www.ijecs.in
*International Journal Of Engineering And Computer Science ISSN: 2319-7242*
*Volume 4 Issue 9 Sep 2015, Page No. 14118-14122*

# Query Response Ranking by Temporal Diversity using User Sessions as Feedbacks

***Divya Seelam[1], B.Hanmanthu[2], B.Ragu Ram[3]***

[1]Kakatiya University, Kakatiya Institute of Technology and Science,
Yerragattu, Warangal-506 015, Telangana, India.
*divyaseelam553@gmail.com*

[2]Kakatiya Institute of Technology and Science, Kakatiya University,
Yerragattu, Warangal-506 015, Telangana, India.
*bhcsekits@gmail.com*

[3]Kakatiya Institute of Technology and Science, Kakatiya University,
Yerragattu, Warangal-506 015, Telangana, India.
*raghu9b.naik@gmail.com*

**Abstract:** ***Retrieval efficiency of temporal issues can be enhanced by getting into account the time aspect. Latest temporal ranking systems use a couple of main strategies: 1) a mixture unit linearly integrating textual resemblance as well as temporal resemblance, and 2) a probabilistic system producing a query from the textual as well as temporal component of report automatically. In this document, we suggest a unique time-aware ranking system according to learning-to-rank strategies. We use two classes of attributes for understanding a ranking system, entity-based as well as temporal attributes, which are based on annotation information. Entity-based attributes are targeted at acquiring the semantic resemblance around a query as well as a document, while temporal attributes determine the temporal resemblance. By using considerable studies we reveal that our ranking system considerably enhances the retrieval efficiency over current time-aware ranking systems.***

**Keywords:** about Query Response Ranking, Temporal Diversity, User Sessions, Feedbacks.

## 1. Introduction

Query Response Ranking, Temporal Diversity, and User Sessions, Feedbacks-based and temporal attributes. Therefore, the major efforts of this document are: 1. a unique time-aware ranking system incorporated utilizing two classes of attributes, 2. recognition of proper attributes to be utilized, and 3. considerable experiments for assessing the suggested time-aware ranking system and the attributes, utilizing the New York Times Annotated Corpus in blend with temporal concerns and significance evaluations from [2].

The planning of the rest of the document is as observe. In segment 2, we provide a review of associated work. In Section 3, we describe the systems for reports, annotated reports, as well as temporal concerns, also provide the ranking system. In Section 4, we suggest two classes of attributes for understanding a time-aware ranking system. In Section 5, we assess the suggested ranking system by evaluating with established time-aware ranking techniques. Lastly, in Section 6, we determine the document.

## 2. Previous scenario

An amount of ranking systems exploiting temporal data being suggested, like [2, 7, 16, 18]. In [16], Li as well as Croft included time into language systems, known as time-based language systems, by determining a document earlier utilizing an rapid decay function of a report generation date. They centered on recency concerns, where the newer reports get higher possibilities of significance. In [7], Diaz as well as Jones also utilized report generation dates to evaluate the delivery of retrieved reports also generate the temporal account of a query. They revealed that the temporal account collectively with the information of recovered reports can enhance average preciseness for the query through a set of various attributes for discriminating around temporal information. Berberich et al. [2] incorporated temporal expressions towards query-likelihood language modeling that views chaos inherent to temporal expressions in a query as well as reports, i.e., temporal expressions can pertain to the equivalent time interval consistent if they aren't precisely equal. Metzler et al. [18] viewed implicit temporal data requires. They suggested exploration query logs as well as evaluate query frequencies eventually being determined strongly time-related concerns. Furthermore, they provided ranking regarding implicit temporal specifications, also the empirical outcomes revealed the enhancement of the retrieval efficiency of temporal concerns for web search.

Together with the work described, there is work that has centered on recency ranking [4, 8, 9, 10]. That perform is assorted to our efforts in expression of a search event, while in our situation a user may distribute time as component of a query, purported temporal requirements. There is work on evaluating queries eventually, e.g., Kulkarni et al. [15] analyzed how users' data requires change eventually, also Shokouhi [21] used a time series evaluation technique for discovering seasonal concerns. Concerning an entity-ranking task, Demartini et al. [6] evaluated news record (i.e., past associated articles) for determining appropriate entities in current news reports.

## 3. Proposed scenario

### 3.1 Advanced Document model

Our document assortment is serene of formless text documents: $C = \{d_1....d_n\}$. A document d is displayed as a bag-of-words or an unordered number of terms: $d_i = \{w_1....w_k\}$, here its periodical date is signified PubTime($d_i$). As the two classes of attributes are taken from annotating data of reports, we provide a system of annotated reports as observe.

For every document $d_i$, its connected annotated document $\hat{d}_i$ is serene of 3 components. Initial, $\hat{d}_i$ consists of a set of named entities $\{e_1....e_n\}$, here a named entity may be a person, location, or firm. The next part is a group of temporal expressions or occasion dates $\{t_1....t_m\}$. Lastly, $\hat{d}_i$ consists of a group of sentences $\{s_1....s_z\}$, here every sentence $s_y$ contains of tokens (terms), the part-of-speech as well as position details of every token.

A vital feature is that we differentiate around two temporal dimensions connected with a document $d_i$: 1) publication time (i.e., when a paper was released), and 2) content time (i.e., what time a paper pertains to). The content time of a paper (denoted ContentTime($d_i$) or temporal terms revealed in $d_i$ can be completely taken utilizing a time as well as event recognition algorithm. The algorithm mines temporal expressions revealed in a document as well as normalizes them to occasions as they may be secured on a timeline. As revealed in [1], temporal expressions may be implicit, explicit or relative. Instances of specific temporal expressions are May 25, 2012 or June 17, 2011 that may be mapped exclusively to dates months, or years in the Gregorian calendar. An implied temporal expression can be an imprecise time point or period, e.g., Independence Day 2011 that is mapped to July 04, 2011. Instances of related temporal expressions are yesterday, last week or one month ago.

### 3.2 Temporal Query Model

A temporal query $q_j$ is consisting of two elements: keywords $q_{text}$, as well as temporal expressions $q_{time}$. Remember that a temporal expression may be explicitly supplied as a part of temporal query, or implicitly supplied. An instance of the initial type is Illinois earthquake 1968 here the customer is involved in reports concerning Illinois earthquake in 1968. Concerns of the second type might be implicitly connected with specific time specifically queries associated with major real-world activities, or seasonal issues [21]. Instances of a real-world occasion query as well as a seasonal query tend to be Thailand tsunami related to the year 2004 also U.S. presidential election, that might be connected with the years 2000, 2004, also 2008. When $q_{time}$ is not provided explicitly by the customer, it should be decided by the system [14]. In this document, we consider that $q_{time}$ is explicitly supplied.

### 3.3 Ranking Model

Place A ranking system $h(d,q)$ is provided by instructing a group of described query/document sets utilizing a learning algorithm. A learned ranking system is effectively a weighted coefficient $w_i$ of a attribute $x_i$. A concealed document/query set $(d',q')$ can be ranked based on a weighted sum of attribute scores:

$$score(d',q') = \sum_{i=1}^{N} w_i \times x_i^{q'}$$

Here N is the amount of attributes. Numerous current learning algorithms being suggested, might be classified into three techniques: pointwise, pairwise, also listwise techniques. For most specified criteria of every strategy, please consider [17]. In this work, we use different learning-to-rank algorithms, like, RankSVM [11], S $SVM^{MAP}$ [22], and three random gradient descent algorithms: SGD-SVM [23], PegasosSVM [20], also PA- Perceptron [3].

## 4. Features

### 4.1 Temporal Features

Temporal attributes describe the temporal resemblance between a query as well as a document. In this process, we use five different techniques for evaluating temporal resemblance: $LmT$ and $LmtU$ [2], TS and TSU [14], and FuzzySet [12].

The time-aware ranking techniques we analyze vary from every other in two primary features: 1) if time uncertainty is involved, also 2) if the magazine time or the content time of a report is utilized in ranking. $LmT$ ignores time ambiguity also it uses the content time of d. $LmT$ may be described as:

$$P(t_q / t_d)_{LMT} = \{0 \text{ if } t_q \neq t_d \\ 1 \text{ if } t_q = t_d$$

where $t_d \in ContentTime(d)$, also the score can be equal to 1 if a temporal expression $t_d$ is quite equal to $t_q$. $LmtU$ involves time concern by supposing equal possibility for any time period $t'_q$ that $t_q$ can consider, that is, $t_q = \{t'_q \mid t'_q \in t_q\}$. The basic computation of $P(t_q / t_d)$ for $LmtU$ is provided as:

$$P(t_q / t_d)_{LMTU} = \frac{t_q \cap t_d}{|t_q| . |t_d|}$$

Where $t_d \in ContentTime(d)$. The outlined calculation of $t_q \cap t_d$, $|t_q|$ and $|t_d|$ is illustrated in [2].

### 4.2 Entity-Based Features

Together with temporal attributes shown above, we utilize ten entity-based attributes intended for evaluating the resemblance between a query as well as a document. The intuition will be a classic term-matching technique that utilize best statistics, e.g., TFIDF, disregard the semantic task of a query term. For instance, determine the temporal query Iraq 2001. A statistics-based system can rank a record having numerous events of the provisions Iraq or 2001 compared to a report with fewer occurrences of the similar provisions without getting into account a semantic connection around query provisions, which may be measured by, e.g., a term distance in a conviction.

Entity-based attributes are calculated for every entity $e_j$ in an annotated document $\hat{d}_i$ also the suggested attributes contains *querySim, title, titleSim, senPos, senLen, cntSenSubj, cntEvent, cntEventSubj, timeDist,* also *tagSim* [13]. The first attribute *querySim* is the term resemblance score around $q_j$ as well as an entity $e_j$ in $\hat{d}_i$. Here, we utilize Jaccard coefficient for determining term resemblance. Component *title* contains whether $e_j$ is in the title of $\hat{d}_i$. Attribute *titleSim* is the term resemblance score around $e_j$ as well as the title. Attribute *senPos* provides a normalized score of the rank of the 1st sentence here $e_j$ happens in $d_i$ while the attribute *senLen* provides a normalized score of the length of the 1st sentence of $e_j$. Attribute *cntSenSubj* is a normalized score of the amount of sentences here $e_j$ is a subject. Attribute *cntEvent* can be a normalized score of the amount of event content (or sentences annotated using temporal expressions) of $e_j$, whereas the attribute *cntEventSubj* a normalized score of the amount of event content that $e_j$ is a subject. Attribute *timeDist* is a normalized distance score of $e_j$ and a temporal expression inside a sentence. Attribute *tagSim* is the term resemblance score around $e_j$ as well as an entity described in $d_i$. Observe that the final feature is just an appropriate for a document selection supplied with tags (e.g., the New York Times Annotated Corpus).

# 5. Experiments

In this segment, we consider various time-aware ranking systems according to learning-to-rank algorithms. We initially illustrate the empirical setting. Perhaps, we reveal the empirical outcomes and execute a feature evaluation.

### 5.1 Experimental Setting

We utilized the New York Times Annotated Corpus (incorporating over 1.8 million news reports released around January 1987 as well as June 2007) as a temporal report selection, also the 40 temporal concerns and crowdsourced significance evaluations from [2]. We utilized a collection of language operating tools for annotating records, such as OpenNLP (for tokenization, sentence dividing as well as part-of-speech tagging, and also short parsing), the SuperSense tagger (for named entity identification) as well as TARSQI Toolkit (for annotating reports with TimeML). The outcome of this is for every report: 1) entity data, e.g., each of persons, locations as well as corporations, 2) temporal expressions, e.g., most of event dates, also 3) sentence details, e.g., each sentences, entities as well as event dates happens in every sentence, and position details. For temporal attributes, an exponential $DecayRate = 0.5$, also $\lambda_2 = 0.5$ are utilized. The fuzzySet variables are n = 2, m = 2 $a_1 = a_2 - (0.25 \times (a_3 - a_2))$, as well as $a_4 = a_3 - (0.50 \times (a_3 - a_2))$. The smoothing variable $\lambda_1$ is assorted, and merely the outcomes of those obtained best can be revealed.

Towards learning a time-aware ranking system, we used various learning-to-rank algorithms, in which default variables of every learner were utilized. We carried out five-fold cross recognition by arbitrarily partitioning 40 temporal concerns into five folds (8 concerns per fold): F1, F2, F3, F4, and F5. For every fold, four different folds (4*8=32 concerns) are utilized for learning a ranking system. To assess ranking systems, the Apache Lucene search engine version 2.9.3 was used. We get five efficient baselines. The initial baseline is Lucene's standard similarity function (a version of TFIDF). The four another baselines are suggested in [2]: LmT-IN, LmT-EX, LmtU-IN, also LmtU-EX, here suffixes IN as well as EX consider inclusive as well as exclusive mode correspondingly (either query's temporal expressions are even incorporated as a component of query keywords $q_{text}$ or tend to be excluded). The baseline TFIDF addresses query's temporal expressions as a component of $q_{text}$ i.e., the comprehensive mode. The asses results of time-aware ranking is determined by the consistency at 1, 5 also 10 reports (P@1, P@5 also P@10 correspondingly), Mean Reciprocal Rank (MRR), as well as Mean Average Precision (MAP). The average efficiency around the five folds is utilized to determine the complete efficiency of every ranking system. For every test, we determine statistical weight utilizing a t-test with $p < 0.05$. In the tables of outcomes, bold face is utilized to show statistically great change from the particular baselines.

### 5.2 Experimental Results

The ranking efficiency of the baselines as well as learned ranking systems are showed in Table 1. The outcomes among the baselines are equivalent to that revealed in [2]. In common, the distinctive mode carried out better than the comprehensive mode for both $LmT$ as well as $LmtU$, also LmtU-EX obtained the optimum efficiency over the another baselines.

Evaluating various ranking systems, RankSVM didn't generate a considerable enhancement above the baselines, though PegasosSVM carried out more compared to the baselines and another learned ranking systems. SGD-SVM as well as PegasosSVM obtained the enhancement over the baselines in every specification. Lastly, the listwise ranking $SVM^{MAP}$ carried out better than the pairwise systems, and outperformed each of the baselines considerably. Utilizing P@1, $SVM^{MAP}$ obtained the enhancement over TFIDF as well as LmtU-EX up to 27.5% as well as 15% correspondingly. Utilizing MAP, $SVM^{MAP}$ obtained the enhancement over TFIDF as well as LmtU-EX up to 13.1% as well as 8.2% correspondingly.

Being comprehend the significance of every feature, we carried out feature evaluation and the

Outcomes are revealed in Table 2. $\hat{x}_i$ is the average

| Model | P@1 | P@5 | P@10 | MRR | MAP |
|---|---|---|---|---|---|
| TFIDF | .375 | .435 | .410 | .562 | .486 |
| LMT-IN | .500 | .370 | .373 | .625 | .428 |
| LMT-EX | .425 | .395 | .385 | .588 | .447 |
| LMTU-IN | .475 | .450 | .433 | .635 | .475 |
| LMTU-EX | .500 | .520 | .520 | .670 | .535 |
| RankSVM | .500 | .550 | .515 | .661 | .578 |
| SGD-SVM | .575 | .610 | .540 | .706 | .595 |
| PegasosSVM | .550 | .610 | .543 | .690 | .595 |
| PA- | .500 | .455 | .433 | .630 | .496 |
| $SVM^{MAP}$ | .650 | .605 | .565 | .753 | .617 |

Table 1: Effectiveness of different ranking models.

of every feature's standards. $w_i$ is a feature's weight received from the learning technique SVMMAP. The top-5 attributes with maximum weights are querySim, TS, FuzzySet, retScore also senPos. Entity-based attributes, i.e., querySim, retScore also senPos, obtained high weights as they are effectively symbolized the significance of query provisions inside a document. It is compelling that TS as well as Fuzzy-Set obtained higher weights compared to other temporal attributes; however they used publication time rather than the content time of a report, although TS did not take time uncertainty. Furthermore,

| Feature | $\hat{x}_i$ | $w_i$ | $add_1$ | $add_2$ | remove |
|---|---|---|---|---|---|
| *retScore* | .49 | 1.65 | 0.00 | 0.00 | -0.25 |
| *querySim* | .52 | 6.29 | 6.45 | 5.50 | 4.55 |
| *Title* | .05 | -0.94 | 0.77 | 0.85 | 0.00 |
| *titleSim* | .08 | -0.77 | 0.95 | 0.78 | 1.16 |
| *senPos* | .74 | 1.60 | 1.93 | 0.50 | 1.05 |
| *senLen* | .64 | -0.66 | 2.78 | 1.80 | 1.88 |
| *cntSenSubj* | .02 | 0.08 | 0.12 | -0.05 | 0.00 |
| *cntEvent* | .14 | 0.23 | -0.02 | 0.79 | 0.01 |
| *cntEventSubj* | .02 | 0.14 | 0.04 | -0.03 | 0.02 |
| *timeDist* | .19 | 0.27 | -0.12 | 0.38 | -0.03 |
| *tagSim* | .18 | 1.37 | 1.78 | 1.06 | 0.87 |
| *LmT* | .30 | -1.92 | 1.56 | 0.92 | 0.15 |
| *LmtU* | .83 | -0.33 | -0.32 | 0.15 | 0.16 |
| *TS* | .25 | 2.86 | 4.04 | 4.82 | 0.62 |
| *TSU* | .26 | 0.95 | 1.13 | 1.15 | 1.21 |
| *FuzzySet* | .29 | 2.37 | 4.00 | 4.53 | 0.27 |

Table 2: Feature analysis results

the outcomes reveal that $LmT$ as well as $LmtU$ obtained negative weights showing a adversely correlation with the access efficiency.

Being notice the efficiency of specific features, we performed 3 additional tests and determine the enhancement in (%)MAP. Initially, we prepared a ranking system with $SVM^{MAP}$ using only *retScore* also chosen one added feature at every time to notice how the chosen feature leads to a ranking system. A standard in this situation is the model trained utilizing *retScore* just with MAP of 0.483. The column $add_1$ reveals the enhancement in (%) MAP that every feature might generate on its own than the baseline. The top-5 attributes leads in MAP for this evaluation are *querySim*, *TS*, *FuzzySet*, *senLen*, also *senPos*, while incorporating *cntEvent*, *timeDist*, or *LmtU* leads to the less efficiency.

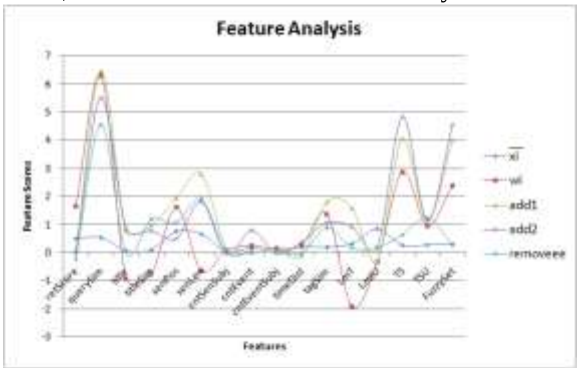


Figure: fig shows the Feature Analysis

We then scrutinized how a single component provided to a ranking system when trained utilizing *retScore* also *another feature class*. You can find two standards in this scenario: 1) the model trained just with *retScore* also every *temporal* attributes with MAP of 0.537, as well as 2) the model trained just with *retScore* also every *entity-based* attributes with MAP of 0.557. The column *add2* reveals the enhancement that all of the entity-based attributes provided to the 1st baseline model, also on another, its demonstrates the enhancement that any of temporal attributes provided to the 2nd model. In overview, the top-2 finest entity-based attributes are *querySim*, *senLen*, also the top-2 finest temporal attributes are *TS* as well as *FuzzySet*.

Lastly, we prepared a ranking system making use of training information that contained many features apart from one at every time to identify how a ranking system is based on that component. The standard is the system trained with many features; also its efficiency (MAP) is 0.617. The column remove reveals the decrease of efficiency than the standard, which is acquired by eliminating every feature. The top-5 attributes that produced a considerable drop in efficiency are querySim, senLen, TSU, titleSim, also senPos.

## 6. Conclusions

In this document, we have suggested a time-aware ranking strategy according to learning-to-rank strategies for temporal concerns. To be able to study the ranking model, we used two classes of attributes derived from annotation information, specifically, entity-based as well as temporal attributes. Using considerable tests we have revealed that the suggested learning-to-rank model substantially enhances the collection efficiency over current time-aware ranking systems.

## 7. Acknowledgment

Our thanks to the management members and principal of Kakatiya Institute of Technology and Science-Warangal who have facilitated resources to read and compute in order to develop this model and narrate this article and our sincere thanks to Head of the Department Prof.P.Niranjan who encouraged us research and publish this paper.

## References

[1] O. Alonso et al. Clustering and exploring search results using timeline constructions. In Proceedings of CIKM'2009, 2009.

[2] K. Berberich et al. A language modeling approach for temporal information needs. In Proceedings ofECIR'2010, 2010.
[3] K. Crammer et al. Online passive-aggressive algorithms. J. Mach. Learn. Res., 7:551-585, 2006.
[4] N. Dai, M. Shokouhi, and B. D. Davison. Learning to rank for freshness and relevance. In Proceeding of SIGIR '2011, 2011.
[5] W. Dakka, L. Gravano, and P. G. Ipeirotis. Answering general time-sensitive queries. In Proceeding of CIKM'2008, 2008.
[6] G. Demartini et al. TAER: time-aware entity retrieval-exploiting the past to find relevant entities in news articles. In Proceedings of CIKM'2010, 2010.
[7] F. Diaz and R. Jones. Using temporal profiles of queries for precision prediction. In Proceedings of SIGIR '2004, 2004.
[8] A. Dong et al. Time is of the essence: improving recency ranking using twitter data. In Proceedings of WWW'2010, 2010.
[9] J. L. Elsas and S. T. Dumais. Leveraging temporal dynamics of document content in relevance ranking. In Proceedings of WSDM'2010, 2010.
[10] A. Jatowt, Y. Kawai, and K. Tanaka. Temporal ranking of search engine results. In Proceedings of WISE'2005, 2005.
[11] T. Joachims. Optimizing search engines using clickthrough data. In Proceedings ofKDD 2002, 2002.
[12] P. J. Kalczynski and A. Chou. Temporal document retrieval model for business news archives. Inf. Process. Manage., 41, 2005.
[13] N. Kanhabua, R. Blanco, and M. Matthews. Ranking related news predictions. In Proceeding of SIGIR '2011, 2011.
[14] N. Kanhabua and K. N0rvag. Determining time of queries for re-ranking search results. In Proceedings of ECDL '2010, 2010.
[15] A. Kulkarni et al. Understanding temporal query dynamics. In Proceedings ofWSDM'2011, 2011.
[16] X. Li and W. B. Croft. Time-based language models. In Proceedings ofCIKM'2003, 2003.
[17] T.-Y. Liu. Learning to rank for information retrieval. Found. Trends Inf.Retr., 3(3):225-331, 2009.
[18] D. Metzler et al. Improving search relevance for implicitly temporal queries. In Proceedings of SIGIR '2009, 2009.
[19] S. Nunes, C. Ribeiro, and G. David. Use of temporal expressions in web search. In Proceedings ofECIR 2008, 2008.
[20] S. Shalev-Shwartz et al. Pegasos: Primal estimated sub-gradient solver for SVM. In Proceedings of ICML '2007, 2007.
[21] M. Shokouhi. Detecting seasonal queries by time-series analysis. In Proceeding ofSIGIR '2011,2011.
[22] Y. Yue et al. A support vector method for optimizing average precision. In Proceedings ofSIGIR '2007, 2007.
[23] T. Zhang. Solving large scale linear prediction problems using stochastic gradient descent algorithms. In Proceedings of ICML 2004, 2004.

## Author Profile

**S.Divya** Currently persuing Master of Technology in Computer Science and engineering with specilization in Softwatre Engineering.Computer Science and Engineering Department, Kakatiya Institute of Technology & Science (KITS), Kakatiya University-Warangal.A.P.,India.

**B.Hanmanthu** obtained his Bachelor's degree in Computer Science and Engineering from JNT University of India. Then he obtained his Master's degree in Computer Science and Engineering with specialization Software Engineering from JNT University Hyderabad, and he is also life member of ISTE. He is currently Assistant Professor of Computer Science and Engineering, Kakatiya Institute of Technology & Science (KITS), Kakatiya University-Warangal. His specializations include Data mining and Data warehousing, Databases and networking.

**B.Raghuram** obtained his Bachelor's degree in Computer Science and Engineering from JNT University of India. Then he obtained his Master's degree in Computer Science and Engineering from Pondicherry Central University Pondicherry, and he is also life member of ISTE. He is currently Assistant Professor of Computer Science and Engineering, Kakatiya Institute of Technology & Science (KITS), Kakatiya University-Warangal. His specializations include Data mining and Data warehousing, Databases and networking.