

An Attempt to Identify Weakest and Strongest Queries

K. L. Kwok

Queens College, City University of NY
65-30 Kissena Boulevard
Flushing, NY 11367, USA

kwok@ir.cs.qc.edu

ABSTRACT

We explore some term statistics features to be used in regression to predict the weakest and strongest of a query set. Some new metrics are introduced to optimize the choice of features. It is found that 2-word phrases need to be accounted for judiciously. A combination of inverse document frequencies and a distribution of average term frequency values for short title queries can predict correctly, with the best feature set, about 1/3 to 1/2 of the weakest and strongest 6 among 50. Longer description queries can return better results, but it seems less consistent

Categories and Subject Descriptors

H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval – *selection process*

General Terms

Experimentation, Measurement

Keywords

Query prediction, weak and strong queries

1. INTRODUCTION

Improving automatic ad-hoc retrieval results is an important objective for IR. One approach to this goal is to first detect certain properties of a query, and then apply appropriate retrieval strategies based on these properties. This is the query-specific approach to retrieval improvement introduced during the Robust Track of TREC 2003 to 2004 [1,2] and attempted by some participants [3,4]. Normal procedures to improving retrieval effectiveness such as pseudo-relevance feedback (PRF) [5], or web-assisted ad-hoc retrieval [6] apply the same procedure for all queries, and hence non-query-specific. However, if one can discover query characteristics that can predict the circumstances under which a particular procedure works, one can turn such a method to operate in a query-specific mode as well.

There are different levels for prediction of query difficulties. Predicting the actual average precision performance of a query is

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SIGIR'05, August 19, 2005, Salvador, Bahia, Brazil.

Copyright 2005 ACM

extremely challenging. Predicting the total ranking of a query set according to their average precision is also a difficult task. In TREC2004 [2], some groups use methods that can predict this better than others. We attempt here to see if it is possible to just predict only the best and the worst few queries in a TREC topic set. The idea is that these two groups of queries exhibit the greatest difference in their average precision, and that this might be reflected in some attributes that are easier to determine for their prediction.

For example, experience shows that PRF improves for about 2/3 of a given query set on average. Fig.1a is a plot of PRF versus initial average precision (from our PIRCS system) for the 249 title queries used in TREC2004. For each interval of initial retrieval effectiveness (horizontal axis), one observes that, except for the highest levels (0.75-1.0), one sees that there are more queries above the diagonal line than falling below. This shows that PRF does not work well for the strongest queries (i.e. those having excellent initial results). If one magnifies the lowest interval into smaller steps Fig.1b (e.g. 0.005), one can also see that PRF does not work well for the weakest queries that have average precision approximately in the 0.0-0.015 range. If the prediction is successful, then as a possible strategy, one may use only initial retrieval for the strongest queries, use web-assistance for the weak ones, and use a combination of web-assistance and PRF for the other queries. This may bring the most benefit to users who issue

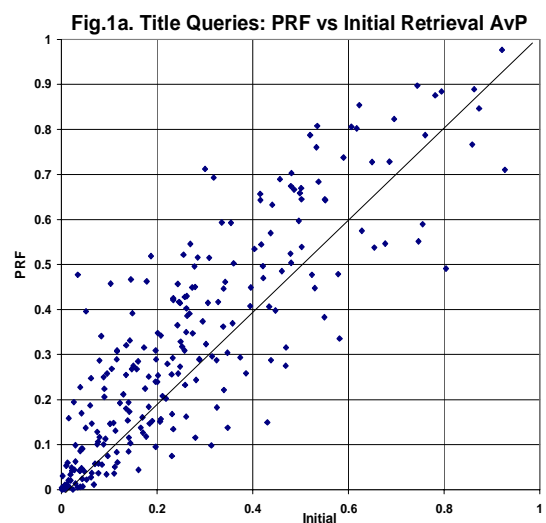
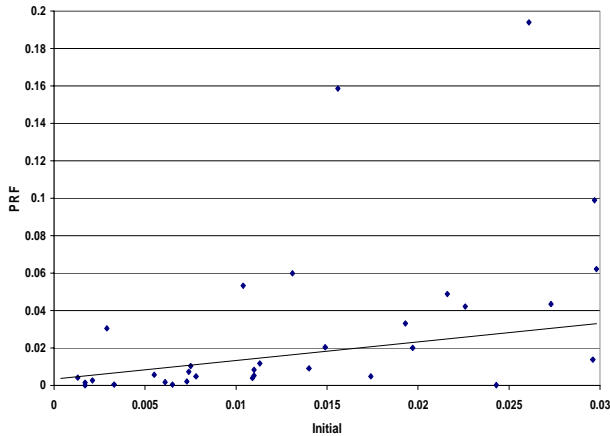


Fig.1b Title Queries - PRF vs Initial (low precision)



the weakest queries, and avoid damaging already good initial results for those who issue good strong queries.

2. METRICS

Assume that a set of queries have been ranked top to bottom. We use query agreements at the top-N and bot-N as primary measures to see how well an algorithm predicts these small sets of N queries. The *larger* these agreements are, the better. In addition, we can also measure the average distance the wrong ones within bot-N (or top-N) that a procedure predicted actually appear in the observed ranking (pre.dis), or how far the observed ones within bot-N appear in the predicted list (obs.dis). The *smaller* these distances are, the better. They will be used when there are ties in the primary measures. To illustrate, suppose we have n=10 queries (labeled a to j), and the focus is N=3 worst and best ones. Assume a sample retrieval has the observed and predicted ranking results shown as follows:

	bot(weakest)					top(strongest)				
rank->	1	2	3	4	5	6	7	8	9	10
observed->	a	b	c	d	e	f	g	h	i	j
predicted->	i	c	b	a	f	e	j	d	g	h

The result of evaluation will be:

	bot-3	Avg pre.dis	Avg obs.dis	top-3	Avg pre.dis	Avg obs.dis
sample	2(bc)	2	.33	1(h)	1.67	2.67
Perfect and Best Predictions with Agreements 2 and 1						
perfect	3	0	0	3	0	0
best-2	2	.33	.33	2	.33	.33
best-1	1	1	1	1	1	1

For our example, the number of agreement in top-3 is 1(query h). Since the predicted topics within top-3 (g and d) are distance 1 and 4 from the region in the observed ranking, the average pre.dis measure is $5/3=1.67$. Since the observed topics within bot-3

region j and i are distance 1 and 7 from the region in the predicted ranking, the obs.dis measure = 2.67. The perfect prediction and best-2 and best-1 predictions are also shown to give an idea of the possible values. We assume that once we succeed in identifying a query in a target region, its ranking within the region does not matter.

3. PREDICTION BY REGRESSION

We employ SVM regression as a tool for learning about query effectiveness – eventually we like to predict its approximate precision as well. The software we used was the LIBSVM package developed at National Taiwan University and downloadable from [7]. Linear kernel function was used. A training set consists of queries with given average precision value: AvP (returned by our PIRCS system) and a set of attributes for each query. A model is formed which is then used to predict AvP for unseen queries. Queries are sorted by predicted AvP, and those lying within the lowest (bot-N) and highest (top-N) regions are to be compared with the corresponding observed sets. The question is what features to use to characterize queries effectively for this problem.

4. CHOICE OF FEATURES

It is not obvious what properties would be most useful to characterize a weak or strong query, except that in general low frequency terms are more effective for retrieval in a query. We employed 249 title queries from the TREC-2004 Robust track to explore some features related to usage statistics for this purpose. These include:

- (a) $idf_k = \log(Nd/df_k)$ (inverse document frequency), with $Nd = \#$ of documents, $df_k =$ document frequency of term k;
- (b) $v_k = (Cf_k/df_k)^a / [\log \max(c, df_k)]$, (average term frequency), with $Cf_k =$ collection term frequency of k;
- (c) selection of terms k;
- (d) influence of phrases;
- (e) v_k distribution of all query terms.

Since idf_k or variants (a) is the basis for *ranking* documents in many retrieval algorithms, it is a natural choice. In the average term frequency v_k (b), there is a (Cf_k/df_k) factor which is similar to term burstiness used in computational linguistics [8] and is useful as an indicator for *content importance* of a term. This is modified by a $1/[\log \max(\text{constant}, df_k)]$ factor that depresses the value for high df_k terms. Trade-off between these two factors is done via a power a . of (Cf_k/df_k) . The constant c is set to 2000 and a to 1.5 as suggested in [9]. We believe it is not necessary to use all query terms; selecting the top few by some sorting will do (c). When selecting terms one can also see the influence of 2-word phrases (d) which are part of our indexing representation for retrieval. The v_k distribution (e) intends to capture the quality of the whole query through its term importance. After some experimentation, the following five intervals with mixed thresholds were found to be more effective: $df_k < 2000$, $df_k > 40000$, $v_k < .239$, $v_k < .299$, $v_k > .299$. The high and low thresholds are defined by frequencies instead, because some terms in these ranges may have abnormal high or low v values. Terms (in a query) satisfying these thresholds sequentially have their v_k values summed for the intervals.

Table 1: Results of Different Choices of Features – Short Title Queries

Av. of 150 random trials	bot-N	Av.Pre.dis	Av.Obs.dis	top-N	Av.Pre.dis	Av.Obs.dis
Predict Initial Retrieval, N=6						
f3: srt-by-v, idf value	1.55	11.85	11.16	2.75	9.58	9.14
<i>f3:srt-by-v, v value</i>	.85	16.2	14.43	1.25	14.63	18.01
<i>f3:srt-by-df, v value</i>	1.19	15.37	12.76	1.29	13.92	15.98
<i>f3:srt-by-df, idf value</i>	1.60	12.88	13.81	2.23	11.78	12.98
<i>f3-phr:srt-by-df,idf value</i>	1.35	12.04	15.09	2.48	13.8	9.85
<i>f3-phr:srt-by-v,idf value</i>	0.90	14.89	13.02	2.40	11.04	9.71
f2	1.56	11.4	10.2	2.51	9.99	9.75
f4	1.53	12.2	10.9	2.78	9.57	9.30
v5	1.13	15.0	12.7	2.04	10.6	11.8
f2+v5	2.15	11.3	9.23	2.63	9.61	9.39
f2+v3hi	1.59	11.7	10.1	2.65	9.64	9.47
f2+v3lo	1.91	12.0	9.45	2.43	10.3	9.21
f3+v5	2.10	11.4	9.93	2.69	9.69	9.13
f3+v3hi	1.63	11.8	11.9	2.82	9.21	9.16
f3+v3lo	2.07	11.2	10.0	2.51	10.3	9.03
f4+v5	2.07	11.7	10.0	2.69	9.82	9.4
Predict Initial Retrieval, N=12						
f3: srt-by-v, idf value	5.16	8.29	8.17	5.87	7.60	7.43
<i>f3:srt-by-v, v value</i>	4.74	11.2	10.4	3.88	11.9	12.4
<i>f3:srt-by-df, v value</i>	4.75	10.8	9.61	4.27	10.1	10.7
<i>f3:srt-by-df, idf value</i>	4.71	10.5	9.81	4.70	9.87	10.1
<i>f3-phr:srt-by-df,idf value</i>	4.78	8.69	10.5	5.33	10.8	8.49
<i>f3-phr:srt-by-v,idf value</i>	4.74	9.57	9.05	5.73	8.58	8.76
f2	5.25	8.71	7.92	5.97	7.18	7.81
f4	5.31	8.25	8.05	5.79	7.74	7.49
v5	4.17	10.3	9.34	5.3	7.88	8.57
f2+v5	5.77	8.77	7.63	5.86	7.56	7.22
f2+v3hi	5.45	8.31	7.73	5.95	7.21	7.47
f2+v3lo	5.69	8.88	7.72	5.87	7.52	7.29
f3+v5	5.44	8.41	8.06	5.75	7.78	7.09
f3+v3hi	5.23	8.35	8.02	5.88	7.62	7.27
f3+v3lo	5.31	8.67	8.14	5.87	7.41	7.08
f4+v5	5.42	8.79	8.13	5.72	7.78	7.25

To determine the usefulness of these attributes, we conducted experiments that involved randomly selecting 149 queries for training and the remaining 50 for testing. The goal is to see how bot-N and top-N are predicted compared to the observed. 150 random trials were repeated for each experiment, and the average number of correct predictions together with the average ‘distances’ from the regions are tabulated. The top part of Table 1 shows results of using these features based on regions of bot-6 and top-6, while the bottom part of Table 1 shows similar results with N=12.

The first six rows of Table 1 showed the selection of top 3 terms (f3) and using their idf_k or v_k as feature values based on sorting by idf_k or v_k . The first four rows for f3 show that prediction is best by selection based on v_k sorting and using idf_k as feature values (bot-6 1.55, top-6 2.75). Sort by idf_k and using idf_k values have better bot-6 prediction (1.60 vs 1.55), but much worse top-6 prediction (2.23 vs 2.75). It appears that both content importance and retrieval ranking can play a role in query difficulty indication. It may be that because our system includes

terms of 2-word phrases, which in general have lower df_k , sorting by idf_k will lead to these phrases being selected, and which are not necessarily good for prediction. Sorting by v_k can compensate for this problem because phrases generally have lower Cf_k/df_k values. To see whether phrases are useful for prediction, all 2-word phrases are excluded from the queries (affecting 118) in the next 2 rows: f3-phr. It is seen that the effectiveness drops in both bot-6 (.9-1.35) and top-6 (2.4-2.48). It seems phrases are useful in our system, but should be chosen judiciously. The rest of the runs make use of sorting by v_k and employing idf_k values. Rows f2 and f4 show use of two and four terms. They give quite close results, and f3 seems a reasonable compromise having reasonably good bottom and top predictions.

The v5 row in Table 1 shows results using the five v_k distribution values as additional features. The distribution by itself performs poorly, predicting only about 1 correct in bot-6. However, when they are used together with idf_k (e.g. f2+v5), it helps bot-6 prediction substantially to more than 2 correct, but

with a slight decrease in top-N. For these small regions of 6 positions, it seems that with these statistical term usage features, one can predict correctly only 2 out of 6 of the weakest queries, and about 2.7 of the strongest queries. On average, the observed queries are ranked some 9 to 11 positions away from the correct regions (e.g. obs.dis). Both f2+v5 and f3+v5 seem good choices.

Since our purpose is to determine top and bottom performing queries, a more suitable objective for prediction may be the rank of a query rather than its AvP. This was tried but turned out to be not as expected. For example, f3+v5 attributes for rank prediction gives results of 1.51 and 2.76 for bot-6 and top-6 respectively, and 4.88 and 5.85 when N=12. Rank prediction appears good for top-N but not for bot-N. Polynomial kernel function of degree 3 was also tried with 1.94 for bot-6, 2.66 for top-6 for rank prediction, and 1.97 for bot-6, 2.65 for top-6 for AvP prediction. These results (not shown in the table) are very similar to those using linear function.

The rows labeled f2+v3hi and f2+v3lo show results when only the higher three or lower three v_k distribution features were kept. It is seen that removing the lower two features adversely impacts on the bot-6 prediction substantially for both f2+v3hi (1.59) and f3+v3hi (1.63), but helps top-6 somewhat. Removing the higher two features however decreases both bot-6 and top-6 a little. For

predicting both bot-6 and top-6 effectively, it appears all five distribution values of v_k are useful.

Predicting effectiveness for a region of N=12 appears easier. It is seen that about 5.6-5.8 are correct in the bot-12, and about 5.8-6.0 in the top-12. They are still a bit less than 1/2 correct. The average observed queries are about 7 to 8 positions away. It appears that the f2+v5 features are good for prediction for this bigger region size. The other features f2+v3lo or f2+v3hi are also reasonable.

5. PREDICTING EFFECTIVENESS IN ROBUST TRACK QUERY SETS

Based on the choice features of Section 4 (f3, f3+v5, f2, f2+v5, etc.), we try blind prediction of the weakest and the strongest queries of the Robust Track query sets employed in TREC2003 and 2004. We use 'h' to denote the hard set, 'p' for the set 601-650 and 'q' for the 49-query set 651-700. The experimental results, shown in Table 2, use one target set (h or p or q) for testing, and the rest for training.

In Table 2 N=6, the best prediction entries for each query set are highlighted. It seems none of the feature groups are effective for both bottom and top predictions, or for which query set. f3, f3+v5 and f2 all have 7 correct top-6 predictions out of 18 of the

Table 2: Result of Predicting Weakest and Strongest Items for h='hard50', p='601-650' and q='651-700' Title Query Sets

Features:	bot-N	Av.Pre.dis	Av.Obs.dis	top-N	Av.Pre.dis	Av.Obs.dis
Robust Track Queries: Predict Initial Retrieval, N=6						
f3						
Query set 'h'	2	11.5	12.3	2	12.3	11.3
Query set 'p'	2	6.17	12.7	3	13.5	15.5
Query set 'q'	1	17.8	18.8	2	5.33	13.0
f3+v5						
Query set 'h'	2	6.33	8.50	2	12.3	13.2
Query set 'p'	3	8.50	14.5	3	8.17	17.3
Query set 'q'	0	18.3	16.0	2	5.83	12.5
f3+v3hi						
Query set 'h'	2	9.17	9.33	1	13.8	12.2
Query set 'p'	2	8.33	15.2	3	15.8	17.0
Query set 'q'	1	15.2	18	2	5.50	12.8
f3+v3lo						
Query set 'h'	2	6.33	9.17	2	12.3	12.2
Query set 'p'	3	8.50	13.2	2	15.0	16.8
Query set 'q'	1	19.2	16.8	2	5.33	12.7
f2						
Query set 'h'	0	14	11.3	2	12.3	10.5
Query set 'p'	2	6.17	14.8	3	13.5	15.0
Query set 'q'	2	16.2	14.3	2	3.83	13.3
f2+v5						
Query set 'h'	1	9.67	7.83	1	11.7	12.1
Query set 'p'	3	8.50	14.5	3	8.17	17.5
Query set 'q'	2	17.5	13.5	2	4.83	12.3
f2+v3hi						
Query set 'h'	1	13.8	9.67	1	13.8	12.0
Query set 'p'	2	8.33	15.7	3	15.8	16.5
Query set 'q'	1	20.3	14.0	2	5.50	13.3
f2+v3lo						
Query set 'h'	1	9.67	10.0	1	13.8	11.7
Query set 'p'	3	8.50	13.3	2	15.7	16.7
Query set 'q'	2	16.5	13.7	3	3.67	12.2

Robust Track Queries: Predict Initial Retrieval, N=12						
f3						
Query set 'h'	5	9.33	10.3	4	11.3	8.67
Query set 'p'	6	8.33	6.75	3	10.3	11.5
Query set 'q'	4	12.0	10.6	6	8.00	9.50
f3+v5						
Query set 'h'	6	8.25	9.08	4	9.83	8.58
Query set 'p'	5	13.9	9.25	3	10.3	14.3
Query set 'q'	5	9.33	9.33	7	5.25	7.67
f3+v3hi						
Query set 'h'	5	10.3	9.00	4	10.2	7.92
Query set 'p'	5	10.6	8.92	4	9.75	12.2
Query set 'q'	5	9.67	10.3	7	5.25	9.00
f3+v3lo						
Query set 'h'	6	8.58	8.92	4	10.3	8.67
Query set 'p'	6	11.9	7.75	3	11.9	14.4
Query set 'q'	6	9.17	9.58	7	5.25	8.08
f2						
Query set 'h'	5	9.08	8.75	4	8.67	9.58
Query set 'p'	5	11.9	7.67	4	10.2	13.8
Query set 'q'	5	11.1	9.25	7	5.25	9.83
f2+v5						
Query set 'h'	7	4.33	7.67	4	10.2	7.50
Query set 'p'	5	13.9	9.25	3	10.3	14.3
Query set 'q'	5	10.1	8.75	7	2.58	8.00
f2+v3hi						
Query set 'h'	5	8.08	8.33	5	7.75	8.58
Query set 'p'	5	10.6	9.17	3	10.3	12.3
Query set 'q'	5	9.67	9.17	7	5.25	9.08
f2+v3lo						
Query set 'h'	5	8.08	8.08	4	8.67	9.00
Query set 'p'	6	11.9	8.00	3	11.9	14.5
Query set 'q'	4	11.8	8.83	7	2.58	8.17

three query sets. Similarly for bot-6, f3+v3lo, f2+v5 and f2+v3lo have 6 correct predictions. f3+v5 has better predictions for the 'h' and 'p' sets, f2+v5 are good for the 'p' and 'q' sets, while f2+v3lo is best for the 'q' set.

For the larger regions of size 12, f3+v3hi, f2 and f2+v3hi have 15 correct top-6 predictions out of 36 of the three query sets, while f3+v3lo has a balanced 6 correct bot-12 predictions for all 3 query sets. For the 'h' set, f2+v5 predicts best, while for 'p' and 'q', it is f3+v3lo. It seems that predicting the queries is easier for bot-12 than for top-12 (except for 'q' set), while it is reverse for bot-6 and top-6.

In Table 3, we show the use of the same f3+v5 and f2+v5 features for predicting strong and weak queries composed of the longer description sections of the TREC topics. The predictions for bot-6 and bot-12 average to about 1/2 correct: better than for the title queries. Top-12 results of 17 correct out of 36 are also slightly better. The top-6 predictions for 'h' and 'p' query sets are similar to titles, but the system made no correct predictions for the 'q' set. It is not clear why this failed.

6. CONCLUSION

Employing the best combination of inverse document frequency

and a distribution of average term frequency values as features for title queries we can predict on average about 1/3 to 1/2 correct of the best and worst queries. Compared to random selections, the probability of landing one correct among six choices from a set of 50 is approximately $(44/50 * 43/49 * 6/48 * 42/47 * 41/46 * 40/45) * 6 \sim 0.41$. This drops to 0.128, 0.017, $7 * 10^{-5}$, ~ 0 and ~ 0 for the selection of two to six correct. From the training run f3+v5 (Table 1), the observed probability for bot-6 are .24, .367, .287, .066, 0 and 0 for one to six correct selections. For top-6, they are: .087, .353, .313, .173, .05 and 0. These are much better than random and show that the features do help the system to get trained to identify these top and bottom queries. These features also work for longer queries obtained from the descriptions of TREC topics except for the top-6 region. However, the results of prediction are still not sufficiently accurate to help improve ad-hoc retrieval. In the future, we plan to explore prediction with additional non-statistical features and features from an initial retrieval.

7. ACKNOWLEDGMENT

This paper benefited from the comments of an anonymous reviewer. Mr. Heng Wu helped with part of the experiments.

Table 3: Result of Predicting Weakest and Strongest Items for h='hard50', p='601-650' and q='651-700' Description Query Sets

Features:	bot-N	Av.Pre.dis	Av.Obs.dis	top-N	Av.Pre.dis	Av.Obs.dis
Robust Track Queries: Predict Initial Retrieval, N=6						
f3+v5						
Query set 'h'	3	6.50	10.7	2	13.3	4.33
Query set 'p'	3	12.0	10.5	3	14.3	16.5
Query set 'q'	2	6.67	2.67	0	6.33	9.67
f2+v5						
Query set 'h'	3	6.50	10.3	2	13.3	4.33
Query set 'p'	3	12.0	10.5	3	14.3	16.3
Query set 'q'	3	4.83	3.17	0	6.33	10.0
Robust Track Queries: Predict Initial Retrieval, N=12						
f3+v5						
Query set 'h'	8	3.17	5.67	6	6.17	6.08
Query set 'p'	5	12.0	7.42	4	11.5	11.7
Query set 'q'	7	3.75	2.83	7	3.42	3.17
f2+v5						
Query set 'h'	6	3.50	5.58	6	6.17	6.08
Query set 'p'	5	12.0	7.17	4	11.5	10.8
Query set 'q'	7	3.75	2.83	7	3.42	3.50

8. REFERENCES

- [1] Voorhees, E.M. (2004). Overview of the TREC 2003 Robust Retrieval Track In *Proc. of the Twelfth Text Retrieval Conference (TREC 2003)* Gaithersburg, MD. NIST SP 500-255, pp.69-77.
- [2] Voorhees, E.M. (200x). Draft overview of the TREC 2004 robust retrieval track. In *Working Notes of Text Retrieval Conference (TREC 2004)* Gaithersburg, MD; November 16-19, 2004. 2003, 183-190.
- [3] Amati, G., Carpineto, C. and Romano, G. (2004). Query difficulty, robustness and selective application of query expansion. In: *Proc.25th European Conference on IR*. Pp.127-137.
- [4] Yom-Tov, E, Fine, S, Carmel, D, Darlow, A and Amitay, E. (2004). Improving document retrieval according to prediction of query difficulty. In *Working Notes of Text Retrieval Conference (TREC 2004)* Gaithersburg, MD; November 16-19, 2004. pp.393-402..
- [5] Harman, D. and Buckley, C. (2004). The NRRC reliable information access (RIA) workshop. *Proc.27th Ann. Intl. ACM-SIGIR Conf. on R&D in IR*. pp.528-9.
- [6] Kwok, K.L., Grunfeld, L, Deng, P, Dinstl, N.. (2004). TREC 2004 Robust track experiments using PIRCS. In *Working Notes of Text Retrieval Conference (TREC 2004)* Gaithersburg, MD; November 16-19, 2004. pp.191-200.
- [7] Chang, C-C and Lin C-J (2004). LIBSVM: a Library for Support Vector Machines. Available at: <http://www.csie.ntu.edu.tw/~cjlin/libsvm>
- [8] Katz, S (1996). Distribution of content words and phrases in text and language modeling. *Natural Language Engineering*, 2(1), pp.15-59
- [9] Kwok, K.L. (1996). A new method of weighting query terms for ad-hoc retrieval. *Proc. 19th Annual Intl. ACM SIGIR Conf. on R&D in IR*. pp.187-195.