A Probabilistic Framework for Time-Sensitive Search

MPII at the NTCIR-12 Temporalia-2 Task

Dhruv Gupta*†
dhgupta@mpi-inf.mpg.de

Klaus Berberich†‡
kberberi@mpi-inf.mpg.de

*Graduate School of Computer Science
Saarbrücken, Germany

†Max Planck Institute for Informatics
Saarbrücken, Germany

‡htw saar
Saarbrücken, Germany

ABSTRACT

This research article presents TimeSearch, a probabilistic framework, that competed in the Temporalia-2 task. The subtasks in Temporalia-2 require an information retrieval system to be informed of the temporal expressions (e.g. 1990s) in documents and queries to identify relevant documents. Analysis of these temporal expressions like natural language understanding is challenging. TimeSearch utilizes an unique time model to address these challenges and to understand temporal expressions. Building on this model it identifies interesting time intervals for a given keyword query. These time intervals are then used to rank and diversify documents in a time-sensitive manner. In this article we describe TimeSearch and its performance in Temporalia-2.

Team Name

MPII

Subtasks

Temporal Intent Disambiguation (English)
Temporally Diversified Retrieval (English)

Keywords

Temporal Expressions; Temporal Information Retrieval

1. INTRODUCTION

Temporal Information Retrieval (T-IR) is a field concerned with organization of longitudinal document collections by utilizing the temporal information in them. In order to identify relevant documents for time-sensitive queries, a system must understand the temporal annotations present in the document contents. Naïve approaches which utilize document metadata such as *creation time* or *publication date* may be insufficient to address information needs for time-sensitive queries (e.g. `history of rap`[1]).

Analysis of temporal expressions in document contents is critical for any time-sensitive search engine. However, analysis of temporal expressions is difficult as they can vary with *granularity* (e.g. 1960 versus May 19, 1960) and further they can be highly *uncertain* (e.g. 1960s). Moreover given a *implicitly* time-sensitive query, where the temporal intent is not explicitly specified, prior art which largely relied on the signals derived from publication dates may fail. The temporal expressions again must be accounted for any method that attempts to retrieve or diversify documents by time.

In this article we describe TimeSearch a probabilistic framework that utilizes temporal expressions in document contents to generate interesting time intervals for implicit time-sensitive queries. Further it utilizes the identified time intervals to retrieve and diversify documents along the temporal dimension. The system described is completely *unsupervised* in nature, i.e. it needs no training labels to function.

Advances in the field of T-IR have many benefits for scholars in humanities who need to analyze massive born-digital document collections for anthropological, historical and linguistic trends.

The two subtasks of the Temporalia-2 task that we participated were: *i.* temporal intent disambiguation and *ii.* temporally diversified retrieval.

Temporal Intent Disambiguation subtask required the participants to estimate the likelihood that the query has an information need in the classes : *past, recent, future, and atemporal* given a keyword query. Formally stated as:

> **Problem Temporal Intent Disambiguation**
> Given, classes $C = \{past, recent, future, atemporal\}$ and keyword query q, estimate $P(C|q)$.

Temporally Diversified Retrieval subtask required the participants to identify temporally relevant search results in the class: *past, recent, future*, and *atemporal* from the document collection *Living Knowledge* given a query. Formally, we can state this subtask as follows:

> **Problem Temporally Diversified Retrieval**
> Given, keyword query q and document collection D, estimate $P(d|q, C)$.

System Overview. Our probabilistic framework consists of models constructed from our earlier research [7, 8, 9]. In short, we analyze the statistics of frequently occurring temporal expressions in highly relevant documents given a keyword query. Time is modeled in such a fashion so as to account for its inherent uncertainty [3] (Section 4.1). Using this model we can generate interesting time intervals (in contrast to only time points in prior-art) [7] (Section 4.3). This analysis is carried out at multiple levels of temporal granularity. The time intervals are then used in a time-sensitive language model [3] (Section 4.4) and a time-sensitive diversification algorithm [9] (Section 4.5). The integral components of the system in an overview are depicted in Figure 1.

[1]We present *query keyword / document text* in `teletype` font and *time annotations* in sans serif font.

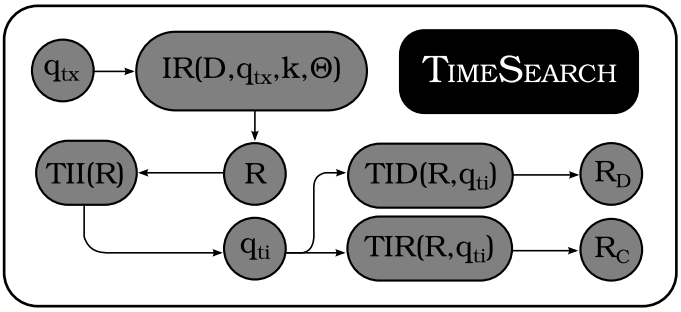

Figure 1: Given a keyword query q_{tx}, TimeSearch uses a pseudo-relevant set of documents R to identify interesting time intervals q_{ti} using the time intervals of interest model (TII). The time intervals q_{ti} are subsequently used for query expansion to obtain temporally diversified set of documents R_D by temporal diversification model (TID). They are also used for retrieving temporally relevant documents R_C using the temporal language model (TIR).

Outline. The article is structured as follows. Section 2 describes the distribution of temporal expressions in the document collection used for the competition and how we preprocess the data for our methods. Section 4 describes the TimeSearch system. In Section 5 we describe our method and its performance in the temporal intent disambiguation subtask. In Section 6 we illustrate our approach and its results for the temporally diversified retrieval subtask. We conclude the report in Section 8 and put our work into context with respect to prior art in Section 7.

2. DOCUMENT COLLECTION: ITS ANALYSIS AND INDEXING

For the subtasks in Temporalia, the *Living Knowledge* [2] Web collection was used. It comprises of news and blogs amounting to approximately 3.8 million documents [11]. The documents are provided with annotations for temporal expressions as well as named-entities.

Temporal Analysis. We did a simple temporal analysis of the *Living Knowledge* document collection; by computing the document frequency of various temporal expressions at year granularity across the collection. The resulting plot is shown in Figure 2.

As can be seen from Figure 2 there is a large number of documents containing temporal expressions in the year "2011" and "2012". Table 1 gives the top-5 frequently occurring years with their relative document frequency to the total number of temporal expression containing documents.

Time	Frequency
2011	0.31
2012	0.25
2010	0.10
2009	0.04
2008	0.04
2013	0.03

Table 1: Top-5 frequently occurring years with their relative frequency in the *Living Knowledge* document collection.

[2] http://livingknowledge.europarchive.org/

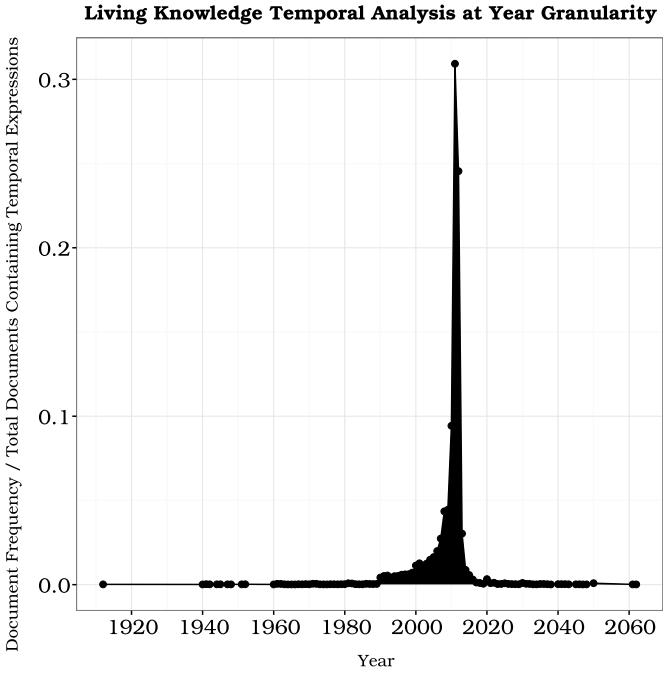

Figure 2: Analysis of the top-100 temporal expressions by document frequency in the *Living Knowledge* document collection.

The kurtosis of the distribution is 421.39 and skewness of 19.85. This highly skewed nature of the distribution affects any probabilistic analysis that is performed on the *Living Knowledge* document collection. We take this background distribution of temporal expressions into account when analyzing time-sensitive queries.

Pre-processing and Indexing. We utilized the temporal expressions provided with corpus for analysis. We did not utilize any external temporal annotator. Each document was pre-processed using Hadoop map-reduce framework to a JSON document representation. The schema used for encoding each document is detailed in Figure 3. We store all the metadata (*category*, *host*, *pubDate*, *url*) along with all the temporal expressions (*allTime*) in the document. We also do not utilize the named entity annotations provided with the document collection.

```
{
  "lk": {
   "mappings": {
     "docs": {
       "properties": {
         "allTime": {
           "properties": {
             "value": {
               "type": "string"
             }
           }
         },
         "category": {
           "type": "string"
         },
         "docId": {
           "type": "string"
         },
         "host": {
           "type": "string"
         },
         "pubDate": {
           "type": "string"
         },
         "source": {
           "type": "string"
         },
         "text": {
          "type": "string"
         },
         "url": {
           "type": "string"
         }
       }
      }
    }
  }
 }
```

Figure 3: JSON schema for a document in our index.

We indexed the documents along with their temporal expressions using the ElasticSearch [3] software. For retrieval of a pseudo-relevant set of documents we used its built-in implementation of the *Okapi BM-25* method with parameters $k_1 = 2.00$ and $b = 1.00$.

3. AN ILLUSTRATIVE EXAMPLE

In order to illustrate the workings of the various models in TimeSearch, consider the following fictitious toy example illustrated in Figure 4.

The likelihood of generating the query q given the document d (document score for the given query) is naïvely measured as a value proportional to the normalized product of term frequency of query terms in document. We will use this as a running example for explaining TimeSearch.

[3] `https://www.elastic.co/`

Query: `summer olympics`		
Id	**Contents**	**Score**
d_1	`summer olympics 2008 took place in beijing, china.`	0.25
d_2	`summer olympics 2012 took place in london, england.`	0.25
d_3	`summer olympic games during 1990s were very competitive.`	0.25
d_4	`summer olympic games during August, 1992 to September, 1992 were very competitive.`	0.25
d_5	`games were competitive during 1973.`	0.17

Figure 4: A toy example for explaining the TimeSearch system. It shows a ordered-set of pseudo-relevant documents returned for the keyword query `summer olympics`.

4. SYSTEM DESIGN

In this Section, we outline the various components that were used to solve the subtasks in Temporalia-2. The entire system was developed in the Java programming language.

4.1 Preliminaries

We consider a document collection D. Each document $d \in D$ consists of a bag of keywords d_{tx} and a bag of temporal expressions d_{ti}. We let $|d_{tx}|$ and $|d_{ti}|$ denote the cardinalities of these bags. Also, let tf(v, d) denote the term frequency of the keyword v, drawn from vocabulary V, in document d.

Let, q_{tx} denote the keywords of the query. To retrieve the pseudo-relevant set of documents R, we utilize a retrieval method:

$$R = \text{IR}(D, q_{tx}, k, \Theta),$$

where D is the document collection, k is the number of top-k results required and $\Theta \in \mathbb{R}^m$ denotes a set of parameters. Each document $d \in R$ is further accompanied by a document score.

4.2 Time Model

To incorporate temporal uncertainty we adopt the time model from [3]. A temporal expression is a four-tuple, $T = \langle b_l, b_u, e_l, e_u \rangle$. Where, $[b_l, b_u]$ & $[e_l, e_u]$, represent the lower and upper bounds on beginning of time interval, b, and its end, e, respectively. Each component of T is drawn from a time domain $\mathcal{T}$ (usually $\mathbb{N}$). A temporal expression T may refer to any time interval $[b, e] \in \mathcal{T} \times \mathcal{T}$ with $b_l \leq b \leq b_u$, $e_l \leq e \leq e_u$, and $b \leq e$. For example the temporal expression *"in the 1960s"* would be represented as $T = \langle 1960-01-01, 1969-12-31, 1960-01-01, 1969-12-31 \rangle$ and time interval such as $[1965-05-10, 1966-04-09]$ can be generated from T. We treat temporal expressions as a set of time intervals and let $|T|$ denote the number of time intervals that $|T|$ may refer to. Figure 5 illustrates the time model with uncertainty. Each element in a temporal expression is computationally represented by its Unix time epoch. That is, equal to the number of seconds elapsed since $01-01-1970$.

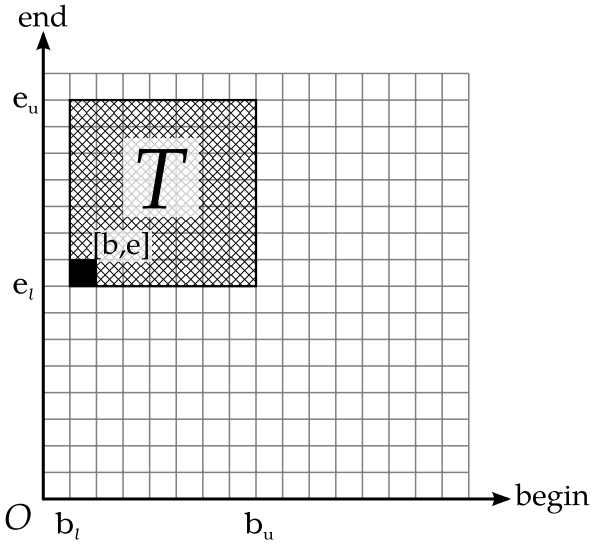

Figure 5: Graphical representation illustrating how a time interval $[b, e]$ is generated from T.

4.3 Time Intervals of Interest

Given a keyword query, we first identify interesting time intervals, which are used for predicting its temporal intent and subsequently for retrieval of documents that satisfy that temporal intent. Time intervals of interest to a given keyword query q_{tx} are identified using our earlier work [7]. The probability that a time interval $[b, e]$ is deemed interesting for a given keyword query q_{tx} is modeled as a two-step generative model:

$$P(\,[b,e] \mid q_{tx}\,) = \sum_{d \in R} P(\,[b,e] \mid d_{ti}\,) \cdot P(\,d \mid q_{tx}\,).$$

The first step involves involves retrieving a pseudo-relevant set of documents R using the keyword query q_{tx}. The probability $P(d|q_{tx})$ thus measures the likelihood of generating the document given the query. This is estimated by using the document scores given by the retrieval method.

In the second step, a time interval $[b, e]$ is in turn generated from each of the temporal expressions in d :

$$P(\,[b,e] \mid d_{ti}\,) = \frac{1}{|d_{ti}|} \sum_{T \in d_{ti}} \frac{\mathbb{1}([b,e] \in T)}{|T|}.$$

Generating time intervals immediately at day granularity is an expensive operation; since it may require discretization of the temporal dimension into hundreds of thousands of days. Subsequently representing each interval in our time model increases the space complexity quadratically. Thus to overcome the problem we apply the generative model recursively to obtain time intervals of interest at year, month, and day granularity. That is, we first identify interesting years; for those years we generate interesting months and subsequently the interesting days in those months.

The time intervals generated for a given keyword query q_{tx} are kept in set q_{ti}. For determining the intent we utilize the time intervals at year granularity.

Let us apply this model to the toy example given in Figure 4 to understand the intuition. Assume all documents are in the pseudo-relevant set i.e. $R = \{d_1, d_2, d_3, d_4, d_5\}$. Since the scores of documents $\{d_1, d_2, d_3, d_4\}$ are larger than d_5; this indicates that the temporal expressions in them have a higher relevance to the temporal intent behind the keyword query.

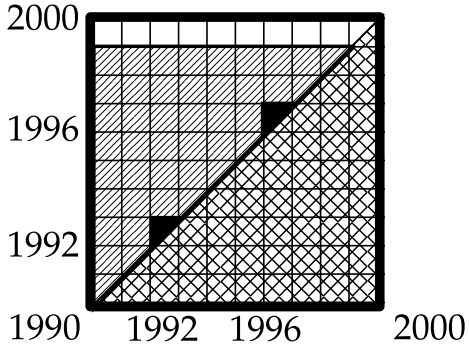

Figure 6: Graphical illustration of how the temporal expressions {1990s; August, 1992; September, 1996} overlap in our time model. The cross-hatched region below the diagonal represents invalid time intervals.

Next all temporal expressions in the documents are represented in our time model and time intervals are generated. Consider the interesting case of {1990s; August, 1992; September, 1996}. The time interval $[08-1992, 09-1996]$ gets a higher relevance to the query then the other temporal expressions present due to high redundancy. Thus the top three time intervals will be $\langle[08-1996, 09-1992]$, $[2008, 2008], [2012, 2012]\rangle$, with $[2008, 2008]$ and $[2012, 2012]$ having equal likelihood (but less than $[08-1996, 09-1996]$) of being generated from the documents.

4.4 Temporal Language Model

Having obtained a set of interesting time intervals q_{ti}, we use them for retrieving documents that lie in these time intervals as explained next.

Given a query q_{tx}, we utilize the approach previously developed by Berberich et al. [3] for re-scoring the documents after expanding it with time intervals from q_{ti}. Assume an independence between generation of textual q_{tx} and temporal intents q_{ti} of query. The probability of generating the query q can be written as [3]:

$$P(q|d) = P(q_{tx}|d_{tx}) \cdot P(q_{ti}|d_{ti}).$$

The first probability $P(q_{tx}|d_{tx})$ can be estimated as described earlier using document scores. The second probability gives the likelihood of generating a time interval of interest $[b, e] \in q_{ti}$ from the document's temporal expressions d_{ti} as follows:

$$P(q_{ti}|d_{ti}) = \prod_{[b,e] \in q_{ti}} P([b,e] \mid d_{ti}).$$

To understand this model, lets return to the toy example. The current document ranking is $\{d_1, d_2, d_3, d_4, d_5\}$. Let the query `summer olympics` be expanded with the time interval $[08-1996, 09-1996]$. Since documents $\{d_3, d_4\}$ both have temporal expressions that contain the highly interesting time interval $[08-1992, 09-1992]$ they will be promoted up in the rankings; followed by $\{d_1, d_2\}$; and finally with $\{d_5\}$ at the end with no temporal expression that mentions a interesting time interval. Thus the final ranking with this model will be $\{d_3, d_4, d_1, d_2, d_5\}$.

4.5 Temporal Diversification

The set of interesting time intervals q_{ti} can also be used as temporal intents to perform temporal diversification, as described in our work [9]. This is done by adapting the work of Agarwal et al. [1], to maximize the following modified objective function over the temporal intents:

$$\sum_{[b,e]\in q_{ti}} P(\,[b,e]\,|\,q_{tx}\,)\left(1-\prod_{d\in R_D}(1-P(\,q_{tx}\mid d_{tx}\,)P(\,[b,e]\mid d_{ti}))\right).$$

The objective maximizes the probability that *at least* one document from each temporal intent is in the diversified set of results R_D. The importance of the temporal intent is given by the probability $P([b,e]|q_{tx})$ as described earlier. The probability that a document satisfies this temporal intent is estimated by:

$$P(\,q_{tx}\mid d_{tx}\,)\,P(\,[b,e]\mid d_{ti}).$$

Lets apply this model on the toy example. Assume that we require only top-2 documents in our diversified set $|R_D|= 2$. In the set of documents we will see that the pairs $\{d_1, d_3\}$, $\{d_1, d_4\}$, $\{d_2, d_3\}$, and $\{d_2, d_4\}$ are the only optimal sets for the objective function above. This is because each document in them represents *at least* one of the temporal intents. Hence we can choose any one of them as R_D.

5. TEMPORAL INTENT DISAMBIGUATION SUBTASK

Various aspects related to the Temporal Intent Disambiguation subtask, such as the query set description, our method and the metrics used for evaluation are described in this section.

5.1 Query Set

The organizers of the Temporalia-2 task made available to the participants a total of 93 queries in the dry run; consisting of 73 training queries and 20 queries for testing. For the formal run; we were provided 300 queries in total. Each query consists of following fields: *i.* query keywords, *ii.* query issue time and *iii.* and probabilities for four different classes (training data). A sample query with the markup is displayed in Figure 7.

5.2 Method

In the Temporal Intent Disambiguation subtask, we were asked to estimate $P(C \mid q)$ given the classes $C = \{past, recent, future, atemporal\}$. For this we used the probability distribution $P([b,e]|q)$ of unit time intervals at the year granularity $\hat{q}_{ti}$.

For the *atemporal* class, we can compute the probability $P(C = atemporal|q)$ as:

$$P(C=atemporal|q)=\sqrt{|\hat{q}_{ti}|}\max_{[b,e]\in\hat{q}_{ti}}|P([b,e]|q)-P([b,e]|D_{ti})|,$$

which is essentially a *two-sample Kolmogorov-Smirnov test* [21] between the distribution of unit time intervals estimated by our model and the distribution of unit time intervals occurring in the background document collection. It tests whether these two samples were generated by a common distribution. For this we utilized the implementation in *Apache Commons Math 3.6 API* [4].

```
<query>
  <id>033</id>
  <query_string>
        weather in London
  </query_string>
  <query_issue_time>
        May 1, 2013 GMT+0
  </query_issue_time>
  <probabilities>
        <Past>0.0</Past>
        <Recency>0.9</Recency>
        <Future>0.1</Future>
        <Atemporal>0.0</Atemporal>
  </probabilities>
</query>
```

Figure 7: Sample query from TID subtask

For the *past, recent* and *future* class, we utilize the query issue time at year granularity t_{issue}. We specifically look at the orientation of the interesting time intervals with respect to the query issue time to determine whether the temporal intent lies in past, present or future. Thus the probabilities for the different classes are measured as follows:

$$P(C = past|q) = \frac{1}{|\hat{q}_{ti}|}\sum_{[b,e]\in\hat{q}_{ti}} \mathbb{1}(t_{issue} > e),$$

$$P(C = recent|q) = \frac{1}{|\hat{q}_{ti}|}\sum_{[b,e]\in\hat{q}_{ti}} \mathbb{1}(b \le t_{issue} \le e),$$

$$P(C = future|q) = \frac{1}{|\hat{q}_{ti}|}\sum_{[b,e]\in\hat{q}_{ti}} \mathbb{1}(t_{issue} < b).$$

5.3 Metrics

The organizers of the task provided the participants with two metrics *loss* and *similarity* for evaluating the participating systems.

Loss. Loss between two k-discrete probability distribution M and N is measured as:

$$loss = \frac{1}{k}\cdot\sum_{i=1}^{k} \mid m_i - n_i \mid$$

Similarity. Similarity between two k-discrete probability distribution M and N is measured by treating each distribution as a vector in k-dimensional space and computing the cosine of the angle between them.

$$sim = \frac{\sum_{i=1}^{k} m_i \cdot n_i}{\sqrt{\sum_{i=1}^{k} m_i^2}\cdot\sqrt{\sum_{i=1}^{k} n_i^2}}$$

The *loss* and *similarity* metrics are computed per query and then averaged to obtain the final system performance.

[4] https://commons.apache.org/proper/commons-math/apidocs/org/apache/commons/math3/stat/inference/KolmogorovSmirnovTest.html

5.4 Results

At each stage of the competition we kept the same system configuration by utilizing the top-1000 documents per query for determining the time intervals of interest.

Training. As an initial experiment we utilized all the 73 training queries for testing our system. This was possible since our method is unsupervised in nature.

A rudimentary baseline for our system can be a uniform distribution assigned to all four classes i.e. $P(C|q) = 0.25$. The results for this training experiment is shown in Table 4. Results for the two systems named MPII-TID-TRAIN and BASELINE corresponding to our method and the baseline respectively are reported in Table 4.

Dry-run. For the dry-run we submitted a single run for the set of 20 queries in the dry-run query set. Our systems performance is reported against the system named MPII-TID-DRY in Table 4.

Formal-run. For the formal-run we again submitted a single run for the set of 300 queries provided for the formal-run. The results for this stage are report against the system named MPII-TID-FORMAL in Table 4.

System	Loss	Similarity	#Queries
MPII-TID-FORMAL	0.35	0.35	300
MPII-TID-DRY	0.34	0.39	20
MPII-TID-TRAIN	0.30	0.48	73
BASELINE	0.26	0.66	

Table 2: Results for our proposed system at different stages of the temporal intent disambiguation subtask.

5.5 Discussion

The proposed method for predicting the probabilities of the temporal classes overall shows poor performance as compared to a naïve baseline. One potential reason we suspect is the temporal bias in the distribution as discussed in Section 2. For most queries the interesting time intervals arose in the time interval $[2011, 2013]$. For queries such as: `uk 2009 balance of payments`, `the advantages of hosting the olympic games`, `freedom of information act`, `when did ww2 start`, `how did bin laden die`, `when was television invented`, `history of slavery`, `occupy wall street movement`, and `susan miller 2012` our system predicted the temporal class distribution with high similarity and low loss. This shows us that given a explicit temporal query or a *history-oriented* query, our method can predict the distribution quite well. However, for queries such as: `naming university buildings with commercial brands`, `body posture alteration`, `dressing code in job interview`, `badminton games`, `advanced english`, and `time warner austin` the predicted distribution deviated from the human-provided distribution. This observation leads us to suspect that comparing the distribution of time intervals of interest with respect to the background distribution for the *atemporal* class may not be correct; no matter how alluring the intuition may be. An alternative approach will be to resort to a *learning* approach whereby the distribution for the *atemporal* class can be induced from the training set of queries.

6. TEMPORALLY DIVERSIFIED RETRIEVAL SUBTASK

This subtask required the participants to retrieve documents for each four of the temporal classes as well a diversified set of documents along time for a given query topic. In this section we describe the query set, our method and the results obtained for our system at various stages of the competition for the temporally diversified retrieval subtask.

6.1 Query Set

The organizers provided us with 8 queries in the dry run and 50 queries in the formal run to evaluate. Each query consisted of the following fields: *i.* query title, *ii.* query description, *iii.* query issue time, *iv.* query subtopic description for *past, recent, future* and *atemporal*. A sample query from this set is shown in Figure 8. As an input to our system we chose only to use the keywords from the *query title* field.

```
<topic>
 <id>002</id>
 <title>
            Junk food health effect
 </title>
 <description>
            I am concerned about the health
            effects of junk food in general.
            I need to know more about their
            ingredients, impact on health,
            history, current scientific
            discoveries and any prognoses.
 </description>
 <query_issue_time>
Mar 29, 2013 GMT+0:00
 </query_issue_time>
 <subtopics>
   <subtopic id="002a" type="atemporal">
            How junk foods are defined?
   </subtopic>
   <subtopic id="002p" type="past">
            When did junk foods
            become popular?
   </subtopic>
   <subtopic id="002r" type="recency">
            What are the latest
            studies on the
            effect of junk
            foods on our health?
   </subtopic>
   <subtopic id="002f" type="future">
            Will junk food continue to be
            popular in the future?
   </subtopic>
 </subtopics>
</topic>
```

Figure 8: Sample query from TDR subtask

6.2 Method

Temporal Ranking of Documents. For retrieval of time-sensitive documents we utilized the temporal language model (outlined in Section 4.4). For retrieving time-sensitive documents we need find q_{ti} for a given keyword query. For the class *recent* we utilized t_{issue} i.e. $q_{ti} = t_{issue}$. For the class *atemporal* the retrieved documents were the same as given by Okapi BM25 retrieval scheme. For classes *past* and *future*, we considered interesting time intervals that lie before t_{issue} and after t_{issue} respectively.

Temporal Diversification of documents was done by considering top-5 identified interesting time intervals as temporal categories for the diversification algorithm.

6.3 Evaluation Metrics

For the evaluation of this task the organizers used standard Cranfield methodology. This is done by first pooling all relevant documents from the submitted runs of participants. Subsequently their relevance grade is identified via online crowdsourcing methods. For temporal ranked documents in specific classes (e.g. *past, recent, future* and *atemporal*), the organizers used nDCG to measure retrieval effectiveness. While for diversified list of documents, α-nDCG and D#-nDCG was used.

6.4 Results

For each stage of the competition we submitted a single run comprising of top-100 documents for each temporal class and for the diversified set of documents. We report the results for metrics that we discussed above for the dry-run and formal-run stage of the competition for our systems.

Category	Dry-run nDCG@20	Formal-run nDCG@20
Atemporal	0.17	0.34
Past	0.19	0.39
Recent	0.05	0.34
Future	0.02	0.34
All	0.11	0.35

Table 3: Results for our proposed system for retrieving time-sensitive documents at different stages of the temporally diversified retrieval subtask.

Stage	nDCG@20	D#-nDCG@20
Dry-run	0.18	0.41
Formal-run	0.33	0.57

Table 4: Results for our proposed system for diversifying time-sensitive documents at different stages of the temporally diversified retrieval subtask.

6.5 Discussion

Concerning the temporal retrieval of documents; from the results we observe that for the dry run stage our system performed very well in the *past* class. However, it did not perform well for the *recent* and *future* classes. On the other hand for the formal run our system performed well for the class *past* and equally well for the rest of the classes.

As for the temporal diversification of documents; our system performed well in formal run stage as compared to the dry run stage.

Overall comparing to organizers system our method did not fare as well. This can be attributed largely to two insights: *i.* the role of the retrieval method for producing an initial set of pseudo-relevant documents and *ii.* the role that document content temporal expressions play in our approach.

In order to improve our system we can attempt to replace the current *Okapi BM-25* method with other state-of-the-art retrieval methods. We also shied away from using any external temporal annotator and opted in favor for the annotations provided with the document collection. The provided annotations had a temporal bias which we discussed in Section 2; which we suspect may be the culprit in our system not performing upto the mark.

7. RELATED WORK

Temporal information retrieval (TIR) is now a well established field of information retrieval which tries to analyze text and the temporal expressions therein. It has received substantial attention given the fact that around 1.5 % of web queries are explicitly time-sensitive in nature [18] and around 7 % of web queries are implicitly time-sensitive in nature [17]. We begin by what types of temporal expressions in text can be identified and which tools can be used to detect them. We then present the relevant prior art for the Temporalia-2 [12] task that belong to two broad classes : *understanding time-sensitive queries* and *time-sensitive document retrieval*. which has practical applications of techniques from temporal information retrieval.

Temporal Expressions in text can be of three types [4]: *explicit*, *implicit*, and *relative*. Explicit temporal expressions are precise notions of time mentioned in language e.g, `December 25, 2014`. However, these explicit temporal expressions may occur at different levels of granularity say, day, month, or year level. For example, `2014` is an expression at year granularity while `December 25, 2014` is at day granularity. Implicit temporal expressions are those which may not immediately be placed in time, e.g, `spring`. Relative temporal expressions may occur *relative* to a temporal expression elsewhere in text e.g, `last year`. Temporal taggers such as SUTime [5] and HeidelTime [19] offer the capability to detect and resolve these temporal expressions to human-interpretable dates.

Understanding Time-Sensitive Queries. One of the early works in temporal query classification was by Jones and Diaz [13]. The authors described a taxonomy for temporal classes; which were *ambiguous, unambiguous* and *atemporal*. Their machine learning approach incorporated signals from the distribution of document publication dates. Some of these features were temporal clarity, kurtosis, and auto-correlation. The first edition of the *Temporalia* competition [11] considered temporal query classification with qualitative set of temporal classes, namely: *past, recency,* and *future*. More recently additional classes have been explored by us [8] and by Kanhabua et al. [14]. Kanhabua et al. [14] study the case of seasonality and periodicity associated with web-queries. They use features acquired from web-query logs, and publication date distribution of an external document collection. In our earlier work [8], we additionally considered the task of disambiguating the temporal

class of a query at multiple levels of granularity and also determining its temporal (a)periodicity. Our approach not only looked at publication dates but also temporal expressions in document contents.

Time-Sensitive Document Retrieval. Berberich et al. [3] presented the *Time-Sensitive Language Model* that answers explicit temporal queries by looking at the temporal expressions in documents. Building on this Kanhabua and Nørvåg [15] look at automatically suggesting years of interest to implicit temporal queries. To this end they utilize publication dates. Work by Dakka et al. [6] relies on publication of documents to improve the retrieval effectiveness by analyzing query-frequency histograms. A recent survey by Campos et al. [4] on temporal information retrieval also noted the lack of any active research in the area of diversifying search results using temporal expressions. In our recent work [9], we presented an algorithm for diversifying search results that utilizes temporal expressions. This approach has been used in the larger system described in this work. An alternative approach would be to anchor documents in time which Jatowt et al. [10] address. They look at the problem of estimating the time period which the document focuses on. They do this by constructing a weighted undirected graph which captures the associations between terms and time.

Applications. Using temporal expressions in documents many interesting research applications have been devised. Swan and Allan [20] investigate how to automatically induce an overview timeline using simple textual features extracted from text. They do this by first constructing a timeline of the corpus at day granularity. They then test the significance of the features e.g. named entities and noun phrases via χ^2 statistic. Alonso et al. [2] specifically look at the temporal information contained in documents to organize and explore them along timelines constructed at multiple granularities. The process first involves creating a timeline outline of documents which considers the temporal expressions in document contents. For this purpose the authors create *temporal document profiles* by using content temporal expressions. Subsequent to this document clustering and re-ranking is performed by utilizing the temporal document profiles. Yeung and Jatowt [16] using temporal expressions and *Latent Dirichlet Allocation* (LDA) study how the past is remembered in text collections. By utilizing topic modelling the authors seek to answer questions such as: *i.* what are significant years and topics? *ii.* what are the events remembered and forgotten? *iii.* and what are historical similarities between countries?

8. CONCLUSION

In this research article we outlined — TimeSearch — a probabilistic framework for time-sensitive search. It understands the temporal uncertainty in time which we leverage to generate time intervals of interest to a given keyword query. These time intervals are then used to retrieve time-sensitive documents or to generate a temporally diverse set of documents. Our methods for the Temporalia-2 task utilize this system in order to identify the temporal intent in *past, recent, future, atemporal* classes and to retrieve time-sensitive documents in those classes.

9. REFERENCES

[1] R. Agrawal, S. Gollapudi, A. Halverson, and S. Ieong. Diversifying search results. In *WSDM 2009*.

[2] O. Alonso, M. Gertz, and R. A. Baeza-Yates. Clustering and exploring search results using timeline constructions. In *CIKM 2009*.

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

[4] R. Campos, G. Dias, A. M. Jorge, and A. Jatowt. Survey of temporal information retrieval and related applications. *ACM Comput. Surv.*, 47(2):15:1–15:41, Aug. 2014.

[5] A. X. Chang and C. D. Manning. SUTIME: A library for recognizing and normalizing time expressions. In *LREC 2012*.

[6] W. Dakka, L. Gravano, and P. G. Ipeirotis. Answering general time-sensitive queries. *IEEE Trans. Knowl. Data Eng.*, 24(2):220–235, 2012.

[7] D. Gupta and K. Berberich. Identifying time intervals of interest to queries. In *CIKM 2014*.

[8] D. Gupta and K. Berberich. Temporal query classification at different granularities. In *SPIRE 2015*.

[9] D. Gupta and K. Berberich. Diversifying search results using time. In *ECIR 2016*.

[10] A. Jatowt, C.-M. Au Yeung, and K. Tanaka. Estimating document focus time. In *CIKM 2013*.

[11] H. Joho, A. Jatowt, and R. Blanco. NTCIR temporalia: a test collection for temporal information access research. In *WWW 2014*.

[12] H. Joho, A. Jatowt, R. Blanco, H. Yu, and S. Yamamoto. Overview of NTCIR-12 temporal information access (temporalia-2) task. In *Proceedings of the 12th NTCIR Conference on Evaluation of Information Access Technologies*, 2016.

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

[14] N. Kanhabua, T. N. Nguyen, and W. Nejdl. Learning to detect event-related queries for web search. In *WWW 2015*.

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

[16] C. man Au Yeung and A. Jatowt. Studying how the past is remembered: towards computational history through large scale text mining. In *CIKM 2011*.

[17] D. Metzler, R. Jones, F. Peng, and R. Zhang. Improving search relevance for implicitly temporal queries. In *SIGIR 2009*.

[18] S. Nunes, C. Ribeiro, and G. David. Use of temporal expressions in web search. In *ECIR 2008*.

[19] J. Strötgen and M. Gertz. Multilingual and cross-domain temporal tagging. *Language Resources and Evaluation*, 47(2):269–298, 2013.

[20] R. C. Swan and J. Allan. Automatic generation of overview timelines. In *SIGIR 2000*.

[21] F. J. Massey. The kolmogorov-smirnov test for goodness of fit. *Journal of the American Statistical Association*, 46(253):68–78, 1951.