Learning to Select a Time-aware Retrieval Model

Nattiya Kanhabua
L3S Research Center
Leibniz Universität Hannover
Hannover, Germany
kanhabua@L3S.de

Klaus Berberich
Max-Planck Institute for Informatics
Saarbrücken, Germany
kberberi@mpi-inf.mpg.de

Kjetil Nørvåg
Dept. of Computer Science
Norwegian University of Science and Technology
Trondheim, Norway
noervaag@idi.ntnu.no

ABSTRACT

Time-aware retrieval models exploit one of two time dimensions, namely, (a) *publication time* or (b) *content time* (temporal expressions mentioned in documents). We show that the effectiveness for a *temporal query* (e.g., illinois earthquake 1968) depends significantly on which time dimension is factored into ranking results. Motivated by this, we propose a machine learning approach to select the most suitable time-aware retrieval model for a given temporal query. Our method uses three classes of features obtained from analyzing distributions over two time dimensions, a distribution over terms, and retrieval scores within top-k result documents. Experiments on real-world data with crowdsourced relevance assessments show the potential of our approach.

Categories and Subject Descriptors H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval
General Terms Algorithms, Experimentation
Keywords temporal queries, time-aware ranking prediction

1. INTRODUCTION

Previous work [1, 4] has shown that the retrieval effectiveness of temporal queries can be significantly improved by modeling and taking into account *publication time* (i.e., when a document was published) or *content time* (i.e., what time a document refers to). The right choice of time-aware retrieval model can make a huge difference for a given temporal query, as we observe empirically. Thus, the model based on publication time proposed in [4] (labeled *PT-Rank*) performs best for temporal queries like iraq 2001 and mac os x 24 march 2001, whereas the model based on content time from [1] (labeled *CT-Rank*) performs best for temporal queries like sound of music 1960s and michael jackson 1982.

Our contribution in this work is *a novel machine learning approach to select the most suitable time-aware retrieval model for a given temporal query* – to the best of our knowledge the first approach tackling this objective. It uses three classes of features obtained from analyzing distributions over two time dimensions, a distribution over terms, and retrieval scores within top-k result documents. We further present experimental results, showing the significance of the problem addressed and the effectiveness of our approach.

2. MODEL

A document d consists of a textual part d_{text} (a bag of words) and a temporal part d_{time} composed of its publication time $PubTime(d)$, and temporal expressions $\{t_1, \ldots, t_k\}$ mentioned in d, denoted $ContentTime(d)$. A temporal query q consists of keywords q_{text}, and temporal expressions q_{time}. *PT-Rank* and *CT-Rank* both employ a mixture model that linearly combines *textual similarity* and *temporal similarity* between q and d as $S(q,d) = (1-\alpha) \cdot S'(q_{text}, d_{text}) + \alpha \cdot S''(q_{time}, d_{time})$ using a mixing parameter α. The *textual similarity* S' can be determined using any existing term-based retrieval model (e.g., tf.idf or a unigram language model). The *temporal similarity* S'' is determined assuming that temporal expressions in the query are generated independently from a two-step generative model, i.e.:

$$S''(q_{time}, d_{time}) = \prod_{t_q \in q_{time}} \frac{1}{|d_{time}|} \sum_{t_d \in d_{time}} P(t_q|t_d) \; .$$

For *CT-Rank* [1] the probability $P(t_q|t_d)$ is estimated according to the LMTU method based on *content time*. For *PT-Rank* [4] $P(t_q|t_d)$ is estimated based on *publication time* using an exponential decay function. For both methods Jelinek-Mercer smoothing eliminates zero probabilities.

3. SELECTING A RETRIEVAL MODEL

Given a temporal query q, we will predict which time-aware retrieval model achieves the best effectiveness by learning a prediction model using three classes of features:

Temporal KL-divergence, originally proposed in [3], measures the difference between the distribution of publication time within a set of top-k result documents D_q and their distribution in the overall document collection C. This definition thus only considers publication time, and we further refer to it as KL_{PT}. While it gives a strong signal, for instance, when all relevant documents were published around the occurrence of an important *real-world event* (e.g., a sports tournament), it does not capture when they all refer to a common time period (e.g., the 19th century). We therefore adapt temporal KL-divergence to also consider *content time* as $KL_{CT}(D_q||C,q) = \sum_{t \in T_C} P(t|q) \cdot \log \frac{P(t|q)}{P(t|T_C)}$, where T_C is a set of all temporal expressions in C. $P(t|T_C)$ is the probability of a temporal expression t in C. $P(t|q)$ is the probability of generating a temporal expression t given q: $P(t|q) = \sum_{d \in D_q} P(t|d) \cdot \frac{P(q|d)}{\sum_{d' \in D_q} P(q|d')}$, where $P(q|d)$ is a retrieval score of d wrt. a particular retrieval model. Since a document can contain more than one temporal expression, $P(t|d) = \frac{c(t,d)}{\sum_{t' \in d} c(t',d)}$, where $c(t',d)$ is the number of occurrences of t' in d. Again, we employ Jelinek-Mercer smoothing when estimating $P(t|d)$ to avoid zero probabilities.

SIGIR'12, August 12–16, 2012, Portland, Oregon, USA.
ACM 978-1-4503-1472-5/12/08.

As suggested in [3], temporal features alone could not achieve high accuracy for query classification. Thus, we also employ a **clarity score** [2] for measuring the KL-divergence between the distribution of terms within top-k results D_q and their distribution in the overall document collection C. A clarity score can be computed as $Clarity = \sum_{w \in V} P(w|q) \cdot \log \frac{P(w|q)}{P(w|C)}$, where w is a term from the vocabulary V of all distinct terms in C. $P(w|q)$ is the probability of generating w given q and $P(w|C)$ is the probability of w in C.

Retrieval scores can also be exploited to select a retrieval model, as proposed in [5]. We employ different features obtained from analyzing/comparing the retrieval scores of a term-based baseline model that is *not* time-aware, *PT-Rank*, and *CT-Rank*, namely: 1) average score of the baseline (AVG_{base}), 2) average score of *PT-Rank* ($AVG_{PT\text{-}Rank}$), 3) average score of *CT-Rank* ($AVG_{CT\text{-}Rank}$), and 4) the divergence of retrieval scores according to *PT-Rank* and *CT-Rank* from those produced by the baseline ($JS_{PT\text{-}Rank}$ and $JS_{CT\text{-}Rank}$). We employ Jensen-Shannon divergence to measure the extent to which the time-aware models alter the scores of the baseline retrieval model, formally:

$$JS(S_b||S_r, q) = \sum_{d \in D_q} S_b(q,d) \cdot \log \frac{S_b(q,d)}{\frac{1}{2} \cdot S_b(q,d) + \frac{1}{2} \cdot S_r(q,d)} \cdot$$

where $S_b(q,d)$ is the retrieval score of d according to the baseline S_b. $S_r(q,d)$ is the score of d when ranked using a time-aware retrieval model $S_r \in \{PT\text{-}Rank, CT\text{-}Rank\}$.

4. EXPERIMENTS

We conducted two sets of experiments: 1) evaluate our prediction model as classification accuracy, and 2) demonstrate how an accurate choice of the retrieval model can improve retrieval effectiveness. We used the New York Times Annotated Corpus containing 1.8M documents published between 1987 and 2007, and the 40 temporal queries and relevance assessments from [1]. Temporal expressions were extracted using the TARSQI Toolkit. Documents were indexed and retrieved with Apache Lucene 2.9.3 using its default similarity function as a baseline retrieval model. We consider both *inclusive* and *exclusive* modes of evaluating queries, described in [1], that differ in whether temporal expressions are also treated as textual query terms. The mixture parameter α was determined empirically: $\alpha = 0.5$ for *PT-Rank* and $\alpha = 0.6$ for *CT-Rank* in *inclusive*, and $\alpha = 0.5$ for *PT-Rank* and $\alpha = 0.1$ for *CT-Rank* in *exclusive*. The parameters for TSU were: $DecayRate = 0.5$, $\lambda = 0.5$, and $\mu = 6$ months. For LMTU, smoothing γ was 0.75. For temporal KL-divergence, smoothing was set to 0.1.

For classification, each query was labeled according to whether *PT-Rank* or *CT-Rank* performs best on it. More precisely, we assumed the model with the best MAP as a *query label*. We excluded queries with a small difference in MAP of two time-aware models. We learned a prediction model using several algorithms: decision tree, Naïve Bayes, neural network and SVM, using 10-fold cross-validation with 10 repetitions. We measured statistical significance using a t-test with $p < 0.05$. In the tables, bold face indicates statistically significant difference from the respective baseline.

Classification results. The baseline method for query classification is the majority classifier. The accuracy of the baseline is 0.54 for *exclusive* and 0.60 for *inclusive*. Table 1 shows the accuracy of the best-performing classifiers, i.e., SVM for *exclusive* and decision tree for *inclusive*. The results show that prediction accuracy tends to be better when using $k = 100$ rather than $k = 500$. One reason for this is that with the larger number of top-k documents, more irrelevant documents are introduced into the analysis. The performance among different feature classes shows that $JS_{PT\text{-}Rank}$ performs well in most case. For *exclusive*, using a small number of top-k documents is better than a large number of top-k documents. For top-100, $JS_{PT\text{-}Rank}$ outperforms the baseline classifier and other features significantly (accuracy=0.72). For top-500, all single features perform worse compared to the baseline classifier. For *inclusive*, the performance of top-100 is better than top-500. For top-100, the best performing feature is the combination of *Clarity*, $JS_{PT\text{-}Rank}$ and $JS_{CT\text{-}Rank}$, which achieves an accuracy of 0.75.

Table 1: Accuracy of query classification.

Feature	*exclusive*		*inclusive*	
	100	*500*	*100*	*500*
Clarity	0.51	0.53	0.59	0.60
KL_{PT}	0.53	0.53	0.60	0.59
KL_{CT}	0.53	0.53	0.60	0.60
AVG_{Base}	0.53	0.53	0.63	0.56
$AVG_{PT\text{-}Rank}$	0.53	0.53	0.60	0.60
$AVG_{CT\text{-}Rank}$	0.53	0.53	0.60	0.59
$JS_{PT\text{-}Rank}$	**0.72**	0.42	**0.74**	0.64
$JS_{CT\text{-}Rank}$	0.38	0.42	0.60	0.60
$Clarity+KL_{PT}+KL_{CT}$	0.54	**0.65**	0.61	0.61
$Clarity+JS_{PT\text{-}Rank}+JS_{CT\text{-}Rank}$	0.42	**0.65**	**0.75**	0.61

Table 2: Effectiveness of different retrieval models.

Method	*exclusive*			*inclusive*		
	P@1	*P@5*	*MAP*	*P@1*	*P@5*	*MAP*
CT-Rank	0.55	0.50	0.53	0.58	0.55	0.56
PT-Rank	**0.63**	0.53	0.55	0.63	0.58	0.61
PR	**0.68**	0.53	**0.59**	**0.70**	0.58	**0.64**
MAX	**0.83**	**0.61**	**0.64**	**0.78**	**0.62**	**0.67**

Retrieval results. For each query, we determined retrieval results using a model chosen according to the best prediction model determined in the previous experiment, such as, 1) $JS_{PT\text{-}Rank}$ for retrieval in *exclusive*, and 2) *Clarity*+ $JS_{PT\text{-}Rank}$+$JS_{CT\text{-}Rank}$ for retrieval in *inclusive*. Table 2 shows the effectiveness of different retrieval models, where *PR* is the retrieval model based on our prediction model. *MAX* is the maximum (or optimal) effectiveness that can be achieved, that is, if a prediction model performs accurately 100%. The retrieval results are compared with the baseline method *CT-Rank*. The results show that our prediction-based retrieval model (*PR*) outperforms the baseline significantly in P@1 and MAP. However, we note that it is difficult for *PR* to achieve the optimal effectiveness because of the classification accuracy as explained above.

5. CONCLUSIONS

We have demonstrated that selecting the right time-aware retrieval model can have a significant impact on the retrieval effectiveness of temporal queries. We proposed a novel machine learning approach to do so automatically and demonstrated its effectiveness through extensive experiments.

6. REFERENCES

[1] K. Berberich, S. Bedathur, O. Alonso, and G. Weikum. A language modeling approach for temporal information needs. In *Proceedings of ECIR'2010*, 2010.

[2] S. Cronen-Townsend, Y. Zhou, and W. B. Croft. Predicting query performance. In *Proceedings of SIGIR'2002*, 2002.

[3] R. Jones and F. Diaz. Temporal profiles of queries. *ACM Trans. Inf. Syst.*, 25, July 2007.

[4] N. Kanhabua and K. Nørvåg. Determining time of queries for re-ranking search results. In *Proceedings of ECDL'2010*, 2010.

[5] J. Peng, C. Macdonald, and I. Ounis. Learning to select a ranking function. In *Proceedings of ECIR'2010*, 2010.