L3S at NTCIR-12 Temporalia

Zeon Fernando, Jaspreet Singh, Avishek Anand

singh@l3s.de
Temporal Intent Disambiguation Task (TID)

- **Objective:** To estimate a probability distribution of the query intent across four temporal classes: *past*, *recency*, *future* and *atemporal*.
- We are given a query string and submission date as input.
- We are allowed to use any external resources to complete the task.
Sample Query

<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>

<query>
  <id>035</id>
  <query_string>value of silver dollars 1976</query_string>
  <query_issue_time>May 1, 2013 GMT+0</query_issue_time>
  <probabilities>
    <Past>0.727</Past>
    <Recency>0.273</Recency>
    <Future>0.0</Future>
    <Atemporal>0.0</Atemporal>
  </probabilities>
</query>
Rule Based Voting

- Each rule is made based on a feature. If the rule is obeyed by a temporal class then it is awarded one vote.
- Votes are normalized across the classes to get a probability distribution.
- Features used:
  - Temporal Distance
  - Linguistic Features like verb tense and modality
  - N-grams
Feature Description

► Temporal Features: Temporal Distance directly from the query or using GTE

► Linguistic feature: Verb tense of the predicate verb

► N-gram: Prior probability of unigrams and bigrams per class
Procedure

► Decision tree trained using only n-grams to predict the temporal class of a query.

► Verb tense of the query counts as a single vote. If no verb tense is detected then the atemporal class is awarded a vote.

► Class of the time mention in the query gets a vote.

► If the standard deviation is low then temporal distance is used as the deciding vote.
## Results

<table>
<thead>
<tr>
<th>Run</th>
<th>Average Absolute Loss</th>
<th>Cosine Similarity</th>
</tr>
</thead>
<tbody>
<tr>
<td>RBV</td>
<td>0.2031</td>
<td>0.7307</td>
</tr>
<tr>
<td>N-gram</td>
<td>0.2452</td>
<td>0.6673</td>
</tr>
</tbody>
</table>

Table 1: Evaluation Results of TID Formal Runs
Temporally Diversified Retrieval Task (TDR)

► We have to build temporal retrieval models for each temporal class: *past*, *recency*, *future* and *atemporal*.

► Produce a list of results diversified over the 4 temporal classes

► Given a topic, description and an indicative search question (subtopic) for each temporal class
<topic>
   <id>001</id>
   <title>Earthquakes</title>
   <description>I suspect that these days the intensity of harsh weather conditions such as earthquakes is increased when compared to the past. In order to make sure I need to collect information on earthquake, their past occurrences, and future forecasts, etc.</description>
   <query_issue_time>Mar 29, 2013 GMT+0:00</query_issue_time>
   <subtopics>
      <subtopic id="001a" type="atemporal">What is an earthquake and how severe it can be?</subtopic>
      <subtopic id="001p" type="past">What past earthquakes were most deadly?</subtopic>
      <subtopic id="001r" type="recency">What was the latest earthquake in Asia?</subtopic>
      <subtopic id="001f" type="future">What are predictions regarding the occurrence of earthquakes in the near future?</subtopic>
   </subtopics>
</topic>

Dry Run Queries: 10; Formal Run Queries: 50
Subtopic Classification

Multi-Class SVM - Greedy Selection
Subtopic Classification Features

► **Verb Tense:** Determine the verb tense of the subtopic using the Stanford POS tagger;
- **E.g.:** “was” in “When was the first Olympics held?” is an indicator of past intent
- Subtopics can have *multiple verbs* so we use a parser to determine the main verb, which is the uppermost verb in the parse tree (E.g. “What *were* Apple company’s strategies *behind* the development of iPhone 5?”)

► **Temporal feature:** We compute the *average temporal distance* for the subtopic from the top 20 pseudo relevant documents that were retrieved
Subtopic Classification Features

► **Dictionary feature:** We identified certain words from *dry run queries* which frequently occurred for certain temporal intents and built a *dictionary*.

<table>
<thead>
<tr>
<th>Future</th>
<th>future, forecast, will, would, should, shall, next, expected, soon, projected, possibility, scheduled</th>
</tr>
</thead>
<tbody>
<tr>
<td>Past</td>
<td>past, history, were, origin, did, been, previous, earlier, former, historical</td>
</tr>
<tr>
<td>Recent</td>
<td>recent, present, current, latest, recently, trendy, now, today</td>
</tr>
</tbody>
</table>

► **Verb tense feature**

► **Average Expected Temporal Distance:** top 20 documents
Temporal Relevance of Document

Temporal relevance is used to estimate the *focus time* of a document using the temporal distribution of time expressions in the content:

- Each document \( d \) is annotated with normalized time expressions \((\tau(d))\).
- We map each time expression \( t_e \) in \( d \) to a time interval \([b,e)\) at day granularity (e.g.: May 2014 to \([01/05/2014, 31/05/2014)\)).
- We then determine the temporal distribution of time expressions in a document at month granularity, i.e., each \( d \) has a set \( \tau_m(d) \) of monthly time intervals \([t_{mb}, t_{me})\).
Temporal Relevance of Document

- We have intent specific filters for *recency*, *past* and *future* that are modeled as exponential distributions:

\[
\lambda e^{-\lambda |\text{distance}|} \\
\lambda e^{\lambda |\text{distance}|}
\]

- **Intuition:**
  - temporal expressions close to \( t_q \) have a higher probability than older temporal expressions for *recency* filter,
  - while for *past* and *future* filter temporal expressions further away from \( t_q \) have a high probability.
Temporal Relevance of Document

We use the intent specific filters to transform the temporal distribution of time references in a document as follows:

\[
h(t_e) = \begin{cases} 
  w(t_e) * f(t_e) * |t_q - t_e|, & \text{if } t_e \in \mathcal{P} \& \mathcal{F} \\
  w(t_e) * f(t_e) * \frac{1}{|t_q - t_e|}, & \text{if } t_e \in \mathcal{R}
\end{cases}
\]

The expected distance of the document with respect to the query hitting time, i.e., temporal relevance score is:

\[
E(d) = \frac{1}{|\tau_m(d)|} \sum_{t_e \in \tau_m(d)} h(t_e)
\]
Subtopic Classification

Multi-Class SVM - Greedy Selection
Retrieval Approach

Query

- Subtopic Classification
  - Past RM
  - Recent RM
  - Future RM
  - Atemporal RM

Top-k Results

Diversified Top-k Results

Learning To Rank
- Earth Movers Distance
- Param. Sum
Parameterized Sum Method

► For all our experiments, we determine a set $R$ of pseudo relevant documents ($|R| = 1000$) using the unigram language model with Dirichlet smoothing ($\mu = 2000$).

► Then re-rank the documents using the scores obtained from the linear combination of the temporal relevance and topical relevance score:

$$R_f = \lambda E(d) + (1 - \lambda) R_c, \quad 0 \leq \lambda \leq 1$$

where $\lambda$ is tunable parameter and $R_c$ is relevance score of language model
Learning to Rank Approach
Learning to Rank Features

**Verb Tense:** We split the document into two sentence types: $S_{\text{noun}}$ those that contain at least a noun search term and $S_{\text{non-noun}}$ those that don’t contain any noun search term.

- the ratio of past, present and future tense w.r.t $S_{\text{noun}}$
- the ratio of past, present and future tense w.r.t $S_{\text{non-noun}}$

**Topical Features:** These include four similarity based features using jaccard similarity between:

- search topic and document title
- search topic and document content
- search subtopic and document title
- search subtopic and document content

*Document relevance score* between query and document is also used as a feature.
Learning to Rank Features

**Temporal Features:** These include two features based on the temporal expressions of the document.

- *temporal relevance score* of a document as described previously
- *temporal density score* which is the ratio of the number of temporal expressions to the length of the document. This feature helps differentiate between atemporal and temporal documents
Retrieval Approach

Query

Subtopic Classification

Past RM

Recent RM

Future RM

Atemporal RM

Top-k Results

Diversified Top-k Results

Learning To Rank

Earth Movers Distance

Param. Sum
Earth-Movers Distance for Diversification

The earth mover’s distance is a measure of distance between two probability distributions, that is the minimum cost required to transform one probability distribution to another.

► Intuition. Get a set of documents that have diverse temporal distributions which would in turn give us a temporally diversified set.
► We use candidate document sets $R_i$ from the top 100 documents retrieved for each temporal intent using the above ranking approaches.
► Select documents that greedily maximise earth movers distance ($*\ 1/rank$) between the diversified list and the new document to be added.
Diversified List

Doc 1

Doc 2

Doc 3
Experimental Setup

► We used Lucene to build the index for the “LivingKnowledge news and blogs annotated subcollection” corpus which comprised of 3.8 million documents.

► The query is constructed from the topic and subtopic, and then searched against the title and content fields of the documents.

► Training data.
  - 10 dry run topics and 50 formal run topics of previous years task along with the qrels are used to generate the training set for the learning to rank approaches.
Experimental Setup

► We created separate training datasets for each temporal class as follows:

- The constructed query is used to retrieve top 1000 pseudo relevant documents using the language model,

- From this we select, relevant and irrelevant documents in the ratio 1:2. The relevance judgments in the *qrels* are of the order 2 (really relevant), 1 (relevant) and 0 (irrelevant).

- Each temporal class-specific training dataset is used to learn a ranking model so as to predict document ranking for a unseen subtopic of the same class.

- List-wise L2R with AdaRank (RankLib)
## Results

<table>
<thead>
<tr>
<th>Run</th>
<th>Atemporal</th>
<th>Future</th>
<th>Past</th>
<th>Recency</th>
<th>All</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manual L2R</td>
<td>0.7264</td>
<td>0.6511</td>
<td>0.7005</td>
<td>0.7151</td>
<td>0.6983</td>
</tr>
<tr>
<td>Auto Param. Sum</td>
<td>0.6109</td>
<td>0.6932</td>
<td>0.7127</td>
<td>0.6758</td>
<td>0.6731</td>
</tr>
<tr>
<td>Auto L2R</td>
<td><strong>0.7299</strong></td>
<td>0.6508</td>
<td>0.6998</td>
<td>0.7116</td>
<td>0.6980</td>
</tr>
<tr>
<td>Auto LM</td>
<td>0.7052</td>
<td><strong>0.7151</strong></td>
<td><strong>0.7297</strong></td>
<td>0.6865</td>
<td><strong>0.7076</strong></td>
</tr>
</tbody>
</table>

<p>|   | P@20     |
|---|----------|----------|--------|---------|--------|
|   | Atemporal | Future  | Past   | Recency | All    |
|   | 0.7960    | 0.7360  | 0.7710 | 0.7970  | 0.7750 |
|   | 0.7330    | 0.7790  | <strong>0.8000</strong>| 0.7760  | 0.7720 |
|   | 0.7960    | 0.7360  | 0.7700 | 0.7930  | 0.7737 |
|   | 0.7690    | <strong>0.7850</strong>| 0.7940 | 0.7580  | 0.7416 |</p>
<table>
<thead>
<tr>
<th>Run</th>
<th>D#-NDCG@20</th>
<th>I-rec@20</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manual L2R</td>
<td>0.8262</td>
<td>0.9850</td>
</tr>
<tr>
<td>Auto Param. Sum</td>
<td>0.6852</td>
<td>0.9900</td>
</tr>
<tr>
<td>Auto L2R</td>
<td>0.8423</td>
<td>0.9850</td>
</tr>
</tbody>
</table>
Take-Aways

- Joint classification approach for subtopic classification does well.

- We are good at recency and atemporal!

- The $nDCG@20$ performance for the atemporal class is poor for parameterized sum method when compared to learning to rank by about 19%.

- The $nDCG@20$ performance for the future intent is higher using parameterized sum method than using learning-to-rank approach by about 7%.

- All our models perform better than the baseline in terms of rank insensitive metric $P@20$. However, in terms of $nDCG@20$ the performance for future and past class of the baseline outperforms all our models.
Questions?

singh@l3s.de
@movedthecheese