Noname manuscript No.
(will be inserted by the editor)

Term Frequency With Average Term Occurrences For Textual Information Retrieval

O. Ibrahim · D. Landa-Silva

Received: date / Accepted: date

Abstract In the context of Information Retrieval (IR) from text documents, the term-weighting scheme (TWS) is a key component of the matching mechanism when using the vector space model (VSM). In this paper we propose a new TWS that is based on computing the *average term occurrences of terms* in documents and it also uses a *discriminative approach* based on the document centroid vector to remove less significant weights from the documents. We call our approach *Term Frequency With Average Term Occurrence (TF-ATO)*. An analysis of commonly used document collections shows that test collections are not fully judged as achieving that is expensive and may be infeasible for large collections. A document collection being fully judged means that every document in the collection acts as a relevant document to a specific query or a group of queries. The discriminative approach used in our proposed approach is a heuristic method for improving the IR effectiveness and performance, and it has the advantage of not requiring previous knowledge about relevance judgements. We compare the performance of the proposed TF-ATO to the well-known TF-IDF approach and show that using TF-ATO results in better effectiveness in both static and dynamic document collections. In addition, this paper investigates the impact that stop-words removal and our discriminative approach have on TF-IDF and TF-ATO. The results show that both, stop-words removal and the discriminative approach, have a positive effect on both term-weighting schemes. More importantly, it is shown that using the proposed discriminative approach is beneficial for improving IR effectiveness and performance with no information in the relevance judgement for the collection.

Keywords Heuristic Term-Weighting Scheme · Random Term Weights · Textual Information Retrieval · Discriminative Approach · Stop-words Removal

1 Introduction

The term-weighting scheme (TWS) is a key component of an information retrieval (IR) system that uses the vector space model (VSM). An effective TWS is crucial to make an IR system more efficient. There are various TWS approaches proposed in the literature and some have been implemented in search engines. Perhaps the most widely used approach is the term frequency-inverse document frequency (TF-IDF). This paper proposes an alternative method called *Term Frequency With Average Term Occurrence (TF-ATO)* which is capable of removing less significant weights from the documents in the collection. The method is based on the average term occurrences of terms in documents and the document centroid.

Some Evolutionary Computation (EC) techniques have been used for evolving TWS or evolving term weights (Cummins, 2008; Cordan et al., 2003). However, such approaches have an important drawback as we discussed next. Usually these EC approaches use the relevance judgements for the document collection on their fitness functions for checking the quality of the

Osman Ali Sadek Ibrahim
School of Computer Science
ASAP Research Group
The University of Nottingham, UK
CS. Dept., Minia University, Egypt
E-mail: psxoi@nottingham.ac.uk

Dario Landa-Silva
School of Computer Science
ASAP Research Group
The University of Nottingham, UK
E-mail: dario.landasilva@nottingham.ac.uk

proposed solutions. The relevance judgement of a collection gives the list of relevant documents for every query. However, test and real IR document collections are usually not fully judged. This means that most documents in the collection are not relevant for any query in the query set. This provokes that when using EC techniques most documents have random term weights representations. This means that best effectiveness is achieved for only user's queries that are similar to the queries in the query set. But for user's queries that are different from those in the query set, only random effectiveness is achieved. In addition, TWS evolved with Genetic Programming (GP) as in (Cummins, 2008; Cordan et al., 2003) are based on the characteristics of the test collections and hence, not easily generalizable to be effective on collections with different characteristics. Moreover, these proposed EC techniques assume that document collections are static and not dynamic also. The dynamic nature of document collections on the web has also inspired the need for effective TWSs that do not depend on static characteristics of the collections.

Given the above, we argue that there is a need for heuristic methods to adapt term weights with little computational cost and a pre-determined procedure in order to achieve better IR system effectiveness and performance even when dealing with dynamic document collection. This is what motivates the work presented in this paper on the development of such a TWS. In this work we propose the *Term Frequency With Average Term Occurrence (TF-ATO)* method which computes the average term occurrences of terms in documents and uses a discriminative approach based on the document centroid vector to remove less significant weights from the documents. In the paper we evaluate the performance of TF-ATO and investigate the effect of stop-words (or negative words) removal (Fox, 1992) and the discriminative approach as procedures for removing non-significant terms and term weights in heuristic TWSs. These procedures do not depend on the relevance judgement.

The intended contributions of this paper are summarized as follows:

1. Based on an analysis of commonly used document collections, we provide an argument in favour of using heuristic (non-learning) TWS instead of TWS and term weights evolved with evolutionary computation techniques (subsection 3.1.2). We believe that this analysis also supports the argument that more appropriate test document collections, instead of general IR test document collections, need to be considered when using EC techniques for evolving TWS or evolving term weights.
2. We propose a new TWS approach called *Term Frequency With Average Term Occurrence (TF-ATO)* and a *discriminative approach* to remove less significant weights from the documents. We conduct a study to compare the performance of TF-ATO to the widely used TF-IDF approach using various types of document collections such as sampled, pooled (Soboroff, 2007) and from real IR systems (Hersh et al., 1994). Our experimental results show that the proposed TF-ATO gives higher effectiveness in both cases of static and dynamic document collections.
3. Using various document collections, we study the impact of our discriminative approach and the stop-words removal process on the IR system effectiveness and performance when using the proposed TF-ATO but also when using the well-known TF-IDF. We find that these two processes have a positive effect on both TWSs for improving the IR performance and effectiveness.

The remainder of this paper is organised as follows. Section 2 gives some key background knowledge on IR systems. Then, the proposed TF-ATO and discriminative approach are presented in Section 3. That same section presents the experimental results comparing TF-ATO to TF-IDF. Section 4 is dedicated to the study on the impact of stop-words removal and the discriminative approach. For better readability, we have decided to include the review of related work within the corresponding section. Also, detailed results of the experimental study in Section 4 are presented in the Appendix. Finally, conclusions and future work are presented in Section 5.

2 Background

2.1 General Information Retrieval Approach

An Information Retrieval (IR) system is an information system that stores, organizes and indexes the data for retrieval of relevant information responding to a user's query (*user information need*) (Salton and McGill, 1986). Basically, an IR system contains the following three main components (Baeza-Yates and Ribeiro-Neto, 2011):

- **Document Collection**. It stores the documents and their representations of *information content.* It is related to the indexer module, which generates a representation for each document by extracting the document features (terms). A *term* is a keyword or a set of keywords in the document.

- **User Information Need**. It is a user's query or set of queries so that users can state their information needs. Also, this component transforms the users query into its *information content* by extracting the query's features (terms) that correspond to document features.
- **Matching Mechanism**. It evaluates the degree of similarity to which each document in the document collection satisfies the user information need.

2.1.1 IR Architecture

The implementation of an IR system may be divided into a set of main processes as shown in Figure 1. Some of these processes (dotted lines rectangles) can be implemented using a machine learning or meta-heuristic approach. An outline of the core processes (solid lines rectangles) in Figure 1 is given next.

The *User Interface* module manages the interaction between the user and the IR system. With this module the user can request information after *Pre-processing* and *Query Transformation* from the index file. The result of this query is in the form of links or document numbers referring to documents in the document collection. The *Pre-processing* module represents the lexical analysis, stop-words removal and stemming procedures that are applied to the user's query and the document collection. The *Indexing* module processes the documents in the collection using a term-weighting scheme (TWS) in order to create the index file. Such index file contains information in the form of inverted indexes where each term has references to each document in the collection where that term appears and a weight representing the importance of that term in the document. Similarly to indexing, the user's query undergoes a process of *Query Transformation* after pre-processing for building queries of terms and their corresponding weights for those terms. The *Searching* module conducts a similarity matching between the query of terms with their weights and the index file in order to produce a list of links or document numbers referring to documents in the document collection. In the past, the *Ranking* of the matching list depended only on the degree of similarity between the documents and the user's query. Nowadays, this ranking may depend on some additional criteria such as the host server criteria among others (Liu, 2009).

After outlining the core processes in the implementation of an IR system, we now focus on the aspect where machine learning and meta-heuristic techniques exhibit some weakness in our opinion. The *Relevance Judgement* file is the file that contains the set of queries for the document collection and their corresponding relevant documents from the collection. Also, this file sometimes contains the degree of relevancy of documents for the queries (i.e. some number indicating that the document is non-relevant, partially or totally relevant). However, all IR test document collections are partially judged as it is not feasible to have fully judged document collections as mentioned in (Qin et al., 2010). Since machine learning and meta-heuristic techniques applied to IR depend on the relevance judgement file, the efficiency of such techniques for IR is limited, we discuss this in more detail in subsection 3.1.

2.1.2 IR Models

The way in which the IR system organizes, indexes and retrieves information from document collections is referred to as the IR model. The IR model also specifies the method used for the user's query representation. From the literature, there are three prominent IR models: the Boolean model, the Probabilistic model and the Vector Space Model (VSM) (Baeza-Yates and Ribeiro-Neto, 2011). A number of extensions of the these models have been built in machine learning for text classification, IR and sentiments analysis among others such as (Kaden et al., 2014; Winkler et al., 2014; Joachims, 1998; Zhou et al., 2009).

The *Boolean Model* is based on binary algebra for representing the term weights in documents and queries. In this model, the indexing module uses binary indexing for representing terms for each document (i.e., 1 if the term exists in that document and 0 otherwise). Queries are expressed as logical statements using logical operators OR, AND and NOT (e.g. term1 AND term2 NOT term3). The limitations on this model are that (Vinciarelli, 2005; Greengrass, 2000): (1) it needs a full matching between the user's query and the document collection; (2) there is no difference expressed in the information content of terms in documents or queries even if one term is repeated frequently and another term occurs once; and (3) it can be difficult to formulate a complex users information need using logical operators only.

The *Probabilistic Model* uses probability approaches to estimate the probability of a document being relevant to a certain query or not. Also, this model uses the probability of relevancy to a query for assigning weights to terms in documents and queries according to the queries training set or according to supervised weight learning. The limitation of the probabilistic model lies in the large set of queries used as a training set. The difficult and time-consuming aspects of the estimating mechanism are other limitations of the probabilistic model (Vinciarelli, 2005; Greengrass, 2000).

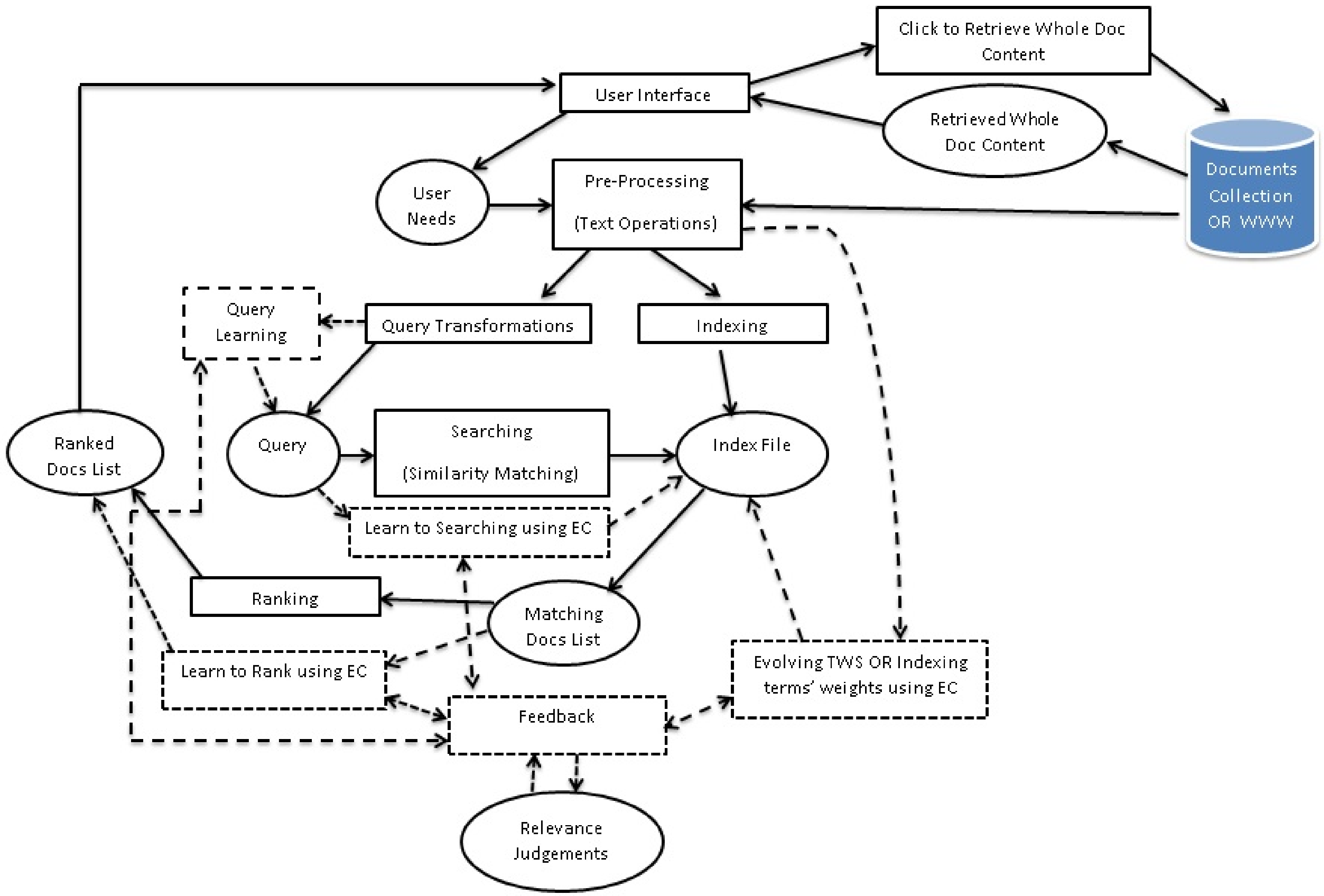
Click to Retrieve Whole Doc Content
User Interface
Retrieved Whole Doc Content
Documents Collection OR WWW
User Needs
Pre-Processing (Text Operations)
Query Learning
Query Transformations
Indexing
Ranked Docs List
Query
Searching (Similarity Matching)
Index File
Learn to Searching using EC
Ranking
Matching Docs List
Learn to Rank using EC
Evolving TWS OR Indexing terms' weights using EC
Feedback
Relevance Judgements

Fig. 1 Main Processes Involved in the Implementation of an IR System Architecture.

In this paper, we use the *VSM* which is the most widely applied model by researchers (Vinciarelli, 2005; Greengrass, 2000). In this model, a document and a query are represented as vectors in an n-dimensional space, where n is the number of distinguishing terms that are used as index terms for the documents in a collection. The values of the document dimensions are the weights of the index terms in the documents space. The VSM model has been extended into some other models using machine learning and mathematical approaches (Greengrass, 2000; Manning et al., 2008). Examples of these extended models are Support Vector Machine (SVM) and Latent Semantic Indexing (LSI) models (Greengrass, 2000; Manning et al., 2008). However, the VSM is simple and efficient in search engines compared to these other extended models. In addition, LSI has a limitation in respect of the document space size (term-document vectors matrix size) (Greengrass, 2000) and SVM has a limitation in respect of relevance feedback of the document collection. Thus, contrary to these extended models, the VSM has been used widely in open source search engines such as (Middleton and Baeza-yates, 2007; Lemur) and in IR index libraries such as (McCandless et al., 2010). The similarity matching between documents vectors and the user's query vector can be measured using a similarity function. There are many similarity functions for retrieving similar user's information need. Some of these similarity functions are described in (McGill, 1979). In this paper, we use the Cosine Similarity as the matching function (see eq. 1) proposed by (Torgerson, 1958). According to the study by (Noreault et al., 1980), this function is one of the best similarity measures for making angle comparisons between vectors.

$$Cosine\_Similarity(D,Q) = \frac{\Sigma_{i=1}^{n} W_{id} \, . \, W_{iq}}{\sqrt{\Sigma_{i=1}^{n} W_{id}^2 \, . \, \Sigma_{i=1}^{n} W_{iq}^2}} \tag{1}$$

In eq.(1) above, $Cosine\_Similarity(D,Q)$ is the cosine similarity between the query and document vectors, n is the number of index terms that exist in the document D and query Q, W_{id} is the weight of term i in a document D and W_{iq} is the weight of the same term i in query Q.

Most textual IR systems use keywords to retrieve documents. These systems first extract keywords from documents to act as index terms and then assign weights to each index term using various approaches. Such systems have two major difficulties. One is how to choose the appropriate keywords to act as index terms precisely. The other is how to assign the appropriate weights for each index term to represent precisely the information content or the importance of the index term in each document in the document collection.

2.2 IR System Evaluation

For IR system evaluation, we use the system effectiveness and system performance (Baeza-Yates and Ribeiro-Neto, 1999) to measure the impact of the stop-words removal and the discriminative approach on the IR system. The performance measurement used here is the ratio of reduction in the index files of each case study. While the effectiveness function used is the average precision (AvgP) (Chang and Hsu., 1999; Kwok, 1997) and Mean Average Precision (MAP) (Baeza-Yates and Ribeiro-Neto, 2011).

Let $d_1, d_2, ..., d_{|D|}$ denote the sorted documents by decreasing order of their similarity measure function value, where $|D|$ represents the number of testing documents. The function $r(d_i)$ gives the relevance value of a document d_i. It returns 1 if d_i is relevant, and 0 otherwise. The average precision per query (AvgP(q)) is defined as follows:

$$AvgP(q) = \frac{1}{|D|} \, \Sigma_{i=1}^{|D|} \, r(d_i) \, . \, \Sigma_{i=1}^{|D|} \, \frac{1}{j} \tag{2}$$

Where $r(d_i)$ returns 1 if d_i is relevant and 0 otherwise, and $|D|$ represents the number of documents. The mean average precision (MAP) for a set of queries is the mean of the average precision values over all queries. This can be given by the following equation:

$$MAP = \frac{\Sigma_{q=1}^{Q} \, AvgP(q)}{Q} \tag{3}$$

Where Q is the number of queries.

3 A New Term-Weighting Scheme: TF-ATO

In an earlier version of this paper we outlined a new TWS and discriminative approach for static and dynamic document collections called Term-Frequency With Average Term Occurrences (TF-ATO) (Ibrahim and Landa-Silva, 2014). Now in this paper, we describe and discuss our proposed approach in more detail plus conduct a comprehensive study on its performance.

3.1 Related Work on TWS

3.1.1 Traditional TWS

In general, term-weighting schemes (TWS) can be classified into non-learning, supervised learning and unsupervised learning approaches (Greengrass, 2000; Jin et al., 2005; kwang Song and Myaeng, 2012). From the literature on non-learning statistical TWS, we found that most of the TWS proposed by researchers are a variation of the TF-IDF weighting scheme (Reed et al., 2006; Salton and Buckley, 1988; Sparck Jones, 1988). These weighting function combinations were tested in various IR test collections. The equations used for each of these TWS are as follows:

1) Basic TF-IDF TWS (Reed et al., 2006; Salton and Buckley, 1988):

$$W_{ij} \; = \; tf_{ij} \, . \, log(\frac{N}{n_i}) \tag{4}$$

Where W_{ij} is the weight of term i in document j and tf_{ij} is the number of occurrences of term i in document j. N is the number of documents in the document collection and n_i is the number of documents that contain term i in this document collection. From this equation, $IDF_i = log(N/n_i)$. This weighting function has been used widely in the literature because its capability in IR effectiveness compared to other weighting functions. Here we use this weighting scheme to evaluate our TF-ATO method and discriminative approach. One of the reasons for choosing this weighting function is because of its suitability for assessing the IR effectiveness capability on the Ohsumed collection compared to other weighting function as discussed by (Hersh et al., 1994).

2) Augmented maximum term normalization-IDF (ATC) (Jin et al., 2001; Salton and Buckley, 1988):

$$W_{ij} = \frac{\left(0.5 \; + \; 0.5 \, . \, \frac{tf_{ij}}{max_{tf_j}}\right) \, . \, log(\frac{N}{n_i})}{\sqrt{\Sigma_{i=1}^{m} \left[\left(0.5 \; + \; 0.5 \, . \, \frac{tf_{ij}}{max_{tf_j}}\right) . \, log(\frac{N}{n_i})\right]^2}} \tag{5}$$

Where m is the number of terms in the documents space and max_{tf_j} is the maximum term frequency in document j (i.e., the term frequency for the highest term repeated in document j). This weighting function did not give a better IR effectiveness than TF-IDF for the Ohsumed collection as demonstrated by (Hersh et al., 1994).

3) Okapi TWS(Jin et al., 2001):

$$W_{ij} = \left(\frac{tf_{ij}}{0.5 + 1.5 \, . \, \frac{dl_j}{avg_{dl}} + tf_{ij}}\right) . \, log\left(\frac{N - n_j + 0.5}{tf_{ij} + 0.5}\right) \tag{6}$$

Where dl_j is the length of document j (i.e., the summation of all terms frequencies in document j) and avg_{dl} is the average document length of the document collection. The limitation of this TWS is that if an index term occurs in over half the documents in the collection, then this TWS gives a negative term weight (Manning et al., 2008), which cannot represent the information content of that term. Furthermore, the original equation of the Okapi TWS is a probabilistic function that depends in its constants on the relevant documents for queries (Robertson et al., 1995). Since real and test collections are usually partially judged, the majority of documents in the collection are not relevant for any query in the query set.

4) Pivoted document length normalization-IDF (LTU) (Jin et al., 2001):

$$W_{ij} = \left(\frac{1 + log(tf_{ij})}{0.8 + 0.2 \, . \, \frac{dl_j}{avg_{dl}}}\right) . \, log\left(\frac{N}{n_i}\right) \tag{7}$$

This weighting scheme has an advantage in Optical Character Recognition (OCR) and longer document collections (Singhal et al., 1996). However, LTU has not shown advantage for better IR effectiveness (compared to TF-IDF) on the Ohsumed collection where all documents are short.

Another limitation of existing TWS is discussed next. The above and other TWS in the literature (Reed et al., 2006; Greengrass, 2000; McGill, 1979) use some of the document collection characteristics, such as the total numbers of documents in the collection and the document term frequency (number of documents in the document collection that contain this term). In real-world IR systems, these characteristics should be considered as changing over time because nowadays document collections are mostly dynamic instead of static. (Reed et al., 2006) studied the effect on IR effectiveness caused by TF-IDF and its variations in dynamic document collections. The above TWS are TF-IDF variations and have shown no advantage compared to TF-IDF in representing the information content of the test collections when using the cosine similarity measure (Reed et al., 2006). In the present paper we also evaluate the performance of TF-IDF and our proposed TF-ATO approach on dynamic variations on the Ohsumed collection.

3.1.2 Limitation of Evolved TWS and Term Weights

We now discuss the motivation for having non-learning IR approaches instead of learning ones such as evolved TWS. Evolutionary computation approaches have been applied for evolving term weights or evolving a TWS like in (Cordan et al., 2003; Cummins and O'Riordan, 2006). The relevance judgement is the set of queries for the document collection and their corresponding relevant documents from the collection. The objective function of learning IR approaches use relevance judgments to check the quality of the evolved TWS and term weights. However, as mentioned earlier, real and test IR document collections are partially judged as it is not feasible to have fully judged document collections (Qin et al., 2010). Consequently, evolved TWS are limited because the trained queries and their corresponding relevant documents do not cover the whole term space of the collection.

When evolving TWS and term weights, the system should be trained using queries and the corresponding relevant documents containing the whole term space (index terms) that exists in the collection. Then, the IR system should be tested with queries different to those used in the learning process. To the best of our knowledge, it appears that works applying evolutionary computation to IR systems use the same queries from the learning stage to then test the candidate solution that represents the documents. Also, index terms that do not exist in relevant documents are given random weights. Hence, these index terms cannot be judged by the fitness function because they do not exist in relevant documents nor the query set. In some large document collections, the majority of documents that exist in the relevance judgement file are non-relevant for any corresponding query. Hence, the number of random weights created in the evolutionary learning process are not really applicable to measure the relevancy for any query.

The problem with evolving TWS and term weights described above is likely to arise in any large document collection created by pooling technique. Table 1 lists the nine document collections (Hersh et al., 1994; UniversityOfGlasgow; Smucker et al., 2012) used in our analysis in this paper and that have also been used in some works evolving TWS and term weights. Each document collection has three main components: a set of documents, a set of queries and the relevance judgment file. The creation of these collections and their relevance judgements has been done using different approaches including sampling, extracting from real IR system and pooling (Soboroff, 2007; Hersh et al., 1994). A number of additional characteristics about the document collection should be taken into consideration in evolved TWS. Table 2 gives the values for such additional characteristics which are defined as follows.

Table 1 Document Collections General Basic Characteristics

ID	Description	No. of Docs.	No. of Queries
Cranfield	Aeronautical engineering abstracts	1,400	225
Ohsumed	Clinically-Oriented MEDLINE subset	348,566	105
NPL	Electrical Engineering abstracts	11,429	93
CACM	Computer Science ACM abstracts	3,200	52
CISI	Information Science abstracts	1,460	76
Medline	Biomedicine abstracts	1,033	30
FBIS	Foreign Broadcast Information Service	130,471	172
LATIMES	Los Angeles Times	131,896	230
FT	Financial Times Limited	210,158	230

NoUR is the number of unique occurrences of relevant documents that exist in the document collection.

NoDR is the number of duplicates occurrences of relevant documents between queries in the query set.

NoInD is the total number of index terms that exist in the whole document collection.

NoInDr is the number of index terms that exist in the relevant documents set.

NoInR is the number of index terms that were not covered by relevance judgement and is given by the difference NoInD − NoInDr. This is the number of index terms that get a random weights in documents representations without testing them with the objective function.

We can see in Table 2 that in those collections created with a pooling technique, such as FT, FBIS and LATIMES collections, the majority of documents in the relevance judgement are non-relevant for any corresponding query. As we discussed above, this is an issue for evolved TWS because the trained queries and their corresponding relevant documents do not cover the whole term space of the collection. Hence we argue for having non-learning IR approaches instead of learning ones.

3.2 TF-ATO TWS

How to assign appropriate weights to terms is one of the critical issues in automatic term-weighting schemes.

Table 2 Limitation in Document Collections Characteristics for Metaheuristic Techniques

ID	NoUR	NoDR	NoInD	NoInDr	NoInR
Cranfield	924	914	5,222	4,236	986
Ohsumed	4,660	177	227,616	22,760	204,856
NPL	1,735	348	7,697	3,536	4161
CACM	555	241	7,154	3,189	3,965
CISI	1,162	1,952	6,643	5,709	934
Medline	696	0	8,702	6,907	1,795
FBIS	4,506	42,873	177,065	41,272	135,793
LATIMES	4,683	497	211,909	56,255	155,654
FT	5,658	55,819	287,876	45,564	242,312

Given the issues with TF-IDF and evolving approaches discussed above, we then propose a new TWS called *Term Frequency Average Term Occurrences* (TF-ATO) and is expressed by:

$$W_{ij} = \frac{tf_{ij}}{\#\ ATO\ in\ document\ j} \tag{8}$$

and

$$\#\ ATO\ in\ document\ j = \frac{\Sigma_{i=1}^{m_j} tf_{ij}}{m_j} \tag{9}$$

Where, tf_{ij} is the term frequency of term i in document j, ATO is the average term occurrences of terms in the document and is computed for each document, m_j represents the number of unique terms in the document j or in other words it is the number of index terms that exist in document j.

While in the TF-IDF scheme and its variations the global part of the term weight depends on the document collection characteristics, the proposed TF-ATO scheme considers that global weights are the same in any term weight that has a value of 1 for any existing term in the collection. The discrimination approach incorporated into TF-ATO uses the documents centroid as a threshold to remove less-significant weights from the documents.

3.3 Discriminative Approach (DA)

This proposed discriminative approach is a non-learning heuristic approach for improving documents representation. To the best of our knowledge, this is the first *non-learning discriminative approach* for improving documents representation. It is similar to the heuristic method Ide dec-hi (Salton and Buckley, 1997) for improving queries representation. However, our discriminative approach is for documents representation instead of queries, it does not require any relevance judgements information and it depends only on document collection representations. This discriminative approach can be represented by:

$$W_{ij} = \begin{cases} W_{ij} & \text{if } c_i < W_{ij} \\ 0 & \text{if } c_i \geq W_{ij} \end{cases}$$

Where, c_i is the weight of term i in the documents centroid vector and W_{ij} is the term weight of term i in document j. This discriminative approach is applied to every term weight W_{ij} in every document in the collection. The documents centroid vector is given by:

$$C = (c_1, c_2, ..., c_i) \tag{10}$$

and

$$c_i = \frac{1}{N} \Sigma_{i=1}^{N}\ W_{ij} \tag{11}$$

Where N is the number of documents in the collection, c_i is the weight of term i in the centroid vector and W_{ij} is the term weight of term i in document j.

This proposed discriminative approach is somehow based on Luhn's approach (cuts-off) (Luhn, 1957) (see Figure 4) for removing non-significant words from text. However, we take into account that some *non-significant* words can become *significant* in different context according to some documents domains (Saif and Alani, 2014). Thus, we use our discriminative approach to remove non-significant term weights when they are non-significant compared to the centroid of the term weights, instead of removing the terms totally from the documents representations.

3.4 Implementation and Experimental Study

3.4.1 Building the IR System

Information Retrieval systems manage their data resources (document collection) by processing words to extract and assign a descriptive content that is represented as index terms to documents or queries. In text documents, words are formulated with many morphological variants, even if they refer to the same concept. Therefore, the documents often undergo a preprocessing procedure before building the IR system model. The model here is based on the vector space model (VSM) as explained in section 2.1.2. The following procedures are applied to each document in our IR system:

1. Lexical analysis and tokenization of text with the objective of treating punctuation, digits and the case of letters.
2. Elimination of stop-words with the objective of filtering out words that have very low discrimination value for matching and retrieval purposes.
3. Stemming of the remaining words using Porter stemmer (Sparck Jones and Willett, 1997) with the objective of removing affixes (prefixes and suffixes) and allowing the retrieval of documents containing syntactic variations of query terms.
4. Index terms selection by determining which words or stems will be used as index terms.
5. Assign weights to each index term in each document using one given weighting scheme which gives the importance of that index term to a given document.
6. Create documents vectors of term weights in the document collection space (create inverted and directed files using term weights for documents from the document collection).
7. Apply the previous steps (1-6) to queries in order to build queries vectors.
8. For our proposed weighting scheme only (TF-ATO), there are two additional steps:
   - Compute the documents centroid vector from documents vectors by using equations (9) and (10).
   - Use the documents centroid for normalizing documents vectors. This can be done by removing small non-discriminative weights using the documents centroid as a threshold.
9. Matching between documents vectors and each query using cosine similarity and retrieving corresponding documents under fixed 9-points recall values.
10. Rank the retrieved documents according to their cosine similarity measures in descending order and then get the top-10, top-15 and top-30 documents.
11. Compute precision values for the top-10, top-15 and top-30 retrieved documents for each corresponding recall value for each query.
12. Compute the average precision values for the query set in 9-points recall values for the top-10, top-15 and top-30 retrieved documents. Then compute the Mean Average Precision (MAP) value.
13. Repeat steps 5 to 12 for each weighting scheme tested and compare results.

The above procedure has been used for experiments with static data stream. For the case of dynamic data stream, there are two approaches. The first one is to re-compute terms weights for each document in the collection by conducting the above procedure for each update to the collection using a non-learning approach. This of course, adds extra computation cost for every data update in a dynamic data stream. The second approach involves using IDF or the documents centroid in the next approach that is measured from the initial document collection. Then assign term weights to the new documents using the term frequency in the document multiplied by the corresponding IDF for the term that computes by the initial document collection or alternatively, use the discriminative approach. Also, for the term-weighting approach proposed here, the old documents centroid vector is used for eliminating non-discriminative term weights from the added documents. The second approach costs less in computation time but there is less effectiveness in both the proposed TF-ATO and TF-IDF. The cause of this drawback is the variation between the actual values of IDF or documents centroid in dynamic document collection compared with the old values that are computed for the initial collection. Most of the proposed term-weighting schemes have drawbacks in their effectiveness if they do not re-compute their weighting scheme after every major update to the collection. However, this issue has not been mentioned explicitly in previous work and this represents a drawback in the IR system effectiveness when considering dynamic data streams as well as static ones. The cost in effectiveness due to this issue has not been investigated in the published literature to the best of our knowledge.

3.4.2 Experimental Results and Analysis

We conducted two experiments using the overall procedure described in section 3.4.1. The purpose of the first experiment was to compare the average recall precision values achieved by the proposed TF-ATO with and without the discriminative approach to the ones achieved by TF-IDF. Also, this experiment considered the document collection as static. For this first experiment we used two document collections, Ohsumed and CISI (outlined in Table 1) and the Ohsumed query set.

Tables 3, 4 and 5 present the results from the first experiment. Each table shows the results for one case of top-k (where k equals to 10 or 15 or 30) retrieved documents. The tables show the average precision value obtained by each TWS method for nine recall values as well as the corresponding mean average precision (MAP) value.

Table 3 shows results for the case of retrieving the top-10 documents. We can observe that the proposed weighting scheme TF-ATO gives high effectiveness compared to TF-IDF. We can see from the table that TF-ATO without the discriminative approach does not achieve better precision values than TF-IDF for all recall values. But when the discriminative approach is used then TF-ATO always outperforms TF-IDF. Considering all the recall values, the average improvement in precision (given by the MAP value) achieved by TF-ATO without discriminative approach is 6.94% while the improvement achieved by TF-ATO using the discriminative approach is 41%.

The same observations as above can be made for the cases of retrieving the top-15 and the top-30 documents (results in Tables 4 and 5 respectively). That is, using the discriminative approach gives TF-ATO the ability to achieve better effectiveness for all recall values tested. But without the discriminative approach, TF-ATO is overall better than TF-IDF but not always. In Table 4 the average improvement in precision (given by the MAP value) achieved by TF-ATO without discriminative approach is 6.14% while the improvement achieved by TF-ATO using the discriminative approach is 40.07%. In Table 5 the average improvement in precision (given by the MAP value) achieved by TF-ATO without discriminative approach is 8.84% while the improvement achieved by TF-ATO using the discriminative approach is 50.70%.

Table 3 Average Recall-Precision using TF-IDF and TF-ATO With and Without The Discriminative Approach For Top-10 Documents Retrieved in Static Document collections

Recall	**AvgP of Top-10 (static dataset)**		
	TF-IDF	**TF-ATO without DA**	**TF-ATO with DA**
0.1	0.694	0.780	0.867
0.2	0.492	0.563	0.692
0.3	0.373	0.412	0.560
0.4	0.269	0.282	0.428
0.5	0.220	0.208	0.357
0.6	0.189	0.164	0.269
0.7	0.139	0.144	0.216
0.8	0.120	0.124	0.158
0.9	0.109	0.110	0.126
MAP	0.289	0.309	0.408

From the results of this first experiment, it is clear that the proposed TF-ATO weighting scheme gives better effectiveness (higher average precision values) when compared to TF-IDF in static document collections. Also, there is an improvement by using the documents centroid as a discriminative approach with the proposed weighting scheme. Moreover, the proposed discriminative approach reduces the size of the documents in the dataset by removing non-discriminative terms and less significant weights for each document. Using the documents centroid gives an average reduction in size of 2.3% from the actual dataset size compared to 0% reduction when using TF-IDF. Further, from Figure 2, we can observe the difference between each weighting scheme in retrieving the top-k documents (where k equals to 10 or 15 or 30). This figure represents the variation in the applied weighting schemes in static document collection.

Table 4 Average Recall-Precision using TF-IDF and TF-ATO With and Without The Discriminative Approach For Top-15 Documents Retrieved in Static Document collections

Recall	**AvgP of Top-15 (Static dataset)**		
	TF-IDF	**TF-ATO without DA**	**TF-ATO with DA**
0.1	0.749	0.831	0.893
0.2	0.444	0.481	0.615
0.3	0.339	0.367	0.525
0.4	0.248	0.236	0.381
0.5	0.199	0.199	0.323
0.6	0.152	0.156	0.241
0.7	0.136	0.144	0.214
0.8	0.117	0.120	0.159
0.9	0.111	0.113	0.140
MAP	0.277	0.294	0.388

Table 5 Average Recall-Precision using TF-IDF and TF-ATO With and Without The Discriminative Approach For Top-30 Documents Retrieved in Static Document collections

Recall	**AvgP of Top-30 (Static dataset)**		
	TF-IDF	**TF-ATO without DA**	**TF-ATO with DA**
0.1	0.545	0.614	0.728
0.2	0.332	0.368	0.544
0.3	0.233	0.260	0.409
0.4	0.182	0.197	0.322
0.5	0.156	0.165	0.266
0.6	0.137	0.143	0.210
0.7	0.124	0.132	0.167
0.8	0.116	0.119	0.145
0.9	0.109	0.112	0.124
MAP	0.215	0.234	0.324

The purpose of the second experiment was to compare the average recall precision values achieved by the proposed TF-ATO with the discriminative approach to the ones achieved by TF-IDF but now considering the document collection as dynamic. TF-ATO with-

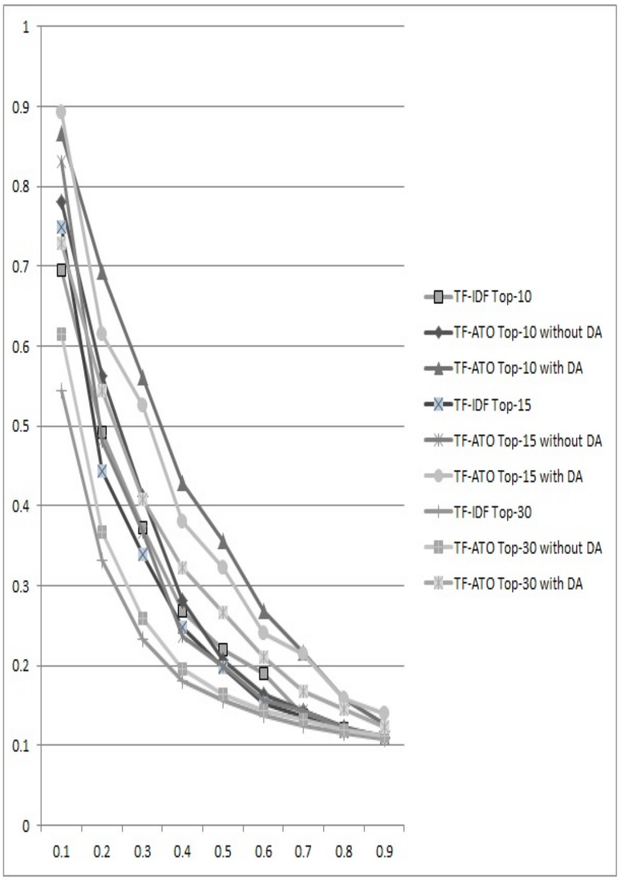

Fig. 2 Graphical Representation of Precision Results From Tables 3, 4 and 5 For Static Document Collections.

out the discriminative approach is not considered here because results from the first experiment showed that TF-ATO performs better using the discriminative approach. For this first experiment we used two document collections, Ohsumed and CISI (outlined in Table 1) and the Ohsumed query set.

In order to conduct this experiment considering the document collection as dynamic, we split the given collection into parts. Then, an initial part of the collection is taken as the initial collection to apply steps 1-8 of the procedure described in section 3.4.1. This allows to compute the index terms IDF values and documents centroid vector weights for the collection. The document collection is then updated by adding the other parts but without updating the index terms IDF values or documents centroid vector weights computed for the initial collection. So, no recalculation is done even after adding a large number (remaining parts) of documents to the initial collection. The reason for this is that recomputing IDF values and assigning new weights (for updating documents in the collection) would have a computational cost of $O(N^2 MLogM)$, where N is the number of documents in the collection and M is the number of terms in the term space (Reed et al., 2006). So, there would be a cost for updating the system in both TF-IDF and TF-ATO approaches but there is no extra cost for using the proposed term weighting scheme without normalization.

In order to determine the ratio for splitting the document collection into parts, we conducted some preliminary experiments. We split the collection into 2, 5, 10 and 30 parts and observed that if the ratio was small (few parts), the variation in MAP values was small and less significant. That is, the simulated effect of having a dynamic data stream was better achieved by splitting the collection into a larger number of parts. Thus, we for this second experiment we split the collection into 30 parts, i.e. the ratio between the initial collection and the final updated collection was 1:30.

Tables 6 and 7 present the results from the second experiment. Each table shows the results for three cases of top-k (where k equals to 10 or 15 or 30) retrieved documents using TF-IDF or TF-ATO. The tables show the average precision value obtained by the given TWS method for nine recall values as well as the corresponding mean average precision (MAP) value.

From these Tables we observe that there is a reduction in effectiveness compared to the case with static data streams. However, the proposed weighting scheme TF-ATO still gives better effectiveness values than those produced with the TF-IDF weighting scheme. We can also see from these Tables that the average improvement in precision of TF-ATO compared to TF-IDF is 42.38% when retrieving the top-10 documents. The improvement is 34.93% when retrieving the top-15 documents and 23.71% when retrieving the top-30 documents. Further, from Figure 3, we can observe the difference between each weighting scheme in retrieving the top-k documents where k equals to 10 or 15 or 30. This figure represents the variation in the two applied weighting schemes (TF-IDF and TF-ATO with discriminative approach) in the case of a dynamic document collection.

4 Stop-words Removal and DA Case Studies

We further investigate the performance of the proposed term-weighting scheme TF-ATO in terms of its discriminative approach and the impact of stop-word removal. For this, we first review related work and then conduct experiments to compare the effectiveness of TF-ATO and TF-IDF in respect to the issues mentioned.

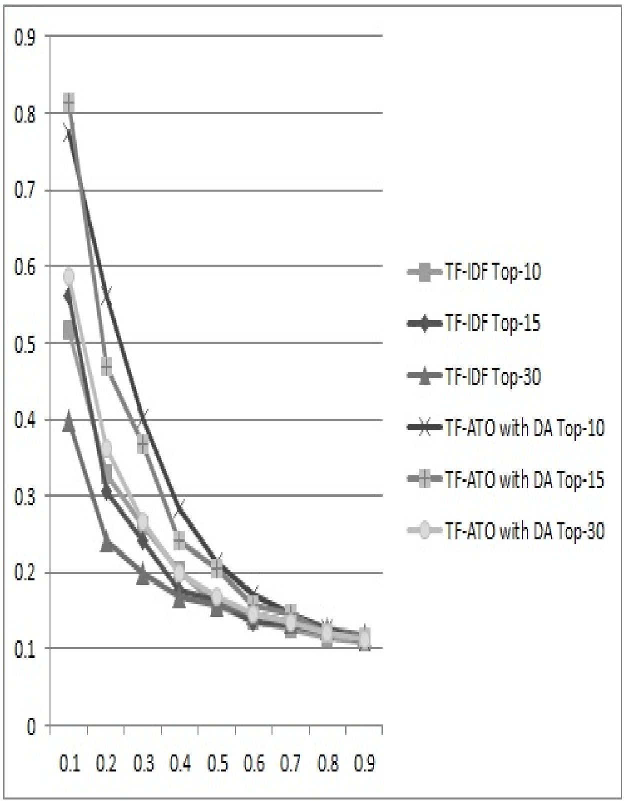

Fig. 3 Graphical Representation of Precision Results From Tables 6 and 7 in Dynamic Document Collections.

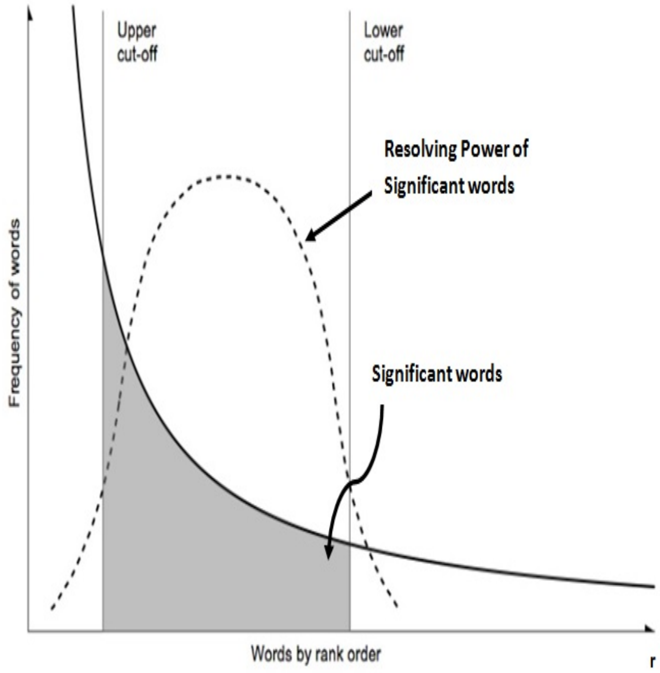

Fig. 4 Zipf's Relationship Frequency vs. Rank Order for Words and Luhn's Cut-off Points for Significant and Non-significant Words on Text.

Table 6 Average Recall-Precision Using TF-IDF For Top-10, Top-15 and Top-30 Documents Retrieved in Dynamic Document Collections

Recall	**AvgP of TF*IDF for dynamic dataset**		
	Precision top-10	**Precision top-15**	**Precision top-30**
0.1	0.516	0.560	0.4
0.2	0.329	0.307	0.242
0.3	0.260	0.242	0.20
0.4	0.202	0.177	0.169
0.5	0.159	0.162	0.157
0.6	0.138	0.136	0.146
0.7	0.126	0.131	0.136
0.8	0.117	0.121	0.127
0.9	0.111	0.116	0.117
MAP	0.217	0.217	0.188

Table 7 Average Recall-Precision Using TF-ATO With Discriminative Approach for Top-10, Top-15 and Top-30 Documents Retrieved in Dynamic Document Collections

Recall	**AvgP of TF-ATO for dynamic dataset**		
	Precision top-10	**Precision top-15**	**Precision top-30**
0.1	0.776	0.813	0.585
0.2	0.561	0.467	0.362
0.3	0.402	0.369	0.266
0.4	0.283	0.241	0.2
0.5	0.213	0.205	0.169
0.6	0.170	0.158	0.145
0.7	0.146	0.146	0.135
0.8	0.125	0.121	0.121
0.9	0.110	0.114	0.112
MAP	0.309	0.293	0.233

4.1 Related Work on Stop-word Lists

Zipf's Law and Luhn's Hypothesis

Zipf states that the relation between the frequency of the use of words and their corresponding rank order is approximately constant (Zipf, 1949). Zipf based his study on American English Newspapers. Based on Zipf's law, Luhn suggested that words used in texts can be divided into significant and non-significant keywords. He specified upper and lower cut-off points on Zipf's curve as shown in Figure 4. The words below the lower cut-off point are rare words that do not contribute significantly to the content of articles. The words above the upper cut-off point occur most frequently and cannot be good discriminators between articles because they are too common in texts. From Zipf's and Luhn's works, researchers have proposed lists of stop-words that should be removed from texts for better effectiveness and accuracy in natural language processing (NLP). From the literature, stop-words lists (stoplists) can be divided into three categories as follows.

1. ***General Stoplists***: These general purpose stoplists are generated from large corpus of text using term ranking scheme and high Document Frequency (high-DF) filtering among other methods inspired by Zipf's law. Examples are the Rijsbergen (Van Rijsbergen, 1975), SMART's (SMART) and Brown's (Fox, 1992) stoplists. Later, (Sinka and Corne, 2003a) generated two ranked list of words in ascending order of their entropy and constructed modern stoplists based on Zipf's and Luhn's work. They showed that their stoplists outperform Rijsbergen's and Brown's stoplists in text clustering problem with respect to accuracy. However, Rijsbergen's and Brown's stoplists perform better on other case studies. Sinka and Corne did not make their stoplists available. It should be noted that the computational cost to build new stoplists from large corpus by this method is high compared to the slight improvement in accuracy.
2. ***Collection-Based Stoplists***: These stoplists are generated from the document collection and can be applied on the test and real IR document collections. The challenge here is in choosing the cut-off points to classify the words in the collection into stop-words, rare (non-significant) words and significant. Four approaches based on Zipf's law and Luhn's principle for choosing corpus-based stop-words list were proposed by (Lo et al., 2005). Further, they used Kullback-Leibler (KL) divergence measure (Cover and Thomas, 1991) to determine the cut-off on these approaches. Their study concluded that the approach using normalized inverse document frequency (IDF) gave better results. It should be noted that the computational cost to build stoplists for each document collection is high compared to generating general stoplists.
3. ***Evolving Stoplists***: In this category, meta-heuristic techniques are used for evolving a group of general stoplists with the aim of producing better stoplists. To the best of our knowledge, only Sinka and Corne (Sinka and Corne, 2003b) have used this approach. Their method starts by combining the top 500 stop-words in the stoplists of (Sinka and Corne, 2003a) with the stoplists of Rijsbergen's and Brown's into one group to be evolved. Then, they applied Hill Climbing (HC) and Evolutionary Algorithm (EA) with 2000 documents in 2-mean clustering problem. In our opinion, the computational cost involved in preparing the documents before applying HC and EA is too high.

Thus, in our opinion, the best option at present for researchers is to use general stoplists which can be generated with less computational cost, are widely available and are easy to apply.

4.2 Experimental Results and Analysis

In these experiments, we investigate the impact of our discriminative approach as a heuristic method for improving documents representations. For this we measure the system effectiveness in terms of the Mean Average Precision (MAP) and the size of the index file. In order to apply the discriminative approach no information about relevance judgement is needed. In these experiments we also examine the impact of stop-words removal. As discussed above, this is an important process for improving the performance and effectiveness of IR systems. Then we investigate the impact of the discriminative approach and the removal of stop-words on two TWS, our proposed TF-ATO and also TF-IDF. We conducted the experiments using the following five document collections: Ohsumed, Cranfield, CISI, FBIS and LATIMES (see Table 1). We excluded the very large collections from these experiments because of the difficulty in processing them on a personal computer but also because the above five collections are commonly used by researchers (Smucker et al., 2012; Voorhees, 2004).

The following four case studies are used in the experiments where TWS is either our TF-ATO or TF-IDF:

- Case 1: apply TWS without using stop-words removal nor discriminative approach.
- Case 2: apply TWS using stop-words removal but without discriminative approach.
- Case 3: apply TWS without using stop-words removal but using discriminative approach.
- Case 4: apply TWS using both stop-words removal and discriminative approach.

Detailed results from our experiments are shown in Tables 10, 11, 12, 13 and 14 in the Appendix. Each table reports for one document collection, the average recall-precision values obtained with the four case studies as described above. The last row in each of these tables shows the MAP values for TWS on each case study across different recall values. Then, these average values are collated and presented in Table 8.

Table 8 Mean Average Precision (MAP) Results Obtained From Each Case in the Experiments. Using and Not-using Stop-words Removal is Indicated With sw(y) and sw(n) Respectively, Similarly for the Discriminative Approach.

Case No.	TWS	Ohsumed	Cranfield	CISI	FBIS	LATIMES
Case 1: sw(n)/da(n)	TF-IDF	0.2150	0.2752	0.2821	0.2871	0.2685
	TF-ATO	0.1883	0.2327	0.2409	0.2486	0.2203
Case 2: sw(y)/da(n)	TF-IDF	0.2680	0.3001	0.3065	0.3479	0.3399
	TF-ATO	0.2793	0.3547	0.3399	0.3917	0.3499
Case 3: sw(n)/da(y)	TF-IDF	0.2774	0.2818	0.2953	0.2925	0.3056
	TF-ATO	0.2781	0.3014	0.3146	0.2954	0.3124
Case 4: sw(y)/da(y)	TF-IDF	0.3488	0.3556	0.3578	0.3938	0.3861
	TF-ATO	0.3636	0.3998	0.3621	0.4267	0.3953

Several observations can be made from the results in Table 8. First, it is clear that for both TWS in all five collections, using both stop-words removal and the discriminative approach (case 4) gives the better results. When comparing cases 2 and 3 (using only one of stop-word removal or discriminative approach), better results in general are obtained when using stop-words removal (case 2) than when using the discriminative approach (case 3). We note that when comparing TF-ATO and TF-IDF on cases 2, 3 and 4, our proposed TWS produces better results. Specifically, in case 2 (using stop-words removal only) TF-ATO outperforms TF-IDF by 2-18%, in case 3 (using discriminative approach only) TF-ATO outperforms TF-IDF by 0.3-7% and in case 4 (using both) TF-ATO outperforms TF-IDF by 2-12%. We believe this is because the ability of the discriminative approach and stop-words removal to remove more non-significant keywords compared to the traditional IDF method. We recognise however, that TF-IDF outperforms TF-ATO by 14-22% in case 1 (not using stop-words removal nor discriminative approach). This is due to the ability of IDF to remove some non-significant words from the documents by assigning values of 0 to words that are repeated in all documents in the collection.

The stop-words removal and discriminative approach have a large impact on the efficiency of the IR system measured in terms of the index file size. Results for this are presented in Table 9. From this table we can see that when comparing cases 2 and 3 for each TWS on the five document collections, using stop-words removal (case 2) helps to reduce the index file size by 30.61-38.4% of the original index file (case 1). Whereas, the reduction when using discriminative approach only (case 3) is between 0.7-30.22%. Using both stop-words removal and discriminative approach (case 4) reduces the index file size between 32.72-39.8%. The positive effect of stop-words removal and discriminative approach is larger on TF-ATO than on TF-IDF. This is because IDF has already the ability to remove non-significant words.

Table 9 The Ratios (%) Of Reduction Of The Size Of The Index File Obtained From Its Original Index Size For Each Case in the Experiments.

Case Id	TF-IDF	TF-ATO
Ohsumed Case1	0.083%	0%
Ohsumed Case2	30.61%	30.65%
Ohsumed Case3	0.7%	0.75%
Ohsumed Case4	32.72%	32.76%
LATIMES case1	0.006%	0%
LATIMES case2	35.21%	35.22%
LATIMES case3	8.17%	8.16%
LATIMES case4	36.8%	36.78%
FBIS case1	9.12%	0%
FBIS case2	38.27%	33.7%
FBIS case3	30.22%	27.4%
FBIS case4	39.8%	39.6%
Cranfield case1	0.17%	0%
Cranfield case2	33.9%	33.83%
Cranfield case3	9.1%	9.4%
Cranfield case4	34.7%	34.5%
CISI case1	0.19%	0%
CISI case2	38.15%	38.4%
CISI case3	7.9%	7.5%
CISI case4	39%	38.9%

5 Conclusion and Future Work

From the study presented in this paper, we conclude that the proposed *Term Frequency - Average Term Occurrences (TF-ATO)* term-weighting scheme (TWS) can be considered competitive when compared to the widely used TF-IDF. The proposed TWS gives higher effectiveness in both cases of static and dynamic document collections. Also, the document centroid vector can act as a threshold in normalization to discriminate between documents for better effectiveness in retrieving relevant documents. We observed a variation and reduction in system effectiveness when using dynamic instead of static document collections, plus there is additional cost for every update to the collection.

We also showed that both stop-words removal and the discriminative approach have a positive effect on both TWS (TF-IDF and TF-ATO) for improving the IR performance and effectiveness. Also, TF-IDF has a positive impact for removing some non-significant keywords from the test collections compared to TF-ATO. However, using stop-words removal and the discriminative approach have a larger impact on removing non-significant weights and keywords from the collection, more significantly on TF-ATO but also on TF-IDF. This means that it is beneficial to use the proposed discriminative approach as a heuristic method for improving IR effectiveness and performance with no information on the relevance judgement for the collection. Our results showed that in general TF-ATO outperforms TF-IDF in terms of effectiveness. Only when both stop-words removal and discriminative approach are not used, TF-IDF outperforms TF-ATO.

In this paper we also discussed approaches to generate stoplists. We find that using evolutionary computation and meta-heuristics for evolving TWS or term weights has some issues. Real and test document collections have limitations in the relevance judgements information available. This means that test collections are partially judged collections. This can cause that most of the index terms in the collections have random weights. It can also cause that the evolved TWS is fit only for the terms existing in the relevant documents assessed in the relevance judgement. Hence, the evolved TWS and weights are practically random for other documents in the collections not assessed in the relevance judgement.

We propose the following future work. Given the limitations of evolving TWS on partially judged collections, we intend to use Genetic Programming (GP) for evolving TWS on fully judged document collections containing approximately 30,350 index terms, 9,732 documents and 581 queries where each document is relevant for at least one query. Furthermore, we intend to develop a *Hybrid Metaheuristic TWS* approach for evolving term weights seeking to address the issues identified in the present study.

References

Ricardo A. Baeza-Yates and Berthier Ribeiro-Neto. *Modern Information Retrieval.* Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1999.

Ricardo A. Baeza-Yates and Berthier A. Ribeiro-Neto. *Modern Information Retrieval - the concepts and technology behind search, Second edition.* Pearson Education Ltd., Harlow, England, 2nd editio edition, 2011.

C. H. Chang and C. C. Hsu. *The design of an information system for hypertext retrieval and automatic discovery on WWW.* PhD thesis, National Taiwan University, 1999.

O. Cordan, E. Herrera-Viedma, C. Lapez-Pujalte, M. Luque, and C. Zarco. A review on the application of evolutionary computation to information retrieval. *International Journal of Approximate Reasoning*, 34 (23):241 – 264, 2003. Soft Computing Applications to Intelligent Information Retrieval on the Internet.

Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory.* Wiley-Interscience, New York, NY, USA, 1991.

Ronan Cummins. *The Evolution and Analysis of Term-Weighting Schemes in Information Retrieval.* PhD thesis, National University of Ireland, Galway, 2008.

Ronan Cummins and Colm O'Riordan. Term-weighting in information retrieval using genetic programming: A three stage process. In *Proceedings of the 2006 Conference on ECAI 2006: 17th European Conference on Artificial Intelligence August 29 – September 1, 2006, Riva Del Garda, Italy*, pages 793–794, Amsterdam, The Netherlands, The Netherlands, 2006. IOS Press.

Christopher Fox. Information retrieval. chapter Lexical Analysis and Stoplists, pages 102–130. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1992.

Ed Greengrass. Information Retrieval : A Survey. Technical Report November, University of Maryland, USA, 2000. URL http://www.csee.umbc.edu/csee/research/cadip/readings/IR.report.120600.book.pdf.

William Hersh, Chris Buckley, T. J. Leone, and David Hickam. Ohsumed: An interactive retrieval evaluation and new large test collection for research. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '94, pages 192–201, New York, NY, USA, 1994. Springer-Verlag New York, Inc.

Osman A. S. Ibrahim and Dario Landa-Silva. A new weighting scheme and discriminative approach for information retrieval in static and dynamic document collections. In *Computational Intelligence (UKCI), 2014 14th UK Workshop on*, pages 1–8, Sept 2014.

Rong Jin, Christos Falusos, and Alex G. Hauptmann. Meta-scoring: Automatically evaluating term weighting schemes in ir without precision-recall. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '01, pages 83–89, New York, NY, USA, 2001. ACM.

Rong Jin, Joyce Y. Chai, and Luo Si. Learn to weight terms in information retrieval using category information. In *Proceedings of the 22Nd International Conference on Machine Learning*, ICML '05, pages 353–360, New York, NY, USA, 2005. ACM.

Thorsten Joachims. Text categorization with support vector machines: Learning with many relevant features. In Claire Ndellec and Cline Rouveirol, editors, *Machine Learning: ECML-98*, volume 1398 of *Lecture Notes in Computer Science*, pages 137–142. Springer Berlin Heidelberg, 1998.

Marika Kaden, Martin Riedel, Wieland Hermann, and Thomas Villmann. Border-sensitive learning in generalized learning vector quantization: an alternative to support vector machines. *Soft Computing*, pages 1–12, 2014. doi: 10.1007/s00500-014-1496-1.

Sa kwang Song and Sung Hyon Myaeng. A novel term weighting scheme based on discrimination power obtained from past retrieval results. *Information Processing & Management*, 48(5):919 – 930, 2012. Large-Scale and Distributed Systems for Information Retrieval.

K. L. Kwok. Comparing representations in Chinese information retrieval. In *SIGIR '97 Proceedings of the 20th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 34–41, New York, NY, USA, 1997. ACM.

Lemur. URL http://www.lemurproject.org/.

Tie-Yan Liu. Learning to rank for information retrieval. *Foundations and Trends in Information Retrieval*, 3 (3):225–331, 2009.

Rachel Tsz-wai Lo, Ben He, and Iadh Ounis. Automatically Building a Stopword List for an Information Retrieval System. *Digital Information Management: special issue on the 5th Dutch-Belgian Information Retrieval Workshop (DIR 2005)*, 3(1):3–8, 2005.

H.P. Luhn. A statistical approach to mechanized encoding and searching of literary information. *IBM Journal of Research and Development*, 1(4):309–317, Oct 1957.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008. ISBN 0521865719, 9780521865715.

Michael McCandless, Erik Hatcher, and Otis Gospodnetic. *Lucene in Action, Second Edition: Covers Apache Lucene 3.0*. Manning Publications Co., Greenwich, CT, USA, 2010. ISBN 1933988177, 9781933988177.

Michael McGill. An evaluation of factors affecting document ranking by information retrieval systems. 1979.

Christian Middleton and Ricardo Baeza-yates. A comparison of open source search engines. Technical report, 2007. URL http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.119.6955.

T. Noreault, M. McGill, and M. Koll. A performance evaluation of similarity measures, document term weighting schemes and representations in a Boolean environment. In *SIGIR '80 Proceedings of the 3rd annual ACM conference on Research and development in information retrieval*, pages 57–76. Butterworth & Co. Kent, UK, 1980.

Tao Qin, Tie-Yan Liu, Jun Xu, and Hang Li. Letor: A benchmark collection for research on learning to rank for information retrieval. *Information Retrieval*, 13 (4):346–374, 2010. ISSN 1386-4564.

Joel W. Reed, Yu Jiao, Thomas E. Potok, Brian A. Klump, Mark T. Elmore, and Ali R. Hurson. Tf-icf: A new term weighting scheme for clustering dynamic data streams. In *Proceedings of the 5th International Conference on Machine Learning and Applications*, ICMLA '06, pages 258–263, Washington, DC, USA, 2006. IEEE Computer Society.

S. E. Robertson, S. Walker, M. M. Hancock-Beaulieu, S. Jones, and M. Gatford. Okapi at TREC-3. In D. Harman, editor, *Proceeding of Third Text REtrieval Conference TREC3*, pages 109–126, Gaithersburg, 1995.

Miriam; He Yulan Saif, Hassan; Fernández and Harith Alani. On stopwords, filtering and data sparsity for sentiment analysis of Twitter. In *In: LREC 2014, Ninth International Conference on Language Resources and Evaluation*, pages 810–817, Reykjavik, Iceland, 2014.

Gerard Salton and Chris Buckley. Readings in information retrieval. chapter Improving Retrieval Performance by Relevance Feedback, pages 355–364. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1997.

Gerard Salton and Christopher Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5):513 – 523, 1988.

Gerard Salton and Michael J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, New York, NY, USA, 1986.

Amit Singhal, Chris Buckley, and Mandar Mitra. Pivoted document length normalization. In *Proceedings of the 19th Annual International ACM SIGIR Con-*

ference on Research and Development in Information Retrieval, SIGIR '96, pages 21–29, New York, NY, USA, 1996. ACM.

Mark P Sinka and David W Corne. Towards Modernised and Web-Specific Stoplists for Web Document Analysis. pages 0–6, 2003a.

Mark P Sinka and David W Corne. Evolving Better Stoplists for Document Clustering and Web Intelligence. *Design and application of hybrid intelligent systems*, pages 1015–1023, 2003b.

SMART. SMART System Stop-words List. URL http://jmlr.org/papers/volume5/lewis04a/a11-smart-stop-list/english.stop.

Mark D Smucker, Gabriella Kazai, and Matthew Lease. Overview of the trec 2012 crowdsourcing track. Technical report, DTIC Document, 2012.

Ian Soboroff. A comparison of pooled and sampled relevance judgments. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '07, pages 785–786, New York, NY, USA, 2007. ACM.

Karen Sparck Jones. Document retrieval systems. chapter A Statistical Interpretation of Term Specificity and Its Application in Retrieval, pages 132–142. Taylor Graham Publishing, London, UK, 1988.

Karen Sparck Jones and Peter Willett, editors. *Readings in Information Retrieval*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1997.

Warren S Torgerson. Theory and methods of scaling. 1958.

UniversityOfGlasgow. Test collections. URL http://ir.dcs.gla.ac.uk/resources/test_collections/.

C.J. Van Rijsbergen. *Information Retrieval*. Butterworths, 1975. URL http://www.dcs.gla.ac.uk/Keith/Preface.html.

A. Vinciarelli. Application of information retrieval techniques to single writer documents. *Pattern Recognition Letters*, 26(14):2262–2271, 2005.

Ellen M. Voorhees. Overview of the trec 2004 robust retrieval track. In *In Proceedings of the Thirteenth Text REtrieval Conference (TREC-2004)*, page 13, 2004.

Stephan Winkler, Susanne Schaller, Viktoria Dorfer, Michael Affenzeller, Gerald Petz, and Micha Karpowicz. Data-based prediction of sentiments using heterogeneous model ensembles. *Soft Computing*, pages 1–12, 2014. doi: 10.1007/s00500-014-1325-6.

Ligang Zhou, KinKeung Lai, and Lean Yu. Credit scoring using support vector machines with direct search for parameters selection. *Soft Computing*, 13(2):149–155, 2009.

George K. Zipf. *Human Behavior and the Principle of Least Effort*. Addison-Wesley (Reading MA), 1949.

Appendix – Detailed Experimental Results of Subsection 4.2

The cases studies on these results are as follows:

- Case 1: applying term-weighting scheme without using stop-words removal nor discriminative approach.
- Case 2: applying term-weighting scheme using stop-words removal but without discriminative approach.
- Case 3: applying term-weighting scheme without using stop-words removal but using discriminative approach.
- Case 4: applying term-weighting scheme using both stop-words removal and discriminative approach.

Table 10 Average Recall-Precision Results Obtained on the Ohsumed Collection From Each Case in the Experiments.

Recall	Average Precision In Ohsumed Collection For Cases Studies							
	Case1		Case2		Case3		Case4	
	TF-IDF	TF-ATO	TF-IDF	TF-ATO	TF-IDF	TF-ATO	TF-IDF	TF-ATO
0.1	0.5469	0.5359	0.6627	0.7419	0.6475	0.7132	0.7967	0.8161
0.2	0.3307	0.2669	0.4560	0.4704	0.4448	0.4696	0.5837	0.6104
0.3	0.2463	0.1829	0.3484	0.3461	0.3434	0.3607	0.4416	0.4724
0.4	0.1757	0.1420	0.2327	0.2383	0.2529	0.2590	0.3430	0.3617
0.5	0.1505	0.1269	0.1916	0.1905	0.2162	0.1961	0.2604	0.2883
0.6	0.1328	0.1166	0.1596	0.1542	0.1764	0.1530	0.2413	0.2400
0.7	0.1243	0.1120	0.1330	0.1399	0.1560	0.1297	0.1938	0.1989
0.8	0.1170	0.1074	0.1179	0.1212	0.1362	0.1140	0.1450	0.1541
0.9	0.1110	0.1044	0.1098	0.1113	0.1230	0.1075	0.1335	0.1301
MAP	0.2150	0.1883	0.2680	0.2793	0.2774	0.2781	0.3488	0.3636

Table 11 Average Recall-Precision Results Obtained on the LATIMES Collection From Each Case in the Experiments.

Recall	Average Precision In LATIMES Collection For Cases Studies							
	Case1		Case2		Case3		Case4	
	TF-IDF	TF-ATO	TF-IDF	TF-ATO	TF-IDF	TF-ATO	TF-IDF	TF-ATO
0.1	0.5218	0.4059	0.5701	0.5800	0.5280	0.5629	0.7226	0.7640
0.2	0.4741	0.3373	0.5403	0.5789	0.4314	0.4414	0.6850	0.6580
0.3	0.3515	0.3157	0.4956	0.5034	0.3920	0.3928	0.4321	0.5103
0.4	0.2944	0.2698	0.3871	0.3912	0.3450	0.3483	0.3665	0.4101
0.5	0.2427	0.1816	0.3171	0.3190	0.3050	0.3204	0.3276	0.3292
0.6	0.1834	0.1589	0.2586	0.2637	0.2910	0.2898	0.2890	0.2675
0.7	0.1423	0.1427	0.2027	0.2079	0.1721	0.1721	0.2543	0.2218
0.8	0.1111	0.1064	0.1617	0.1613	0.1581	0.1581	0.2169	0.2008
0.9	0.0953	0.0644	0.1254	0.1436	0.1259	0.1259	0.1813	0.1961
MAP	0.2685	0.2203	0.3399	0.3499	0.3054	0.3124	0.3861	0.3953

Table 12 Average Recall-Precision Results Obtained on the FBIS Collection From Each Case in the Experiments.

Recall	Average Precision In FBIS Collection For Cases Studies							
	Case1		Case2		Case3		Case4	
	TF-IDF	TF-ATO	TF-IDF	TF-ATO	TF-IDF	TF-ATO	TF-IDF	TF-ATO
0.1	0.4867	0.4579	0.5818	0.6011	0.5130	0.5067	0.6226	0.6693
0.2	0.4217	0.4030	0.5010	0.5777	0.4570	0.4914	0.5585	0.6333
0.3	0.3810	0.3096	0.4934	0.5498	0.4182	0.4282	0.5321	0.5992
0.4	0.3430	0.2905	0.4103	0.4283	0.3765	0.3483	0.4066	0.5169
0.5	0.2947	0.2473	0.3722	0.4213	0.2100	0.2204	0.3828	0.4292
0.6	0.2058	0.1954	0.3021	0.3968	0.1950	0.1898	0.3890	0.3902
0.7	0.2069	0.1661	0.2098	0.2710	0.1609	0.1721	0.2543	0.2922
0.8	0.1377	0.1174	0.1501	0.1587	0.1581	0.1581	0.2169	0.1928
0.9	0.1060	0.0500	0.1101	0.1208	0.1436	0.1436	0.1813	0.1174
MAP	0.2871	0.2486	0.3479	0.3917	0.2925	0.2954	0.3938	0.4267

Table 13 Average Recall-Precision Results Obtained on the Cranfield Collection From Each Case in the Experiments.

Recall	Average Precision In Cranfield Collection For Cases Studies							
	Case1		Case2		Case3		Case4	
	TF-IDF	TF-ATO	TF-IDF	TF-ATO	TF-IDF	TF-ATO	TF-IDF	TF-ATO
0.1	0.6426	0.4536	0.6587	0.6982	0.6526	0.6636	0.7287	0.7650
0.2	0.4251	0.4261	0.4852	0.5297	0.4561	0.4638	0.5480	0.6549
0.3	0.3734	0.3429	0.4026	0.4568	0.3619	0.4021	0.4611	0.5262
0.4	0.2916	0.2771	0.3319	0.4019	0.2961	0.3310	0.3619	0.4084
0.5	0.2050	0.2370	0.2682	0.3593	0.2570	0.2898	0.3168	0.3605
0.6	0.1831	0.1283	0.1948	0.2672	0.1533	0.1617	0.2699	0.2925
0.7	0.1544	0.1091	0.1398	0.2050	0.1171	0.1466	0.2292	0.2255
0.8	0.1072	0.0724	0.1121	0.1558	0.1352	0.1327	0.1558	0.1923
0.9	0.0948	0.0483	0.1079	0.1187	0.1068	0.1216	0.1287	0.1727
MAP	0.2752	0.2327	0.3001	0.3547	0.2818	0.3014	0.3556	0.3998

Table 14 Average Recall-Precision Results Obtained on the CISI Collection From Each Case in the Experiments.

Recall	Average Precision In CISI Collection For Cases Studies							
	Case1		Case2		Case3		Case4	
	TF-IDF	TF-ATO	TF-IDF	TF-ATO	TF-IDF	TF-ATO	TF-IDF	TF-ATO
0.1	0.5633	0.4682	0.6429	0.6243	0.5096	0.6030	0.7274	0.7398
0.2	0.4208	0.3546	0.5208	0.5597	0.4600	0.5438	0.6261	0.6213
0.3	0.3953	0.3080	0.4267	0.4568	0.4230	0.4308	0.5257	0.5370
0.4	0.3423	0.2663	0.3193	0.4029	0.3900	0.3602	0.4422	0.4652
0.5	0.2894	0.2346	0.2643	0.3306	0.2034	0.2981	0.3390	0.2633
0.6	0.1827	0.1856	0.2078	0.2672	0.1890	0.1689	0.2446	0.2458
0.7	0.1399	0.1754	0.1400	0.1857	0.1709	0.1466	0.1488	0.1474
0.8	0.1108	0.1076	0.1267	0.1219	0.1632	0.1537	0.1083	0.1230
0.9	0.0947	0.0673	0.1097	0.1098	0.1482	0.1267	0.0579	0.1150
MAP	0.2821	0.2408	0.3065	0.3399	0.2953	0.3146	0.3578	0.3620