NTCIR-6 CLIR-J-J Experiments at Yahoo! Japan

Sumio FUJITA
Yahoo Japan Corporation,
Roppongi Hills Mori Tower, 6-10-1, Roppongi, Minato-ku,
Tokyo 106-6182, Japan
sufujita AT yahoo-corp DOT jp

Abstract

This paper describes NTCIR-6 experiments of the CLIR-J-J task, i.e. Japanese monolingual retrieval subtask, at the Yahoo group, focusing on the parameter optimization in information retrieval (IR). Unlike regression approaches, we optimized parameters completely independent from retrieval models so that the optimized parameter set can illustrate the characteristics of the target test collections. We adopted the genetic algorithm as optimization tools and cross-validated with 4 test collections, namely NTCIR-3,4,5, and 6 CLIR-J-J.

Keywords: *Information retrieval, Test collections, Parameter calibration, Genetic algorithm.*

1. Introduction

The choice of scoring function is crucial for better search effectiveness. In our past NTCIR-4 CLIR-J-J and NTCIR-5 CLIR-J-J experiments, we ended by choosing BM25TF*IDF runs for official submissions. In fact, the choice was found to be good for previous NTCIRs. Our choices of NTCIR-X were based upon the pre-submission experiments using the test collection of NTCIR-(X-1). The scoring functions have some coefficient parameters to be determined during the pre-submission experiments. Failing to well calibrate these parameters results in poor effectiveness in the official evaluation even the scoring function is good enough. Then the calibration of parameters becomes a major work in pre-submission experiments of the NTCIR tasks. Such coefficient parameters make the scoring function adaptable to diverse environments, where requiring the efforts of calibration.

Given available four test collections in the NTCIR-6 CLIR task, we try to evaluate if an automatic calibration does work for such limited number of training data. In order to compare with retrospective runs, we adopted the same system as for the NTCIR-5 experiments.

Optimization of scoring function is studied by several regression approaches to information retrieval [3][6]. We consider the calibration process as a simple optimization problem and we adopted the genetic algorithm (GA) to optimize the parameters to given test collections. The process is challenging because:

1) The optimization process is easily fallen into local maximum points and it failed to find the global maximum.
2) The optimized parameters might be overfitted to the training collection and they do not perform well for other collections.

Adopting the GA, the main concern seems to be the second issue rather than the first one. In this paper, we present our NTCIR-6 CLIR-J-J task experiments, i.e. Japanese monolingual runs, focusing on the possibility of automatic calibration of search parameters.

The rest of the paper is organized as follows: Section 2 describes our experiment environment and retrieval system. Section 3 explains briefly the genetic algorithm and Section 4 reports our official runs and post submission experiments. Section 5 concludes the paper.

2. System Description

Our evaluation environment: YLMS system is implemented based on Lemur toolkit 4.0 for indexing system [12], which is being developed by the Lemur project.

2.1 Indexing language

Chasen version 2.2.9 Japanese morphological analyzer with IPADIC dictionary version 2.5.1 are utilized for Japanese text segmentation and output single words are used as indexing units. Stop word lists for newspaper documentation are prepared.

2.2 BM25TF*IDF

A Retrieval Status Value between a document d and a query q is calculated as a dot product between the document term vector and the query term vector, where each term is weighted by TF*IDF [15]. Okapi BM25 TF [13][14] is used.

$$RSV(q,d) = \sum_{t \in q \cap d} w(q,t) w(d,t)$$

$$w(d,t) = TF(d,t) IDF(t)$$

$$w(q,t) = TF(q,t) IDF(t)$$

$$TF(d,t) = \frac{(k1+1)\,freq(d,t)}{k1((1-b)+b\frac{dl_d}{avdl})+freq(d,t)}$$

$$IDF(t) = (k4 + \log\frac{N}{df(t)})$$

d : document or query

t : term

N : total number of documents in the collection

$df(t)$: number of documents where t appears

$freq(d,t)$: number of occurrences of t in d

dl_d : document length of d

$avdl$: average document length in the collection

2.3 Feedback strategies

The strategy of "feedback from top k documents in a pilot search" is applied. The Rocchio feedback for TF*IDF is adopted as term extraction method, where the term precision measure to select salient terms is calculated as an element of the centroid vector of pseudo-relevant documents. Finally, an updated query vector Q' was computed from the original query vector Q and a set of (pseudo) relevant document vectors R.

$$Q' = Q + posCoeff \cdot \frac{1}{|R|} \cdot \sum_{d \in R} d$$

Instead of using a linear mean to average through feedback document vectors, we also used an alpha average, which is introduced by information geometry society to mixture probability distributions [1]. Alpha averaging is also used by an experimental IR system [8][9] to mixture document feature vectors.

$$Q' = Q + posCoeff \cdot \left(\frac{1}{|R|} \cdot \sum_{d \in R} d^{\frac{1-\alpha}{2}} \right)^{\frac{2}{1-\alpha}}$$

Giving -1 to alpha, this is equivalent to a linear mean.

2.4 Parameter calibration

In our past experiences in NTCIR-4 and 5 [4][5], where we compared BM25TF*IDF with KL-Divergence approach. The BM25TF*IDF scoring method is usually a very good function to achieve better effectiveness against diverse test collections but it is subject to some coefficients to be calibrated based on training collections, namely k1, b, k4. As for k1 and b of the query side TF, these are fixed to 1000 and 0 respectively, reducing the query TF function to a simple count of in-text term occurrences. Feedback effectiveness is also subject to the feedback parameters, namely feedback document number (fbDocCnt), feedback term number (fbTermCnt) and feedback term coefficient (fbPosCoeff).

3. Genetic Algorithm for IR

GA is applied to information retrieval systems mainly on relevance feedback contexts as has been noted by Lopez-Pujalte et al.[11]. Fan et al. proposed to directly learn ranking functions by applying genetic programming, i.e. an extension of GA [2].

3.1 Genetic algorithm at a glance

From the metaphor of the organic reproduction systems, the genetic algorithm is generally applied to optimization problems in diverse domains, considering each trial point in the search space as an individual. Individuals to be examined are generated by applying genetic operations on each chromosome, representative of an individual, on which parameters to generate a particular individual are encoded. Given a population of individuals, genetic operations are applied iteratively in order to produce a new generation of the population. For each generation, each individual is evaluated by the given fitness function and the process terminates when a targeted fitness value is achieved or the predefined number of generations are processed. On top of the traditional genetic operations, namely selection, crossover and mutation, the distributed genetic algorithm adopts the migration operation: a population is divided into several islands and GA is performed in each island, where the migration operation moves a certain number of individuals to another island as described in Hiroyasu et al. [7]. We used their implementation.

Parameter	Range
K1	0 .. 3.0
B	0 .. 1.0
K4	0 .. 3.0
fbDocCnt (Integer)	0 .. 31
fbTermCnt (Integer)	0 .. 255
fbPosCoeff	0 .. 2.0

Table 1: Parameter range of GA search space

3.2 GA process and operations

Unlike many applications of GA to IR in literature, we do not encode term vectors directly to chromosomes. Instead, we encode the parameters of our retrieval model, namely k1, b, k4, fbDocNum, fbTermNum, fbPosCoeff. These integer or real numbers are encoded on 45 bits of Boolean strings in the ranges shown in Table 1. We carried out GA optimization on a 8-node cluster of Xeon 3.00GHz Dual CPU machines running a Free BSD operating system. The process is divided into 8 islands and each island contains 10 individuals. Initial

	MAP-Rigid	RP-Rigid	Rel-Ret Rigid	P@10 Rigid	P@20 Rigid	MAP-Relax	RP-Relax	Rel-Ret Relax	P@10 Relax	P@20 Relax
YLMS-J-J-T-01	0.3182	0.3240	2753	0.4260	0.3820	0.3898	0.3929	4043	0.5940	0.5250
YLMS-J-J-T-01-N3	0.3733	0.3552	1442	0.4476	0.3893	0.4362	0.4158	2195	0.5714	0.5238
YLMS-J-J-T-01-N4	0.3905	0.3970	5896	0.5964	0.5491	0.4843	0.4848	9316	0.7291	0.7000
YLMS-J-J-T-01-N5	0.4464	0.4480	1968	0.5426	0.4383	0.5259	0.5067	3859	0.7000	0.6298

Table 2: Effectiveness of CLIR-J-J 1st stage, the official title only run and corresponding 2nd stage runs

	MAP-Rigid	RP-Rigid	Rel-Ret Rigid	P@10 Rigid	P@20 Rigid	MAP-Relax	RP-Relax	Rel-Ret Relax	P@10 Relax	P@20 Relax
YLMS-J-J-D-02	0.2719	0.2797	2505	0.3460	0.3250	0.3480	0.3577	3744	0.4900	0.4690
YLMS-J-J-D-02-N3	0.3725	0.3659	1654	0.4643	0.3952	0.4285	0.4052	2231	0.5833	0.5143
YLMS-J-J-D-02-N4	0.3747	0.3967	7137	0.5618	0.5309	0.4719	0.4865	9215	0.7182	0.6836
YLMS-J-J-D-02-N5	0.3983	0.3898	2112	0.4787	0.4106	0.4961	0.4874	3847	0.6468	0.5883

Table 3: Effectiveness of CLIR-J-J 1st stage, the first official description only run and corresponding 2nd stage runs

	MAP-Rigid	RP-Rigid	Rel-Ret Rigid	P@10 Rigid	P@20 Rigid	MAP-Relax	RP-Relax	Rel-Ret Relax	P@10 Relax	P@20 Relax
YLMS-J-J-D-03	0.2743	0.2813	2513	0.3780	0.3290	0.3615	0.3700	3748	0.5580	0.4860
YLMS-J-J-D-03-N3	0.3276	0.3347	1456	0.3857	0.3631	0.3827	0.3821	2171	0.5000	0.4655
YLMS-J-J-D-03-N4	0.3649	0.3833	5379	0.5436	0.5100	0.4602	0.4756	8556	0.6818	0.6527
YLMS-J-J-D-03-N5	0.3863	0.3820	1935	0.4511	0.4074	0.4639	0.4518	3743	0.6000	0.5670

Table 4: Effectiveness of CLIR-J-J 1st stage, the second official description only run (word + char-bigram fusion) and corresponding 2nd stage runs

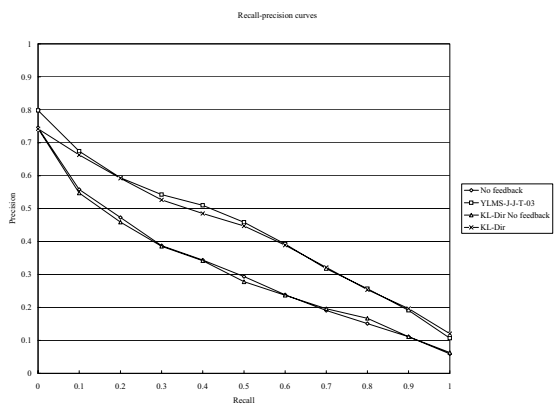
	K1	b	K4	#fb Docs	#fb Terms	Fb pos coeff	FB Alpha	Indexing	Fusion
YLMS-J-J-T-01	1.1	0.4	1.5	9	70	0.8	-1.2	Words	
YLMS-J-J-D-02	1.4	0.4	1.0	6	80	0.7	-1.0	Words	
YLMS-J-J-D-03 bigram	1.2	0.2	1.0	6	80	0.7	-2.1	Bigrams/ Words	0.5:0.5

Table 5: Parameters of CLIR-J-J official runs

populations are randomly generated and the following operations are sequentially applied in each island.

3.2.1 Migration

The migration operation moves randomly chosen individuals from an island to another island. This operation continues from the island to the next island and finally returns to the first island so that the population in each island remains the same.



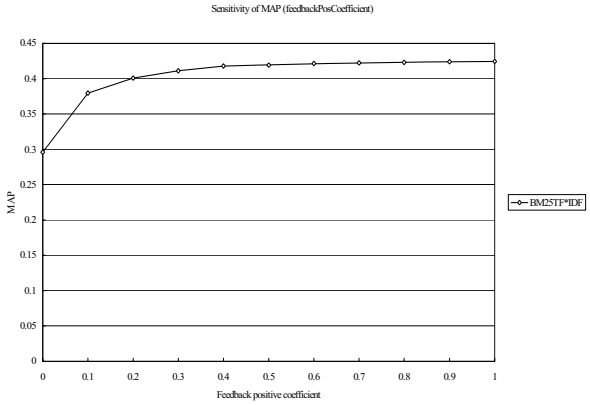
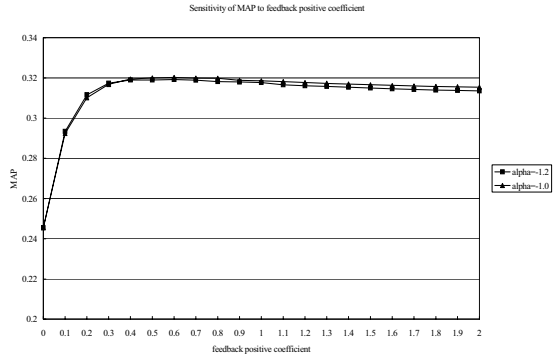


Figure 1 (Left): Recall-precision curves of NTCIR-5 YLMS-J-J-T-03, its no feedback baseline, KL-Dir run and its no feedback baseline
Figure 2 (Right): Sensitivity of MAP to feedback positive coefficient (coefficient of positive term weight) in NTCIR-5 CLIR-J-J



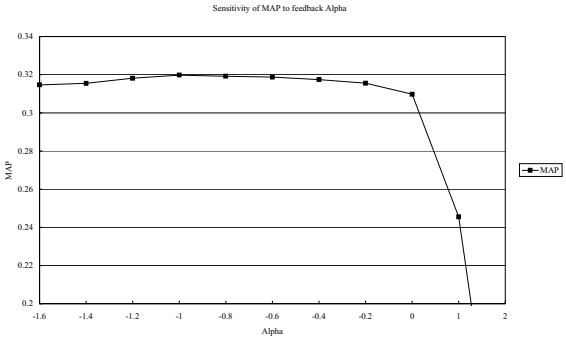


Figure 3 (Left): Sensitivity of MAP to feedback positive coefficient (coefficient of positive term weight) in NTCIR-6 CLIR-J-J, YLMS-J-J-T-01
Figure 4 (Right): Sensitivity of MAP to feedback alpha (coefficient of alpha averaging operation in positive document vector aggregation) in NTCIR-6 CLIR-J-J, YLMS-J-J-T-01

3.2.2 Crossover

We used two point crossover operator with the crossover ratio 1.0 . It takes couples of individuals, chooses randomly two positions and exchanges the part between positions of each couple.

3.2.3 Mutation

The mutation operation consists of reversing randomly chosen one bit on each chromosome.

3.2.4 Evaluation by fitness function

Each individual is evaluated by the fitness function, i.e. the MAP that the retrieval system using the decoded parameters achieved against the training collection. As we are trying to maximize the MAP against test collections, this fitness function is naturally adopted. However there may be other strategies, e.g. combining several measures in order to avoid overfitting to the training collection.

3.2.5 Elitism

Given the number of elite, the elite group of the previous generation and the same number of the best fitted individuals in the current generation are merged. The best fitted individuals as the same number as elites in this group are saved in the current generation.

3.2.6 Selection

As recommended in [7], we used tournament selection with the tournament size 4, i.e. select randomly 4 individuals from the island and take the best fitted individual to the next generation and repeat this until the next generation is complete.

4. CLIR Experiments

The details of the NTCIR-6 CLIR task are described in Kishida et al.[10].

	Test collections				
Training Collections		N3	N4	N5	N6
	N3	0.4015	0.3766	0.3973	0.3022
	N4	0.3532	0.4044	0.3616	0.3177
	N5	0.3578	0.3800	0.4525	0.3147
	N3,N4,N5	0.3941	0.3833	0.4355	0.3154
	N6	0.3565	0.3892	0.4080	0.3308

Table 6: MAP of GA parameter optimization of CLIR-J-J runs, the training collection and the test collection are the same in shadowed cells

		Parameters					
		K1	b	K4	fdDocCnt	fbTermCnt	fbPosCoeff
Training collection	N3	0.890625	0.3671875	2.12109375	5	73	0.1796875
	N4	2.625	0.5390625	1.734375	22	115	1.1328125
	N5	1.1484375	0.39453125	1.3828125	10	104	1.6953125
	N3,N4,N5	0.73828125	0.50390625	0.73828125	8	85	0.7890625
	N6	1.359375	0.55078125	2.44921875	12	71	0.6953125
	Avg.	1.35234375	0.47109375	1.68515625	11.4	89.6	0.8984375
	Offic. run	1.1	0.4	1.5	9	70	0.8

Table 7: Coefficient parameter sets optimized by 4 test collections of CLIR-J-J

Population	80	Chromosome length	45 or 24
Number of Islands	8	Mutation rate	1/45 or 1/24
Population / Island	10	Tournament size	4
Elite/Island	5	Migration rate	0.5
Crossover rate	1.0	Migration interval	5

Table 8: Control parameters of GA process and GA operations

	Test collections				
Training Collections		N3	N4	N5	N6
	N3	0.3413	0.3190	0.3216	0.2520
	N4	0.3368	0.3215	0.3237	0.2495
	N5	0.3326	0.3148	0.3300	0.2432
	N3,N4,N5	0.3407	0.3187	0.3267	0.2484
	N6	0.3379	0.3194	0.3217	0.2539

Table 9: MAP of GA parameter optimization of CLIR-J-J runs without feedback, the training collection and the test collection are the same in shadowed cells

		Parameters					
		K1	b	K4	fdDocCnt	fbTermCnt	fbPosCoeff
Training collection	N3	1.359375	0.22265625	2.7421875	0	N/A	N/A
	N4	1.91015625	0.171875	1.83984375	0	N/A	N/A
	N5	2.1796875	0.06640625	2.4140625	0	N/A	N/A
	N3,N4,N5	1.640625	0.12109375	2.98828125	0	N/A	N/A
	N6	1.6171875	0.25	2.9765625	0	N/A	N/A
	Avg.	1.74140625	0.16640625	2.5921875	0	N/A	N/A

Table 10: Coefficient parameter sets optimized by 4 test collections of CLIR-J-J without feedback

4.1 CLIR official runs for J-J SLIR

We submitted a title only run, two description only runs for the 1st stage of Japanese monolingual retrieval subtask. All the official runs are using the TF*IDF method with BM25 TF and a Rocchio feedback with a top k documents strategy. The parameters for the models are calibrated by using NTCIR-5 CLIR-J-J test collections. The exactly same parameter sets are applied for corresponding 2nd stage runs.

YLMS-J-J-T-01 is our title only run and YLMS-J-J-D-02 is our description run. YLMS-J-J-D-03 is the fusion of YLMS-J-J-D-02 and other description only

run using a bigram indexing language. Table 2,3 and 4 show effectiveness of these three official runs in 1st stage and their corresponding 2nd stage runs. Table 5 shows parameters used in the official runs.

4.2 Sensitivity of feedback effectiveness to test collection characteristics

We emphasized that the top k document feedback strategy was exceptionally successful with the NTCIR-5 CLIR-J-J test collection as shown in Figure 1. Figure 2 illustrates the situation where the higher the fbPosCoeff parameter, the better the results obtained. Because our official submission YLMS-J-J-T-01 is calibrated to NTCIR-5 CLIR-J-J, the feedback positive coefficient is adjusted to 0.8, a comparatively higher value. Figure 3 shows the sensitivity of MAP to feedback positive coefficient (coefficient of positive term weight) in NTCIR-6 CLIR-J-J (YLMS-J-J-T-01). The MAP value arrives at its best when feedback posCoeff is 0.6. The situation is quite different from NTCIR-5 CLIR-J-J.

From Figure 4, we can see that giving any value other than -1.0 to feedback alpha causes the degradation. For the following experiments, we fixed the feedback alpha parameter to -1.0.

4.3 GA optimization of NTCIR-6 runs

We tried to re-optimize the parameters of the title runs of NTCIR-6 CLIR-J-J by applying a distributed GA on a 8 node cluster servers. NTCIR-3,4,5 and 6 CLIR-J-J test collections (N3,N4,N5,N6) are used for cross-validation. Each of these test collections and combination of N3,N4 and N5 are used for training. Table 6 shows the MAP of runs where coefficient parameters are optimized by the training collection. Table 7 shows optimized parameter set against each training collection. In our GA optimized run to N5, the feedbackPosCoeff parameter is adjusted to 1.6953125, whereas 0.7890625 in GA optimized to N3,N4,N5. This is consistent with our observation that the impact of feedback is comparatively large in N5. Unlike regression approaches, where optimized coefficients are difficult to be interpreted, or genetic programming approaches, which generate incredibly complicated formulae, optimized parameters are interpreted by referring to our experiences of past empirical experiments.

See the column of N5, using N5 itself for training, the MAP of as high as 0.4525 is achieved whereas using N4 for training, the MAP of 0.3616 is even much worse than our best official title run in NTCIR-5, which achieved the MAP of 0.4193. In effect, this NTCIR-5 official run was manually calibrated with N4 [5].

GA optimization processes seem to be finding approximately the maximum fitness against the training collection. The problem is rather overfitting to the training collection. Human calibrators do much work than simply looking for the best fitness setting in the search space but they carefully do avoid overfit problems.

4.4 GA optimization without feedback

In order to illustrate different characteristics of test collections, we carried out optimizations by GA without feedback as well. As shown in Table 9, overfitting to the training set seems to be alleviated. Overfitting is mainly caused by the feedback parameters like the number of terms/documents to use for feedback or feedback coefficient, which are presumably collection dependent.

4.5 Optimization costs

As can be seen from Table 6, GA optimized N6 runs are very close to our official title only run, which is set manually. The GA optimization using N5 took 36 hours for 20 generation reiteration operated on a 8 node cluster of Xeon 3.00GHz Dual CPU machines. These hardware environments are not at all cheap even today. But it presumably pays for efforts to optimize these parameters manually from scratch. We stopped the iteration at the 20th generation because no drastic improvement is observed even at the 100th generation.

5. Conclusions

We reported our NTCIR-6 evaluation experiments of the CLIR-J-J task. We adopted a TF*IDF approach and applied GA optimization of coefficient parameters of the retrieval model. GA optimized runs achieved almost the same effectiveness as human calibrated runs, although GA runs overfitted to the training collections. Automatic calibration also illustrates different characteristics of the test collections especially such as feedback effectiveness. These optimized parameters can be utilized to help experts in system calibration.

Acknowledgments

We thank NTCIR management/secretariat members and CLIR task organizers for providing us of the data.

References

[1] Amari, S. Alpha-integration of Stochastic Evidences. In Proceedings of 2nd International Symposium on Information Geometry and its Applications, Tokyo, 2005.

[2] Fan, W., Gordon, M.D. and Pathak, P., A generic ranking function discovery framework by genetic programming for information retrieval, *Information Processing and Management*, vol. 40, issue 4, 587-602, July 2004.

[3] Fuhr, N. and Pfeifer, U. Probabilistic information retrieval as combination of abstraction, inductive learning and probabilistic assumptions. *ACM Transactions on Information Systems*, 12(1), 92–115, 1994.

[4] Fujita, S. Revisiting the Document Length Hypotheses --NTCIR-4 CLIR and Patent Experiments at Patolis. In *Working notes of the fourth NTCIR workshop meeting*, 238-245, 2004.

[5] Fujita, S. A Decade after TREC-4 –NTCIR-5 CLIR-J-J Experiments at Yahoo!Japan. In *Proceedings of NTCIR Workshop 5 Meeting*, 130-137, 2005.

[6] Gey, F. C. Inferring probability of relevance using the method of logistic regression. In *The proceedings of seventeenth annual international ACM SIGIR conference on research and development in information retrieval*, 222–231, 1994.

[7] Hiroyasu, T., Yoshida, J., Sano, M., Fukunaga, T. and Kataura, T. Distributed Genetic Algorithms ga2k (ver. 1.3) Specification, Intelligent System Design Laboratory, Doshisha University, Sep. 2002.

[8] Kanazawa, T., Takasu, A., and Adachi, J. The effects of the Relevance-based Superimposition Model for Effective Information Retrieval. *IEICE Transactions*, Vol. E83-D, No. 12, pp.2152-2160, Dec.2000.

[9] Kando, N., Kanazawa, T., and Miyazawa A. Retrieval of Web Resource Using a Fusion of Ontology-based and Content-based Retrieval with the RS Vector Space Model on a Portal for Japanese Universities and Academic Institutes. In *Proceedings of the 39th Hawaii International Conference on System Sciences (HICSS 39)*, 2006.

[10] Kishida, K., Chen, K.-H., Lee, S., Kuriyama, K., Kando, N., Chen, H. -H. and Myaeng., S. -H. Overview of CLIR task at the sixth NTCIR workshop. In *Proceedings of the Sixth NTCIR Workshop*, 2007.

[11] Lopez-Pujalte, C., Guerrero-Bote, V.P. and Moya-Anegon, F. de, Genetic algorithm in relevance feedback: a second test and new contributions, *Information Processing and Management*, vol 39, issue 5, 669-687, September 2003.

[12] Ogilvie,O. and Callan,J. Experiments Using the Lemur Toolkit, In *NIST Special Publication 500-250: The Tenth Text REtrieval Conference (TREC 2001)*,103-108, 2002.

[13] Robertson, S. and Walker S. Some Simple Effective Approximations to the 2-Poisson Model for Probabilistic Weighted Retrieval. In *Proceedings of the 1994 ACM SIGIR Conference on Research and Development in Information Retrieval*, 232-241, 1994.

[14] Robertson, S. E., Walker, S., Jones, S., M.Hancock-Beaulieu, M., and Gatford, M. Okapi at TREC-3. In *NIST Special Publication 500-226:Overview of the Third Text REtrieval Conference (TREC-3)*, 109-126, 1995.

[15] Salton, G. Automatic Text Processing –The Transformation, Analysis, and Retrieval of Information by Computer, Addison-Wesley publishing company, Massachusetts ,1988.