggest any change to the weights of keywords after their initial assignment. It means that these approaches are static. In this study, we propose a dynamic weight assignment approach for weight assignment. In our opinion, using this proposed approach can be helpful in improving the retrieval performance of an IR system. This approach can be used as part of any IR model after initial assignment of weights.

Key words: Dynamic weight assignment, information retrieval techniques, RST, indexing

INTRODUCTION

The growth rate in the volume of information on the Internet is currently 300% per annum and if this present growth rate is maintained, then retrieval of relevant information will become a serious problem. Many efforts have been devoted towards developing information retrieval (IR) systems on the Web[1-4]. Despite of all these efforts by using currently available indexing techniques, it has been estimated that the average only 30% of the returned documents are relevant to the user's need and remaining 70% of these relevant documents in the collection are never returned[5]. These results are far from ideal and acceptable level. In the existing indexing techniques, keywords are used by and IR systems and search engines. In these techniques, each document in a collection is represented by a set of meaningful terms (also called *descriptors, index terms* or *keywords*) that are believed to express the content of the document. These keywords are assigned weights using the methods provided by an Information Retrieval (IR) model (such as Boolean Model, Vector Model, Probabilistic Model and their extensions) that is used in the development of IR systems[6-8]. The major drawback of the keyword-based indexing and retrieval techniques is that they only use a small amount of the information associated with a document as the basis in making relevant decisions. As a result, many irrelevant documents may be retrieved. To achieve a better retrieval performance, more semantic information about documents needs to be captured. Some attempts are made for improving the traditional indexing techniques using Natural Language Processing[9], logic[2,10] and document clustering[11] and they have gained some improvements.

The retrieval performance of an IR system mainly depends on two parameters: i) extraction of a good representative set of keywords from text documents, ii) weight assignment approach provided by an IR model. The weight assignment approaches provided by the available IR models are s*tatic*, which means that once weights are assigned to keywords, they do not change.

In IR systems, we can identify two main entities that have potential to influence the performance of the systems. The first gentility is a group of text documents writers and the second entity is a group of users that use the systems after their development. These two entities differ in their characteristics and participation in an IR system. In our opinion, we can improve the performance of an IR system by bringing closer these two entities. One way to bring them closer is by making dynamic weights of keywords and to achieve this purpose, we propose a dynamic weight assignment approach in this study.

Rhetorical Structure Theory (RST) defines relationships among different structures in a text document[12]. On the basis of the *cue phrases*, the rhetorical relationships between units of text documents are identified and they can be saved into a database. We can then query that collection of relationships using not only keywords, as traditional Information retrieval

Corresponding Author: Kamel Eddine Haouam, College of Computer and Information Sciences, King Saud University, P.O. Box 51178, Riyadh 11543, Saudi Arabia, Tel: 966 1 4679678, Fax: 966 1 4675423

systems (IRSs), but also rhetorical relationships. In this work, we use Rhetorical Structure Theory (RST) and its relationships for indexing and retrieval purpose instead of keywords and propose a dynamic weight assigned approach to the RST relationships. As mentioned earlier, our proposed dynamic weight assignment approach can also be used for assigning weights to RST relations in those IR systems which use RST relationships for indexing, searching and retrieval purposes. In this study, we use our proposed approach for the weight assigning to the RST relationships considering them as index terms of the collection. In this study, we also study the performance of these relationships using our proposed approach.

Related work: Here, we describe some available weight assigning approaches and their analysis. We have divided the weight assignment approaches into two main categories. These two categories use keywords for RST relationships, respectively for indexing, searching and retrieval purposes. Here, we also give the basics of Rhetorical Structure Theory (RST) and its relationships. Currently available weight assignment approaches of both categories are *static*, which means that once weights are assigned to keywords or RST relationships in an IR system, they remain unchanged in the life-span of the IR system.

Weight assigning approaches to keywords: The classical Information Retrieval (IR) models (Boolean, Vector and Probabilistic) and their extensions provide weight-assigning approaches as a part of these IR models[13]. Both Boolean and Probabilistic IR models assign weights to keywords, extracted from a collection of text documents after the text operations, from the binary set *{0, 1},* whereas the Vector model assigns weights from the closed interval *[0, 1]*[4,11,13]. Further details about the IR models, their extension and their weight assignment approaches can be seen in[13]. Note that all classical IR models and their extensions suggest static weight assignment approaches.

Rhetorical structure theory (RST): The need for an efficient document structuring was first realized by Aristotle and he recognized that in coherent documents, parts of text could be related in a number of ways[14]. Many researchers have pursued this idea and developed theories to relate sentences of text document. Among these theories, the theory developed by Mann & Thompson, called Rhetorical Structure Theory (RST), which has many interesting characteristics[12]. This theory postulates the existence of about twenty-five (25) relationships based on the view that these relationships can be used in a top-down recursive fashion to relate parts and sub-parts of a text. RST determines relationships between sentences and through these relationships the text semantics can be captured. In Table 1, we give some of the RST relationships[12,15]). Also, these relationships can be identified by *cue words* in text. This top-down nature of the RST relationships means that text documents can be decomposed into sub-units containing coherent sub-parts with their own rhetorical structure, therefore, opens up the possibility of extracting only relevant information from the text documents.

RST is a linguistically useful method for describing text documents and characterizing their structure. It explains a range of possibilities of structure by comparing various kinds of "building blocks"[16] that can be observed in text documents. Using this theory, two spans of text (adjacent in most cases, but exceptions can be found) are related such that one of them has a specific role relative to the other. For example, an evidence for the claim follows a claim. The claim spans a *nucleus* and the evidence spans a *satellite*. The order of these spans is not constrained, but there are more likely and less likely orders for all of the RST relationships. A general format of a RST relationship and its two spans are shown in Fig. 1.

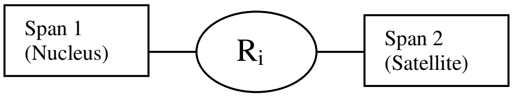

Fig. 1: General view of a RST relationships between its two spans

Four (4) types of constraints are used in describing RST relationships, which are listed as follows:

- Constraints on the nucleus
- Constraints on the satellite
- Constraints on the combination of nucleus and satellite
- The effect

Text coherence in RST is assumed to arise due to a set of constraints and an overall effect that are associated with each relationship. These constraints can operate on the nucleus (N), the satellite (S) and the combination of nucleus and satellite (N+S). For an example, we give the definition of the relationship Evidence as follows:

Relationship Name: Evidence

Constraints on N: The reader R might not believe the information that is conveyed by the nucleus N to a degree of satisfaction to the writer W.

Constraints on S: The reader believes, the information that is conveyed by the satellite S will find it credible.

Constraints on: The N+S combination: R’s comprehending S increases R’s belief of N.

The effect: R’s belief of N is increased.

Locus of the effect: N

Table 1: Some common RST relationships and their spans

Relationship Name	Nucleus	Satellite
Contrast	One alternative	The other alternative
Elaboration	Basic information	Additional information
Background	Text whose understanding is being facilitated.	Text for facilitating understanding
Preparation	Text to be presented	Text which prepares the reader to expect and interpret the text to be presented.
Antithesis	Ideas favored by the author	Ideas disfavored by the author
Circumstance	Text expressing the events or ideas occurring in the interpretative context	An interpretative context of situation or time
Condition	Action or situation resulting from the occurrence of the conditioning situation	conditioning situation

Rhetorical relationships can be represented as the rhetorical tree-structures (called RS-trees) that are organized into the five (5) schemas as shown in Fig. 2[12].

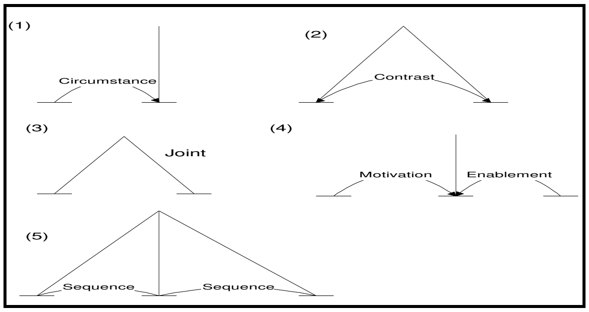

Fig. 2: RST five (5) schemas

There are four (4) criteria that determine the well-formedness of an RST tree[12]. These four criteria are given as follows:

Completeness: A single tree covers the entire text.

Connectedness: Each text span in a text, with the exception of the text span that covers the entire text, is a node of the tree.

Uniqueness: Text spans have a single parent.

Adjacency: Only adjacent text spans can be grouped together to form larger text spans.

Cue phrases: Cue phrases are words that connect two or more spans and add structure to the discourse of text, for example, some cue phrases are given: “first”, “and”, “now”, “accordingly”, “actually”, “also”, “although” etc. Marcu created a set of more than 450 cue phrases[15,17,18]. Also, Simon H Corston-Oliver describes a set of linguistic cues that can be identified in a text as an evidence of discourse relations[19].

Mann and Thompson recognize that rhetorical relationships are often signaled by cue words and phrases, but emphasize that rhetorical relationships can still be found even in the absence of such cues[12]. This connective provided by cue words and phrases can be used to determine rhetorical relationships between elementary units and between large spans of text. Using only the knowledge of cue phrases, an algorithm may be able to hypothesize the rhetorical relationships. For example, the relationship *Contrast* (Table 1) can be hypothesized on the basis of the occurrence of the cue word “but”, “however” etc. In Table 2, we give a sample set of the cue phrases.

Table 2: Set of sample cues phrases

Contrast	Whereas, but, however.
Elaboration	Also, sometimes, usually, for-example.
Circumstance	After, before, while.
Condition	If, unless, as long as.
Cause	Because, since.
Concession	Although, without, even-though.
Sequence	Until, before and later, then.
Purpose	In order to, so, that.

O'Donnell weight assignment approach: O’Donnell proposed a weight assignment scheme to the RST relationships and this approach first extracts RST relationships in pairs from a text (a collection of documents)[20]. Then, these extracted relationships are transformed into a tree structure, in which a node denotes a pair of RST relationships and a level of the tree denotes hieratical structure of a text. The weight to the pair of relationships at a root node is assigned intuitively and weights to nodes at the lower levels are assigned as product of the weights of the relationship pair at its immediate upper level. The weights to the RST relationships are assigned as a real number from the interval *[0, 1]*. It is also a static weight assignment approach.

There are, however, some cases where a weight assignment approach breaks down – non-clarity does not always reflect the centrality of information. Sometimes an author of a text writes information in the text at a rhetorically unimportant place, yet that information may be needed later to understand the argument. Other investigators have applied similar approaches for weight assignment to RST relationships[21,22].

Semantic vector space model: Liu has proposed a Semantic Vector Space Model (SVSM) for text representation and searching based on the combination of Vector Space Model (VSM), heuristic syntax parsing and distributed representation of semantic case structures[7]. In this model, both documents and queries are represented as semantic matrices. A search mechanism is designed to compute the similarity between the two semantic matrices (documents and query) to predict the level of relevancy. A prototype system is developed to implement this model by modifying the SMART system and using the Xerox Part-of Speech (P-O-S) tagger as the pre-processor of the indexing process. The prototype system is used in an experimental study to evaluate this technique in terms of precision, recall and effectiveness of relevance ranking. The results of the study showed that if documents and queries are too short (typically less than two lines in length), then the technique is less effective than VSM. But with longer size documents and queries, especially when original documents are used as queries, it is found that the system gives significantly better performances than SMART.

SMART system is one of the first and the best available IR system. It was developed by Gerard Salton at Cornell University using the vector space model for representing and querying documents. This system performs text operations such as removing stop word from a predetermined list, stemming via suffix deletion and weight assigning for the indexing purpose. It converts a given query to a vector and then measures the similarity between query and the documents in the vector space. The SMART IR system ranks the documents and returns the top *n* relevant documents, where *n* is a number given by the user. It can perform relevance feedback based on the result of the retrieval. Its weight assignment approach is static.

Need for a dynamic weight assignment approach: In an IR system, we identify two separate and independent entities: i) Entity-I and Entity-II. Entity-I consists of a set (or a group) of text writers (or authors), who write (or wrote) text that is placed as a collection of text documents on the IR system. Entity-II consists of a group of users who search and retrieve the stored collection through their queries. These two identified entities influence the relevance results and consequently affect retrieval performance (as precision and recall) of an IR system. We consider these two entities as two independent entities/parameters due to their different characteristics and can get a better retrieval performance if we some way are able to overlap their characteristics, or tune-up them. In other words, retrieval performance of an IR system is directly proportional to the degree of overlapping the characteristics of the two entities or their tuning.

Now we give and discuss the characteristics of Entity-I (a group of text writers). A text document reflects the writing style of a writer and it also affects index terms of the text documents. A writing style of a writer mainly dependents upon the following three factors:

* Personality of writer
* Knowledge and vocabulary size of writer
* Application domain of IR system

These three factors may not be independent, especially factors (i) and (ii). These two factors mean that a writer uses his/her personal preferences in selecting words and grammar rules of the language in his/her writing text based on his/her own knowledge and vocabulary size of the language. Generally, any writer uses a sub-vocabulary of total size of his/her vocabulary while writing. The third factor, application domain of IR system, also influences the writing style of a text writer. For example, a writer will use two different sets of words, terminology and semantics in writing text documents for two IR systems such as medical IR system and geographical IR system.

These three factors influence the writing text documents of a text writer and also the presence of index terms (keywords or RST relationships) especially if index terms are extracted using the RST relationships. For instance, one writer may use more a certain group of RST relationships in his/her text than another writer.

Now we list the characteristics of Entity-II, that is, a group of users of an IR system as follows:

* Incomplete knowledge about the collection in an IR system
* Knowledge level of a user
* Incomplete and vague user needs

We know that sometime users do not have complete knowledge and information about the IR system which they are going to use for their retrieval of their needs. They also know their needs most of the times in vague and incomplete form as keywords. One reason of this can be that diversified kinds of users use IR systems for retrieval of their needs. Their knowledge about information technology and about target IR systems differs.

As we have mentioned earlier, the two entities (Entity-I and Entity-II) function are in isolation without any cooperation among themselves. We argue that a good retrieval performance can be achieved if in anyway, we could able to overlap and match the above-mentioned characteristics of these entities, or make them to cooperate. The characteristics of Entity-I are not controllable especially after development of an IR system because their characteristics and requirements are captured only at the time of their development.

We know that most of IR systems are generally *semi-static*. It means that once a text document is put in an IR system, the text document remains unchanged for a long period of time. Also, the index terms and their weights in an IR system are unchanged after its development. We also know that the performance of an IR system mainly depends on a good use of approaches for the selection of index terms and assigning weights (or an IR model) to index terms. Therefore, if we improve one of these two approaches, we can get improvement in the retrieval performance of IR systems. In this study, we propose improvements to the weight assignment approach by making Entity-I to cooperate with Entity-II. For this purpose, we propose a dynamic weight assignment approach that can change weights of index terms of the collection (Entity-I considering it semi-static) using user needs (Entity-II). This change in weights is at the run-time after their initial assignments assigned by IR model. For the selection of a good set of index terms, we use RST relationships which are extracted from the collection of an IR system.

Proposed dynamic approach for assigning weights: We propose an approach for assigning weights to the RST relations or keywords (that are referred to as *Index Terms*). This approach is a dynamic and self-learning approach. It makes dynamic the weights of index terms because they evolve over time in an IR system. The change in weight of an index term depends on the use of the index term and we refer this process of evolution of weights to as *self-learning* characteristic of our approach. This approach consists of the following two steps.

Initialization step: In the beginning at the time of system generation, all index terms that are detected in each text documents in the collection, are assigned a weight by the IR system developer. The initial weights can be assigned using one of the following schemes:

* Assign a value using some random scheme
* Intuitively or with the consultation of a linguistic specialist.
* Same weight to each index term

This step assigns initial weights to all index terms using one the above-mentioned scheme. This assignment of weights to index terms by this step is one time task in the life-span of an IR system through human interaction. The main objective of this step is to initialize index terms of an IR system. Later, these initialized weights may evolve as described in Step 2.

Self-learning step: After the initialization of weights, weight of an index term is increased every time the index term is referred by a user query. In other words, weight of an index term is incremented, whenever it is referred by a user query, otherwise, it is unchanged. After using an IR system for some time, the weight of the RST relations may achieve stable levels. The weights of those index terms that are referred more frequently will have higher values than the weights of those index terms which are referred less frequently by user queries. The dynamic and evolution of weights of Index terms in an IR system will mainly depend on the usage pattern of the IR system.
Suppose that in Step 1, the weight Wi *is assigned to the index term* $\mathrm{IT_i}$. *In Step 2, the weight* $\mathrm{W_i}$ *of the index term* $\mathrm{IT_i}$ *is incremented provided a user query makes a reference to the index. This increment in the weight is defined as:*

$$\mathbf{W_i} = \mathrm{W_i} + \text{increment} \qquad (1)$$

In Equation (1), increment *is the increment in the weight after each reference to the index term and* $\mathrm{W_i}$ *and* $\mathbf{W_i}$ *(bold) are the weights of the index term* IT_i *before and after (or previous and latest) the increment. This increment in weight of an index term is defined as the function of time duration between two consecutive references to an index term, as follows:*

$$increment = W_{i,} * 1/(\,|t_c - t_p|)^n \qquad (2)$$

where $|t_c - t_p| <> 1$

In Equation (2), W_i is the weight of the index term IT_i before the current increment and t_c and t_p are current time and previous time when the index term is referred by current query and previous query, respectively. If there exists the boundary condition, $|t_c - t_p| = 1$, then there is no increment in the weight of then index term. The time duration $|t_c - t_p|$ in Equation (2) ensures that, if an index term is referred frequently, then it should get more and linear increment in its weight than an index term which is referred less frequently. In Equation (2), the real number n is the controlling (or normalizing) power that restricts an index term to reach its maximum weight value, 100, rapidly. The value of n can be tuned at any time to control the rate of increment in weights by increasing the value of n. The value of n is enumerated by the following empirical formula giving in Equation (3).

$$n = ceiling\ (\sqrt{(\mathrm{Wi},}\ /2) \qquad (3)$$

Minimum *0.1* increment is allowed in a weight and if the increment is less than *0.1*, then the index term retains its previous weight. Also, if weight of an index term attains its maximum value which is *100*, then there will be no further increment in the weight of that index term.

System stabilization: As mentioned earlier, The Initialization Step assigns the initial weights to all index terms of a collection of documents. Later, Self-Learning Step continuously updates the weights of the index terms, whenever, they are referred by the user

Table 3: Changes in weights when randomly initial weight assignment at time instance 0

Relationship/ Time	Preparation (R1)	Background (R2)	Elaboration (R3)	Contrast (R4)
0	80.00	90.00	70.00	50.00
1	80.00	90.00	70.00	50.00
3	**82.50**	**92.81**	**72.19**	**53.13**
5	**85.08**	92.81	72.19	**56.45**
6	85.08	92.81	72.19	56.45
9	**85.43**	92.81	**72.49**	56.45
11	**88.10**	**95.71**	**74.75**	56.45
13	88.10	**98.70**	74.75	**59.98**
15	88.10	98.70	**77.09**	**63.73**
17	**90.77**	**100.00**	**79.49**	63.73
18	90.77	100.00	79.49	63.73
20	**93.61**	100.00	**81.97**	**67.71**
22	93.61	100.00	**84.54**	**69.83**
24	93.61	100.00	**87.18**	**72.01**

Table 4: Changes in weights after intuitively initial weight assignment at time instance 0

Relationship/Time	Preparation (R1)	Background (R2)	Elaboration (R3)	Contrast (R4)
0	87.00	34.00	66.00	17.00
1	87.00	34.00	66.00	17.00
3	**89.72**	**38.25**	**68.06**	**19.13**
5	**92.52**	38.25	68.06	**21.52**
6	92.52	38.25	68.06	21.52
9	**92.90**	**38.72**	**68.34**	21.52
11	**95.80**	**41.14**	**70.48**	21.52
13	**98.80**	**43.71**	70.48	**24.21**
15	**100.00**	43.71	**72.68**	24.21
17	100.00	**46.44**	**74.95**	24.21
18	100.00	46.44	74.95	24.21
20	100.00	46.44	**77.29**	**27.24**
22	100.00	46.44	**79.71**	**30.64**
24	100.00	**49.34**	**82.20**	**34.47**

Table 5: Changes in weights when same initial weights are assigned at time instance 0

Relationship/ Time	Preparation (R1)	Background (R2)	Elaboration (R3)	Contrast (R4)
0	50.00	50.00	50.00	50.00
1	50.00	50.00	50.00	50.00
3	**56.25**	**56.25**	**56.25**	**56.25**
5	**63.28**	56.25	56.25	**63.28**
6	63.28	56.25	56.25	63.28
9	**65.62**	56.25	**58.33**	63.28
11	**69.73**	**63.28**	**65.63**	63.28
13	69.73	**71.19**	65.63	**71.19**
15	69.73	71.19	**69.73**	**75.64**
17	**73.83**	**75.64**	**74.09**	75.64
18	73.83	75.64	74.09	75.64
20	**78.44**	75.64	**78.72**	**78.00**
22	78.44	75.64	**83.64**	**80.44**
24	78.44	**80.37**	**88.87**	**82.96**

Table 6: Average increment in all three initial weight assignment techniques

Relationship/Time	Intuitive	Random	Identical
0	72.50	51	50
1	72.50	51	50
3	**75.16**	**53.79**	**56.25**
5	**76.63**	**55.09**	**59.77**
6	76.63	55.09	59.77
9	**76.79**	**55.37**	**60.87**
11	**78.75**	**57.24**	**65.48**
13	**80.38**	**59.3**	**69.44**
15	**81.9**	**60.15**	**71.57**
17	**83.5**	**61.4**	**74.8**
18	83.5	61.4	74.8
20	**85.82**	**62.74**	**77.7**
22	**86.99**	**64.2**	**79.54**
24	**88.2**	**66.5**	**82.66**

queries. This process of updating (or Self-Learning) continues with life-span of an IR system. The stabilization of an IR system and hence better performance of the system depends upon one of the following two factors or both.

- * Heavy use of an IR system
- * Age of the system

In other words, the more the system is used, the better the performance and the system gets the *level of stabilization*. The level of stabilization of the proposed weight assignment approach means that when an IR system starts giving good retrieval results in terms of Recall and Precision.

Case study: This case study was done for two purposes; the first to study the performance of the RST relationships and the second to study the three initial weight assignment was given to index terms in the environment of our proposed approach.

Here, we give a case study to demonstrate our proposed dynamic weights assignment approach. We assign initial weight to the index terms in three (3) ways. The objective of this case study is also to compare these three ways for initial weight assignment to index terms, in this case, they are the RST relationships extracted from the above documents. Note that in this case study, we use RST relationships as index terms.

Document: Lactose and Lactase (1), Lactose is milk sugar (2), the enzyme Lactase breaks it down (3). For want of Lactase, most adults cannot digest milk (4) .In populations that drink milk; the adults have more Lactase, perhaps through natural selection (5).

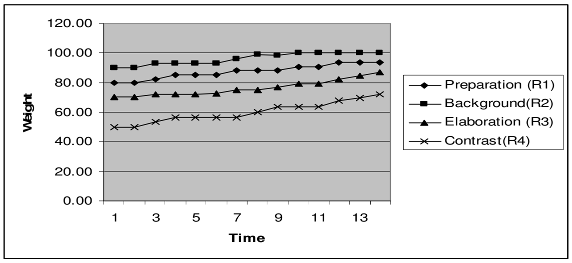

Fig. 3: Graph of Table 3

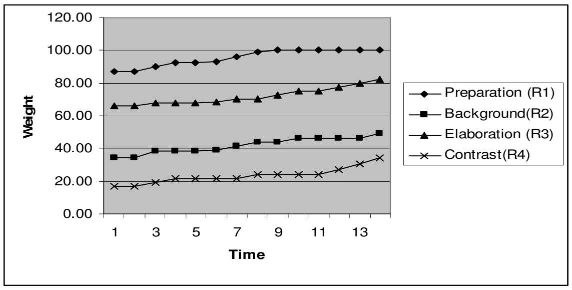

Fig. 4: Graph of Table 4

From the above document, the following set of RST relationships are extracted.

R1: Preparation,
R2: Background
R3: Elaboration
R4: Contrast,

Consider the following set of queries on the document. The time instances when these queries are posed, are randomly selected between the time interval *[0, 24]* of one day *and* the initial weights assigned to these RST relationships (or index terms) in the three ways: intuitively, randomly and same/equal value as shown in Table 3, Table 4 and 5, respectively. An increment in weight of a RST relationship occurs when the relationship is referred by a query and the increment is calculated by using the formula given in Equation (2). These increments are recorded in Table 3-5 and are shown in bold in these tables. To visualize the performance of this approach and these RST relationships, we have plotted these three tables separately as shown in Fig. 3-5.

The set of queries with their posing time instances:

At time 1:Query1(R1,R4)	
At time 3:Query2 (R2, R3)	At time 5: Query3 (R1, R4)
At time 6: Query4 (R2)	At time 9: Query5 (R1, R3)
At time 11: Query6 (R1, R2, R3)	At time 13: Query7 (R2, R4)
At time 15: Query8 (R3, R4)	At time 17: Query9 (R1, R2, R3)
At time 18: Query10 (R4)	At time 20: Query11 (R1, R3)
At time 22 Query12 (R3, R4)	At time 24: Query13 (R2, R3, R4)

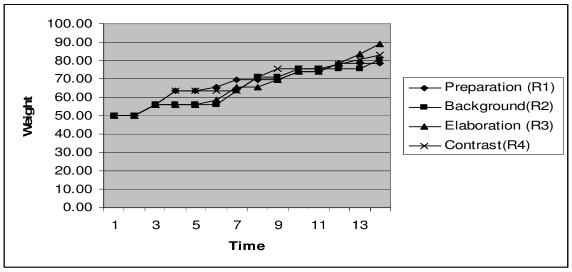

Fig. 5: Graph of Table 5

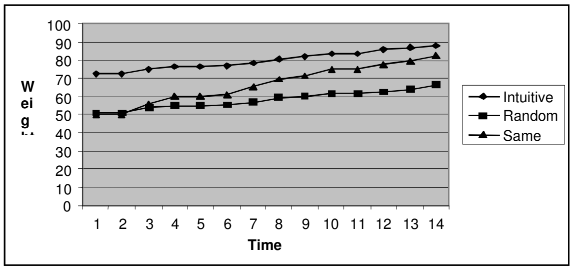

Fig. 6: Graph of Table 6

Observations: From these plotted graphs (Fig. 3-5), it is clear that these four RST relations have performed differently (in term of increments in their weights) and the reason for this is that those relationships that performed well, are referred more frequently by the queries. In other words, the performance of all these relationships is almost linear and also evident from these graphs are well-behaved graphs. Also, the performance of each relationship is quite close and similar.

We observe in Fig. 3-5, that the performance of an index term depends on the queries that reference it and the performance of the index term is influenced by the intensity of user needs for that index term. It means that by using this approach, the indexing can be improved and consequently the retrieval performance of an IR system as we pointed out earlier. Note that, as we said that, the rate of increment in the weights of index terms can be controlled by controlling the value of *n* in Equation 3.

In Table 6, we take the average increment in the weights of the four RST relationships recorded in Table 3-5, for each initial weight assignment way. This is done to look at the performance of each initial weight assignment way in our proposed dynamic environment. Table 6 is plotted as shown in Fig. 6. From Fig. 6, it is clear that performance of the intuitive initial weight assignment way performed better than other two ways, i.e., randomly and the same weight assignments.

CONCLUSION

We have proposed two important but independent identities (Identity-I and Identity-II) of any IR system and their characteristics and issues related to them. In our opinion if we can bring these two identities closer to each other, then we can achieve a better retrieval performance of an IR system. By consequently, we have proposed a dynamic approach for assigning weights to index terms after their initial weight assignments. A change in weight of an index term occurs when the index term is referred by a user query. This proposed approach can be used with any IR model that supports partial matching. We have demonstrates our proposed approach by a case study and compare three performance of the index terms (RST relationships) and the ways initial weight assignments in our proposed dynamic weight assignment approach.

Currently we are working on the extension of our proposed weight assignment approach and consider both keywords and the RST relationships of a collection for the purpose of indexing and refer it to as this indexing technique as composite dynamic indexing technique. Other interesting issues also need a serious attention to study the effects of the operations deletion and addition of documents in the collection of an IR system.

REFERENCES

1. Frants, V., I. Shapiro, J. Voiskunskii and G. Vladimir, 1997. Automated Information Retrieval: Theory and Methods. Academic Press, California.
2. Haouam, K. and F. Marir, 2003. SEMIR: Semantic indexing and retrieving web document using rhetorical structure theory. Proc. of the Fourth Intl. Conf. on Intelligent Data Engineering and Automated Learning (IDEAL2003), Hong Kong, pp: 596-604.
3. Pollitt, A.S., 1998. Information Storage and Retrieval Systems. Ellis Horwood Ltd., Chichester, UK.
4. Salton, G., 1989. Automatic Text Processing. Addison-Wesley, USA.
5. Sparck-Jones, K. and W. Peter, 1997. Readings in Information Retrieval. Morgan Kauffman, California, USA.
6. Korfhage, R., 1997. Information Storage and Retrieval. John Wiley and Sons, London.
7. Liu, G.Z., 1997. Semantic vector space model: implementation and evaluation. The J. Am. Soc. Inform. Sci., 48: 395-417.
8. Losee, R.M., 1997. Comparing boolean and probabilistic information retrieval systems across queries and disciplines. The J. Am. Soc. Inform. Sci., 48: 143-156.
9. Smeaton, A.F., 1992. Progress in the application of natural language processing to information retrieval. The Computer J., 35: 268-278.
10. Lalmas, M.B. and D. Peter, 1998. The use of logic in information retrieval modeling. The Knowledge Engg. Rev., 13: 263-295.
11. Hagen, E., 1997. An information retrieval system for performing hierarchical document clustering. Thesis, Dartmouth College.
12. Mann, W.C. and S.A. Thompson, 1998. Rhetorical structure theory: Towards a functional theory of text organization. The J. Text, 8: 243-28.
13. Baeza-Yates, R. and B. Ribeiro-Neto, 1999. Modern Information Retrieval. Addison-Wesley Publishing Company.
14. Aristotle, 1954. The Rhetoric, in W. Rhys Roberts (translator). The Rhetoric and Poetics of Aristotle, Random House, New York.
15. Marcu, D., 1996. Building up rhetorical structure trees. Proc. Thirteenth Natl. Conf. on Artificial Intelligence, 2: 1069-107, USA.
16. Vadera, S. and F. Meziane, 1994. From english to formal specifications. The Computer J., 37: 9.
17. Marcu, D., 1997. The rhetorical parsing of natural language texts. Proc. 35th Ann. Meeting of the Association for Computational Linguistics (ACL-97), pp: 96-103.
18. Marcu, D., 2000. The theory and practice of discourse parsing and summarization. Proc. 35th Ann. Meeting of the Association for Computational Linguistics (ACL-97), pp: 96-103.
19. Simon, H Corston-Oliver, 1998. Computing representations of the structure of written discourse. Technical Report MSR-TR-98-15, Microsoft research, Microsoft Corporation, One Microsoft way, Redmond, WA 98052.
20. O'Donnell, M., 1997. Variable-length on-line document generation. Proc. Flexible Hypertext Workshop of the Eighth ACM International Hypertext Conference, UK.
21. Marcu, D. and A. Echihabi, 2002. An unsupervised approach to recognizing discourse relations. Proc. 40th Ann. Meeting of the Association for Computational Linguistics (ACL-2002), USA, pp: 38-275.
22. Ono, K., K. Sumita and S. Miike, 1994. Abstract generation based on rhetorical structure extraction. Proc. 15th Intl. Conf. Computational Linguistics (COLING- 94), Vol. 1, Kyoto, Japan.