Time-Based Language Models

Xiaoyan Li and W. Bruce Croft
Center for Intelligent Information Retrieval
Department of Computer Science
University of Massachusetts, Amherst, MA 01003
{xiaoyan,croft}@cs.umass.edu

ABSTRACT

We explore the relationship between time and relevance using TREC ad-hoc queries. A type of query is identified that favors very recent documents. We propose a time-based language model approach to retrieval for these queries. We show how time can be incorporated into both query-likelihood models and relevance models. We carried out experiments to compare time-based language models to heuristic techniques for incorporating document recency in the ranking. Our results show that time-based models perform as well as or better than the best of the heuristic techniques.

KEYWORDS

Information retrieval, language models, relevance models, time-based language models, recency queries

1. INTRODUCTION

The task of information retrieval is to retrieve relevant documents that satisfy the user's information need. Relevance is an abstract measure of how well a document satisfies the user's information need, which is approximated by a query. In the process of approximation, a time-related information need is usually not captured by the query. For example, an old document, which is topically relevant to the query, may not satisfy the information need if the user is only interested in more recent documents. Many news-related queries would fall into this category. It is also possible that a recent document that appears to be topically relevant may not satisfy the user's information need if the user is only interested in documents within a specific period in the past. For example, the query "star wars" could have most of the relevant documents in the Reagan era rather than in recent documents. Most document retrieval systems built for the corporate environment recognize the importance of time and have provided, for many years, default rankings based on recency as well as the ability to specify a time period as a query attribute or field. The problem with these systems is that they are either based on a Boolean retrieval model or the time attribute is combined in a heuristic manner with the document scores to produce a final ranking.

In this paper, we introduce the time-based language model approach that incorporates time as part of the retrieval model. Time-based language models are a simple extension of the language model approaches to retrieval that have been developed over the past few years (e.g. [1-6]). Instead of assuming uniform prior probabilities in these retrieval models, we assign document priors based on creation dates.

In the next section, we explore the relationship between time and relevance on TREC ad-hoc title queries, and identify recency queries that favor very recent documents for evaluating the proposed models. Section 3 describes the time-based language model approaches to retrieval. Section 4 gives the experimental design and results. The experimental results show that time-based language models generally outperform heuristic techniques. Related research is discussed in section 5, and section 6 discusses future research directions.

2. TIME AND RELEVANCE

In this section, we explore the relationship between time and relevance based on an analysis of TREC ad-hoc queries. The first part shows the average distribution over time for the TREC relevant documents. The second part highlights the differences between individual queries with respect to time sensitivity.

2.1 Time and Relevance in TREC

Figure 2.1 is an example of the time distribution of relevance judgments for TREC queries (in this case, queries 251-300). The x axis represents time in months (in the past) and the y axis represents the percentage of total relevant documents. The origin corresponds to the most recent date in all the TREC collections. These averages are affected by a number of factors, such as when the collections were introduced, and which collections were used in a given year, but some trends can be observed.

In the distribution shown, relevant documents are distributed fairly evenly across the time line, but are more concentrated in the older documents. There is even a short period where there were no relevant documents, due to gaps in the source collections.

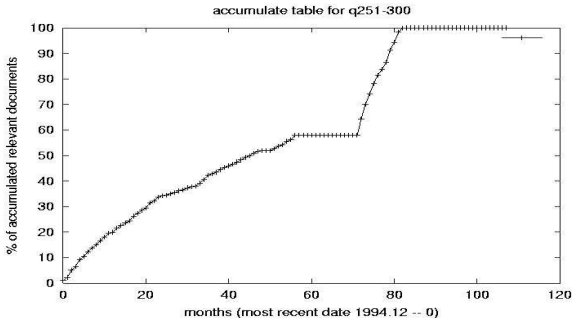

Figure 2.1: Distribution over time for relevant documents (queries 251-300)

2.2 Examples of Different Types of Queries.

Individual queries can show much more time sensitivity than the averages. As we mentioned previously, there are two main types of queries that do not have a "uniform" distribution of relevant documents over time (there are actually many types of distributions but these are more common). The first type of query favors very recent documents and the other has more relevant documents within a specific period in the past. Query 301 is an example of the first type of query. (See figure 2.2). Query 156 is an example of the second type of query, which has more relevant documents within a particular period in the past. (See figure 2.3) Query 165 is an example of a query that has a more uniform distribution of relevant documents along the time line. (See figure 2.4). This group is the most numerous, but there are still a significant number of examples of the first two types. In this paper we are more interested in the first type of queries: *recency queries*. These queries favor very recent documents. In the 100 TREC queries 301-400, we manually identified 36 recency queries that are used in the experiments described in section 4. It is important to emphasize that the distribution of relevant documents over time is substantially more biased than the background distribution of the collection.

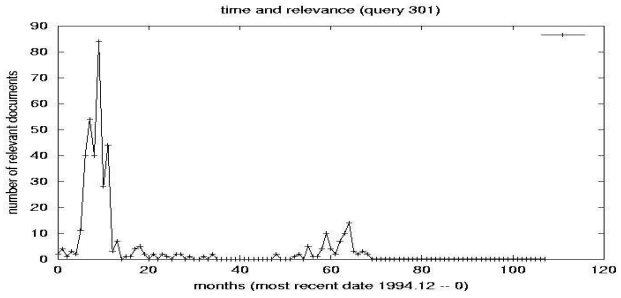

Figure 2.2: Query 301 - A "recency" query.

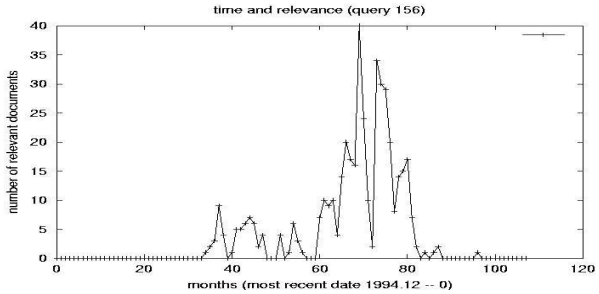

Figure 2.3: Query 156 - Relevant documents mostly in the past.

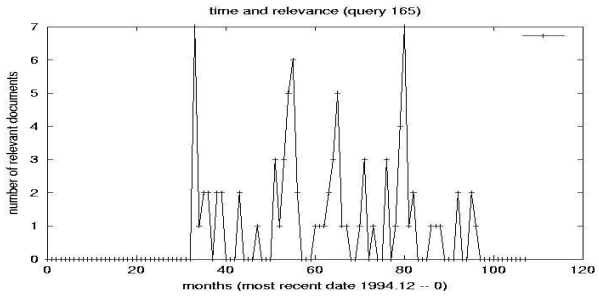

Figure 2.4: Query 165 - More uniform distribution.

3. LANGUAGE MODELS FOR RETRIEVAL

3.1 Query Likelihood Models

Language modeling frameworks were introduced to information retrieval by Ponte and Croft [1], followed by some variations [2,3,4,5] that adopted a similar framework. In the language modeling framework, there are basically three approaches to ranking documents: the query likelihood model, the document likelihood model and comparing query and document language models directly. In the simplest case, the posterior probability of a document given in (3.1) is used to rank the documents in the collection.

$$p(d/q) \propto p(q/d)p(d) \tag{3.1}$$

The prior probability of the document $p(d)$ is usually assumed to be uniform and is ignored for ranking. Ponte and Croft treat the query Q as a binary vector over the entire vocabulary and use (3.2) for estimating of the probability of generating query text (the notation M_D is used to indicate that the query is generated by a document language model).

$$P(Q/M_D) = \prod_{w \in Q} P(w/M_D) \prod_{w \notin Q} (1 - P(w/M_D)) \tag{3.2}$$

Song and Croft [2], Hiemstra [3], and Miller et al [6] treat the query Q as a sequence of independent words instead of a binary

vector and use (3.3) for query likelihood (q_w is the number of times the word w occurs in the query).

$$P(Q / M_D) = \prod_w P(w / M_D)^{q_w} \quad (3.3)$$

In the present study, the formula specified in equation (3.3) is used as a baseline in the experiments.

3.2 Relevance models

Lavrenko and Croft [5] incorporate relevance feedback and query expansion into language modeling frameworks. They proposed a technique for estimating a relevance model based on the query. The relevance model, *P(w/R)*, is estimated using a joint probability of observing the word w together with query words $q_1, q_2, ..., q_m$.

$$P(w/R) \approx P(w/Q) = \frac{p(w,q_1,...,q_m)}{P(q_1,...,q_m)} = \frac{p(w,q_1,...,q_m)}{\sum_{vocabulary} p(v,q_1,...,q_m)} \quad (3.4)$$

Lavrenko and Croft describe two methods of estimating the joint probability. The two methods differ in the independence assumptions that are being made. The first method assumes that *w* was sampled in the same way as the query words. The second method assumes that *w* and the query words were sampled using two different mechanisms. There is no significant difference on performance between these two methods and the first method was reported more efficient. Therefore, we use the first method in this paper. If we assume that *w* and $q_1, q_2, ..., q_m$ are mutually independent once we pick a distribution *M*, then we get:

$$p(w,q_1,...,q_m) = \sum_{M \in \mathrm{M}} p(M) p(w/M) \prod_{i=1}^{m} p(q_i / M) \quad (3.5)$$

Here P(M) denotes some prior probability which is kept uniform over all distributions *M*.

The KL divergence between the relevance model and a document model, which is given in equation (3.6), can be used to rank documents. Documents with smaller divergence are considered more relevant.

$$KL(R \| M_d) = \sum_w P(w / R) \log \frac{P(w / R)}{P(w / M_d)} \quad (3.6)$$

In the present study, we use equation (3.6) for the baseline relevance model in the experiments.

3.3 Time-Based Language Models

The study of the relationship between time and relevance in section 2 shows that for time-based queries, documents with different document creation dates/timestamps may have different prior probabilities for relevance. Therefore, we propose to replace p(d) in equation (3.1) and P(M) in equation (3.5) with some probability dependent on documents date T, say $p(d / T_d)$ or $p(M / T_D)$. This gives us the time-based language models:

$$p(d / q) \propto p(q / d) p(d / T_d) \quad (3.7)$$

and

$$p(w,q_1,...,q_m) = \sum_{M \in \mathrm{M}} p(M / T_D) p(w/M) \prod_{i=1}^{m} p(q_i / M) \quad (3.8)$$

Although $p(d / T_d)$ in the query likelihood language model and $p(M / T_D)$ in the relevance model have somewhat different meanings, we refer to both as $p(D / T_D)$ for simplicity. In the case of the relevance model, the time-based prior will affect the documents that are used to construct the model. When viewed as a form of query expansion, this means that the expansion will be based on the top-ranked documents subject to a time constraint, such as favoring the most recent documents. This property could be exploited to change the interpretation of a query in, for example, systems with user models that change over time.

The next challenge is to estimate the probability $p(D / T_D)$. We suggest some simple method for estimating this probability for recency queries.

For queries where recency is a major requirement of a user's information need, we used an exponential distribution for prior probability assignment. The prior $p(D / T_D)$ is given in equation (3.9). Documents with a more recent creation date are assigned higher probability.

$$p(D / T_D) = P(T_D) = \lambda e^{-\lambda(T_C - T_D)} \quad (3.9)$$

Here T_C is the most recent date (in month) in the whole collection and T_D is the creation date of a document.

The training of the parameters in equation (3.9) and the experimental results are detailed in section 4.

4. EXPERIMENTAL DESIGN AND RESULTS

4.1 Data

The data consists of 36 recency queries from TREC queries 301-400 over collections from TREC volumes 4 and volume 5. The collection we used is also time-biased since it has more documents in the recent past, which is similar to the information sources of web pages.

The data set is then randomly split into two sets: 20 queries are randomly picked for training the parameters in the time-based language models and the heuristic techniques. The other 16 recency queries are used as a test set to test the performance of different approaches. The specific queries used in these sets are listed in the appendix.

4.2 Baselines

We considered three baselines for comparison. The first baseline is retrieval using the query likelihood language models or relevance-based language models with uniform priors. The second baseline involves reranking the top N documents solely by

recency, which is determined by document creation date. We reranked the top 100 and 500 documents respectively and keep the rest of the retrieved documents unchanged. This technique is used as an option in many retrieval systems.

The third baseline is reranking documents by a linear combination of the original "topicality" rank and a "recency rank" based on creation date. The algorithm is as follows:

(1). Take top 1000 retrieved documents.

(2). Compute Score = $(1-\alpha) * R_{topicality} + \alpha * R_{recency}$

(3). Rerank documents by increasing score.

$R_{topicality}$ is the rank of a document in terms of the original belief score, i.e. the original rank in the list of retrieved documents. $R_{recency}$ is the rank of a document in terms of recency. The most recent document retrieved in the top 1000 documents will have a value 1 for $R_{recency}$.

4.3 Experimental design

We carried out four sets of training experiments and two sets of test experiments. The first set of experiments was used to determine the best value of λ in the exponential distribution on time-based query likelihood language models. The second set of experiments was to determine the best value of λ in the exponential distribution on time-based relevance language models. The third set of experiments was to determine the best value of α in the linear combination on documents retrieved with query likelihood language models. The forth set of experiments was to determine the best value of α in the linear combination on documents retrieved with relevance-based language models. For each set of training experiments, a number of different parameter values were tested. The parameter value with highest performance in terms of average precision was chosen as best parameter value for the experiments with the test set.

Table 1 shows results with the time-based relevance models on the training set. Only three values of λ are shown here, although more were tried. The best value in terms of performance was 0.02. This means that the exponential distribution given in figure 4.1 is used to assign prior probability for time-based query likelihood language models. The best value of λ was 0.01 with the time-based query likelihood model.

Table 2 shows the results using different values of α with the linear combination and the recency queries in the training set. In this case the value of 0.04 for α produced the best results.

The two sets of test experiments use parameters determined from the training experiments. The first set of test experiments is for the comparison of the time-based query likelihood model to the three baselines: query likelihood language models, reranking solely by recency and linear combinations. The second set of test experiments is for the comparison of the time-based relevance model to the three baselines: relevance-based language models, reranking solely by recency and linear combinations. The results are shown in Table 3 and Table 4 respectively. Discussion about the results is detailed in section 4.4.

Table 1: Training for Time-based Relevance Models

	RM	**TB2-0.01**	**TB2-0.02** (λ*=0.02)	**TB2-0.03**
Rel	2677	2677	2677	2677
Rret	1174	1206	1152	1130
0.00	0.602	0.669	0.682	0.675
0.10	0.441	0.497	0.501	0.501
0.20	0.342	0.384	0.414	0.396
0.30	0.275	0.317	0.331	0.312
0.40	0.216	0.248	0.274	0.249
0.50	0.168	0.177	0.191	0.182
0.60	0.126	0.128	0.126	0.104
0.70	0.084	0.086	0.070	0.058
0.80	0.052	0.044	0.037	0.030
0.90	0.017	0.014	0.012	0.011
1.00	0	0	0	0
Avg	0.1923	0.2134	0.2193	0.2078

RM: relevance-based language model.

TB2-a: time-based relevance model with λ = a in the exponential distribution.

Table 2: Training for Linear Combination

	RM	**LC-0.03**	**LC-0.04** (α*=0.04)	**LC-0.05**
Rel	2677	2677	2677	2677
Rret	1174	1174	1174	1174
0.00	0.602	0.664	0.645	0.628
0.10	0.441	0.437	0.444	0.441
0.20	0.342	0.360	0.370	0.375
0.30	0.275	0.307	0.308	0.301
0.40	0.216	0.215	0.214	0.221
0.50	0.168	0.170	0.172	0.171
0.60	0.126	0.125	0.124	0.116
0.70	0.084	0.077	0.079	0.080
0.80	0.052	0.055	0.057	0.058
0.90	0.017	0.017	0.017	0.017
1.00	0	0	0	0
Avg	0.1923	0.2027	0.2038	0.2028

LC-a: Linear Combination with α = a.

4.4 Discussion

Table 3 shows that the time-based query likelihood model with its best value of λ, which is learned from the training process, outperforms the query likelihood language model, reranking solely by recency with two document cut-off levels and the linear combination with its best value of α. Table 4 shows that the time-based relevance model with its best value of λ outperforms the first two baselines but is not as good as the linear combination with its best α value. However, after a closer look at the training process, we found that the best performance of the time-based relevance model is much better, a 7.9% increase in terms of average precision, than the best performance of the linear combination on training set. See Table 1 and Table 2. Another observation on the test set is that the performance of the

relevance-based language model is worse than the performance of the query likelihood language model. That probably explains why the time-based relevance model achieves little improvement over relevance-based language model on this test set, and doesn't perform as well as the linear combination. For the queries in the training set, on average the relevance-based language model outperforms the query likelihood language model and our time-based relevance model outperforms the relevance model and the linear combination. In both test sets of experiments, time-based language models substantially outperform reranking solely by recency at different document cut-off levels.

Table 3: Comparison of time-based query likelihood language models to baselines on test set

	LM	**RR top100**	**RR top500**	**LC-0.01**	**TB1-0.01**
Rel	2168	2168	2168	2168	2168
Rret	946	946	946	946	951
0.00	0.477	0.459	0.325	0.459	0.488
0.10	0.296	0.301	0.210	0.304	0.312
0.20	0.245	0.274	0.197	0.247	0.252
0.30	0.217	0.255	0.196	0.217	0.227
0.40	0.198	0.200	0.173	0.199	0.201
0.50	0.176	0.178	0.156	0.177	0.178
0.60	0.156	0.153	0.149	0.156	0.162
0.70	0.119	0.119	0.111	0.119	0.125
0.80	0.018	0.018	0.018	0.018	0.023
0.90	0.009	0.009	0.009	0.009	0.009
1.00	0.0009	0.0009	0.0009	0.0009	0.0010
Avg	0.1582	0.1552	0.1184	0.1590	0.1644

LM: query likelihood language model.

RR topN: Rerank top N ranked documents solely by recency.

TB1-0.01: time-based query likelihood models with λ=0.01.

LC-0.01: Linear Combination with α=0.01.

Given that on average relevance-based language models outperform query likelihood models significantly [5], it is possible to show that our time-based relevance model may outperform linear combination when a large data set is available. Figure 4.2 and figure 4.3 show the sensitivity of the results in terms of average precision to parameter values in time-based language models and linear combinations respectively on the whole data set. It can be seen that both approaches are very sensitive to this value. The results of time-based language models are more sensitive to λ when λ is in the range of [0, 0.1] than the results of linear combination with α in the same range.

Table 4: Comparison of time-based relevance models to baselines on test set

	RM	**RR top100**	**RR top500**	**LC-0.04**	**TB2-0.02**
Rel	2168	2168	2168	2168	2168
Rret	935	935	935	935	916
0.00	0.408	0.475	0.388	0.489	0.428
0.10	0.339	0.268	0.186	0.329	0.332
0.20	0.282	0.250	0.174	0.289	0.258
0.30	0.211	0.230	0.166	0.216	0.223
0.40	0.189	0.184	0.119	0.189	0.200
0.50	0.154	0.151	0.106	0.154	0.166
0.60	0.125	0.124	0.087	0.123	0.123
0.70	0.082	0.082	0.073	0.083	0.078
0.80	0.050	0.050	0.048	0.051	0.047
0.90	0.013	0.013	0.013	0.013	0.0152
1.00	0.0009	0.0009	0.0009	0.0009	0.0007
Avg	0.1557	0.1450	0.0942	0.1611	0.1577

RM: relevance model

TB2-b: time-based relevance model with λ = b in the exponential distribution

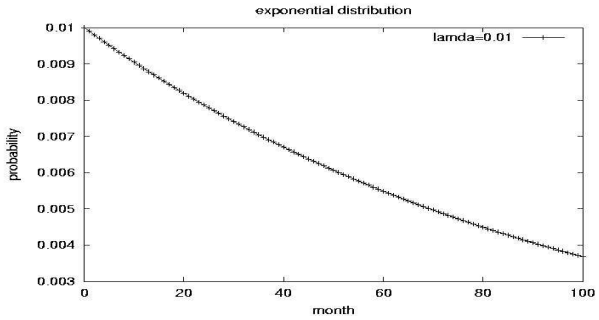

Figure 4.1: Exponential distribution used for priors

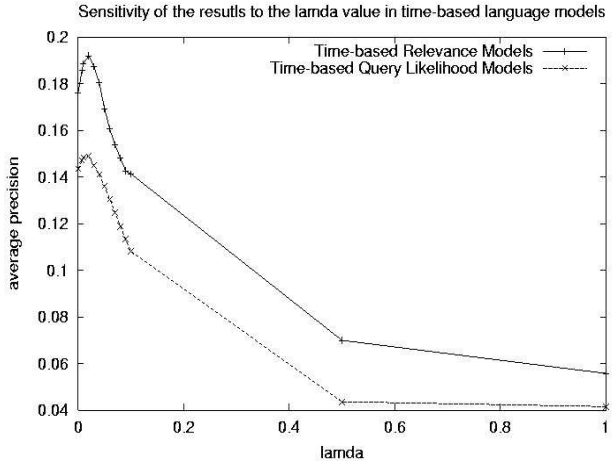

Figure 4.2: Sensitivity of average precision to λ in time-based language models

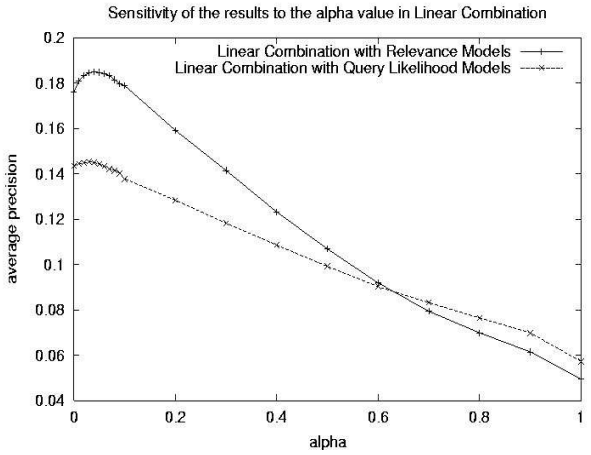

Figure 4.3: Sensitivity of average precision to α in the linear combination reranking

5. RELATED RESEARCH

As mentioned previously, the creation date of a document has long been recognized as an important attribute in commercial IR systems [12, 13]. In terms of research, the role of time in retrieval has been somewhat neglected, although recency is often mentioned in discussions of relevance and utility. There has been work on constructing timelines automatically from time-tagged retrieved documents as a visualization and discovery tool (e.g. [7, 8]). There has also been research that exploits the temporal aspect of news streams to improve topic tracking and the detection of novel information [9]. Other related work includes efforts to improve the extraction of time tags for question answering [10] and incorporating prior probabilities into language models for entry page search [11]. Unlike the fixed probability learned for each category of web pages, an exponential distribution is used in this paper to replace uniform distribution in both time-based query likelihood models and time-based relevance models.

6. CONCLUSIONS AND FUTURE WORK

In this paper, we studied the relationship between time and relevance based on TREC ad-hoc title queries. We proposed time-based language model frameworks, which incorporate time into both query likelihood language models and relevance-based language models. Exponential distributions are used to replace the uniform prior probability in these models. Our empirical results show that, for a particular set of recency queries, time-based query likelihood language models outperforms three baselines: query likelihood language models, reranking solely by recency and linear combination reranking. The time-based relevance model outperforms the relevance-based language model and reranking solely by recency. The main contribution of this work is to show that contextual features such as time constraints can be incorporated into the underlying retrieval model without resorting to heuristic approaches.

In future work, we will develop techniques to automatically classify time-based queries and set parameters. We have also started using these techniques for time-based question answering. A number of questions, such as "Who is the prime minister of Australia?", have time-dependent answers. We are attempting to use the time-based language models to change the ranking of answer passages and the subsequent answers that are extracted. Our goal is to have a time "slide bar" that would change the answer as it is moved. For this work, we are using extracted dates in addition to document dates.

7. ACKNOWLEDGMENTS

This work was supported in part by the Center for Intelligent Information Retrieval, in part by SPAWARSYSCEN-SD grant numbers N66001-99-1-8912 and N66001-02-1-8903, and in part by Advanced Research and Development Activity under contract number MDA904-01-C-0984.

Any opinions, findings and conclusions or recommendations expressed in this material are the author(s) and do not necessarily reflect those of the sponsor.

8. REFERENCES

[1] J. Ponte and W. B. Croft, "A Language Modeling Approach to information retrieval". *Proceedings of the 21st annual international ACM SIGIR conference*, 275-281, 1998.

[2] F. Song and W. B. Croft. "A general language model for information retrieva". *Proceedings of the 22nd annual international ACM SIGIR conference*, 279-280, 1999

[3] D. Hiemstra. *Using language* models *for information retrieval*. PhD thesis, University of Twente, 2001.

[4] J. Lafferty and C. Zhai. "Document language models, query models, and risk minimization for information retrieval". *Proceedings of the 24th annual international ACM SIGIR conference*, 111-119, 2001.

[5] V. Lavrenko and W. B. Croft. "Relevance-based language models". *Proceedings of the 24th annual international ACM SIGIR conference*, 120-127, 2001.

[6] D. Miller, T. Leek, and R. Schwartz. "A Hidden Markov Model information retrieval system". *Proceedings of the 22nd annual international ACM SIGIR conference*, 214-221, 1999.

[7] Swan, R. and Allan, J. "Automatic Generation of Overview Timelines". *Proceedings of SIGIR 2000 Conference*, Athens, 49-56, 2000.

[8] Swan, R. and Jensen, D. "TimeMines: Constructing Timelines with Statistical Models of Word Usage". *Proceedings of KDD 2000 Conference*, 73-80, 2000.

[9] J. Allan, R. Gupta, and V. Khandelwal. "Temporal Summaries of News Topics". *Proceedings of ACM SIGIR 01 conference*, 10-18, 2001.

[10] J. Pustejovsky, "TERQAS: Time and Event Recognition for Question Answering Systems", ARDA Workshop, MITRE, Boston (2002). (http://www.cs.brandeis.edu/~jamesp/arda/time/index.html)

[11] K. Wessel, W. Thijs, and H. Djoerd. "The Importance of Prior Probabilities for Entry Page Search", Proceddings of SIGIR 2002, 27-34.

[12] http://www.google.com.

[13] http://www.teoma.com

APPENDIX: QUERIES USED IN EXPERIMENTS

(1) Training set consists of following TREC queries:

302, 304, 306, 319, 321, 330, 333, 334, 340, 345, 351, 352, 355, 370, 378, 382, 385, 391, 395, 396

(2) Test set consists of following queries:

346, 400, 301, 356, 311, 337, 389, 307, 326, 329, 316, 376, 357, 387, 320, 347