L3S at the NTCIR-12 Temporal Information Access (Temporalia-2) Task

Zeon Trevor Fernando
L3S Research Center, Germany
fernando@l3s.de

Jaspreet Singh
L3S Research Center, Germany
singh@l3s.de

Avishek Anand
L3S Research Center, Germany
anand@l3s.de

ABSTRACT

This paper describes our participation in the NTCIR-12 Temporalia-2 task including Temporal Intent Disambiguation (TID) and Temporally Diversified Retrieval (TDR) subtasks. In the TID subtask, we extract linguistic features from the query, time distance features and multinomial distribution of the query n-grams which are then combined using a rule based voting method to estimate a probability distribution over the temporal intents. In the TDR subtask, we perform temporal ranking based on two approaches, linear combination of textual and temporal relevance method, and learning to rank method. Three classes of features comprising of linguistic, topical and temporal features were used to estimate document relevance in the learning to rank approach.

Team Name

L3S

Subtasks

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

Keywords

temporal information retrieval, learning to rank, temporal intent disambiguation, text classification

1. INTRODUCTION

Temporal information retrieval is a sub-branch of information retrieval concerned with improving retrieval effectiveness by leveraging temporal information found in documents and queries [3, 1, 9]. Previous studies have shown that nearly 1.5% of all queries issued explicitly mention time expressions [14], while about 7% of web queries have an implicit temporal intent [13]. In the TID (Temporal Intent Disambiguation) subtask, the objective is to estimate a probability distribution of the query intent across four temporal intent classes: *past*, *recent*, *future* or *atemporal*. For example, the query "history of rap" implies a high probability should be assigned to the *past* intent. The temporal intent disambiguation of queries is then useful when searching in longitudinal document collections for selecting the appropriate temporal retrieval model [2, 10, 11]. In the TDR (Temporally Diversified Retrieval) subtask, teams must devise retrieval models to produce a ranked list of documents that are diverse across all of the above defined temporal intents for a given query. For a more detailed overview of the subtasks of NTCIR-12 Temporalia-2 task please refer to [8].

For the TID subtask, we identify a set of query-specific features such as verb tense of the query, distance between the temporal expressions identified in the query topic and the query hitting time, and the frequently occurring n-grams for each temporal intent using a multinomial distribution. We combine them using intent specific rules that if satisfied contribute a vote to the respective intent class. After applying all the rules, votes are aggregated and normalized to determine the distribution across the temporal intents.

In the TDR subtask, our approach is to first classify query subtopics to the correct temporal intent class with high accuracy using a joint classifier (Section 3.1). Using the classified subtopics in conjunction with the query topic, we produce a ranked list for each temporal intent class using retrieval models trained by a *listwise* learning-to-rank approach. One of the key features we use is the temporal relevance of a document. Temporal relevance is a score that estimates the expected temporal distance of the document from the query hitting time using the distribution of temporal expressions in the content (refer Section 3.2). Finally to produce a diversified list from the top-k results for each class, we use a greedy approach that maximizes the earth mover's distance between the distribution of temporal references in the result set.

Outline. The rest of the paper is organized as follows. In Section 2 we describe our approach for the TID subtask along with the discussion of the results detailed in Section 2.3.2. In Section 3 we first describe our subtopic classification approach in 3.1 and then explain how temporal relevance of a document is computed in 3.2. We then highlight the features used for the learning-to-rank models in Section 3.4. Our diversification approach using earth mover's distance is the subject of Section 3.5. In Section 3.6 we discuss the experimental setup including the training procedure for the learning-to-rank models. Finally we discuss our performance in the TDR subtask in Section 3.6.2 and highlight key takeaways in Section 4.

2. TEMPORAL INTENT DISAMBIGUATION

In the TID subtask, given a query (q) and query submission date (t_q) we have to estimate a distribution across four temporal intent classes (*atemporal, past, recent, future*). Thus to estimate this distribution, we use a rule based voting method that comprises of intent specific rules designed from the dry run queries. The rules use various query-specific features such as verb tense, distance between the date in the

query and query submission time and the multinomial distribution of the n-grams extracted from the queries. In this section, we first describe how the features are extracted from the queries and then the approach of how the rule based voting method is used to build the required distribution across the temporal classes.

2.1 Features extracted for TID

- **Time Distance Features**: A temporal mention in a query is a good feature to help disambiguate the temporal intent of a query. For example, "French Open 2012" becomes a strong indicator of *past* intent given that t_q is "May 1, 2013 GMT+0". We use the SUTime Library [5] available part of the Stanford CoreNLP pipeline to recognize and normalize temporal expressions in the queries. The distance of the normalized time expressions from the query hitting time is measured to provide an estimate of the intent.

  In the dry run queries only 16 out of 100 contained time expressions that can be used to estimate the time distance, this reflects how rare it is to find explicit or implicit temporal expressions in a search query. Thus in order to obtain more candidate temporal expressions for the formal and dry run queries, we used the freely accessible GTE[1] web service detailed in [4]. Given a query, the GTE web service returns a set of candidate years extracted from the *top 50 web snippets* returned by the Bing Search API.

- **Linguistic Features**: Verb tense is a strong temporal indicator for search queries [19]. For example, the verb "was" in "When *was* the first Olympics held" is a strong indicator of *past* intent. We use the Stanford Part-of-Speech Tagger library [16] that recognizes verbs along with the tenses in a sentence using the Penn Treebank tag set. Verb tenses included in the Penn Treebank set include past tense, past participle, present tense, present participle and base verb form. Since the Penn Treebank set doesn't include any tags for future tense, we consider a *base* verb that is preceded by a *modal* verb such as *will/shall* to be an indicator of *future* tense. A query can include multiple verbs with different tenses. Thus we do a syntactic parsing of the query using the Stanford Parser Library [15] to determine the main predicate, by selecting the uppermost verb in the parse tree.

- **NGram Features**: As a set of baseline features, we extract the uni-gram and bi-gram terms of the queries from the training data, which is 73 dry run queries. We model the per class multinomial distribution of the *n-grams* by using the n-grams overall frequency (T) and the per class n-gram count (C). The per class n-gram count is computed by counting the n-grams per class (like past) generated from those queries which have a non-zero past probability in the training data.

$$p(ng, class) = \frac{C}{T} \tag{1}$$

  Each formal run query (q) is represented as a set of all possible permutations of the uni-gram and bi-gram terms that are extracted from the query. Then the probability distribution of a query across the temporal classes is calculated using the following equation.

$$p(q, class) = \arg\max_{i \in q} \prod_{ng \in i} p(ng, class) \tag{2}$$

[1] http://www.ccc.ipt.pt/ ricardo/software.html

2.2 Rule Based Voting Method

We designed rules based on the dry run queries, which if satisfied contributes a vote to a particular temporal class. Once all the rules are applied, the votes are accumulated for each temporal class and a probability distribution across the temporal classes are built using these votes. We apply the rules in the same order as defined below.

- NGram features provides the probability of a query to a particular temporal class. We learn a decision tree model based on regression for the dry run queries using the NGram features. This learned model is then used to predict the probabilites for the formal run queries. The temporal class with the maximum probability output from the learned model gets a vote.

- From the verb tense features, if the tense of a query is either past, present or future, a vote is assigned to that respective temporal class [7]. If no tense information can be extracted from the query, a vote is assigned to the atemporal class.

- If the temporal mention extracted from the query contains the words (*"past_ref", "present_ref" or "future_ref"*), a vote gets assigned to that temporal class. If the temporal mention is a date and if its earlier or after the query hitting time, a vote gets assigned to past and future class respectively.

- If after applying the above rules, the votes across all the temporal classes are low and their standard deviation is low. We then take the mean of the candidate years extracted for the query from the GTE service. If the difference between the mean and the query hitting time is about 1 year in the past, we give a vote to recent. If the difference is more than 2 years, a vote is assigned to the past. If the mean date is after the query hitting date, a vote is given to future. If there are no candidate years, atemporal gets a vote.

2.3 Experiments

2.3.1 Temporal Intent Disambiguation Runs

We submitted 2 runs for the TID subtask.

- L3S-TID-E-1: This run uses the rule based voting method to generate the probability distribution across the temporal classes for a given query.

- L3S-TID-E-2: This run uses the probability distribution across temporal classes for a query generated only from the NGram features.

2.3.2 Results and Discussion

The evaluation results in table 1, show that the rule based voting method does better than the baseline which uses only NGram features across both performance measures. This highlights how the linguistic and time distance features helps

Run	Average Absolute Loss	Cosine Similarity
L3S-TID-E-1	**0.2031**	**0.7307**
L3S-TID-E-2	0.2452	0.6673

Table 1: Evaluation Results of TID Formal Runs

improve the estimation of the temporal intent of a query. Our method helps determine the correct distribution for queries that contain linguistic information such as *"what was the great awakening"* and *"when did elvis die"*. However, for queries that have an implicit/explicit temporal expression identified, such as, *"uk 2009 balance of payments"* or *"John Galliano Spring 2011 Mens Preview"* the probabilities are spread across all temporal classes with higher probability for the past intent. This happens since we aggregate the votes for time distance only after accumulating the votes for NGram and linguistic features. Consequently, this increases the average absolute loss because these type of queries clearly indicate a particular temporal intent, thus for queries that contain temporal expressions only the time distance rule should be applied.

Using a rule, based on dictionary words such as *"future", "schedule", "releases", "forecast", "plan"* could help estimate queries that have higher probability of future intent. Certain queries indicating future intent were misclassified as past intent because of the ambiguity introduced by the temporal tagger. For example, *"November Calendar printable"* and *"December calendar"* were resolved to "2012-11", "2012-12" respectively.

The probability for atemporal queries like *"causes of global warming", "light pollution", "literature critics"*, were easily estimated as all the features contributed votes to the atemporal intent and also no candidate years were returned by the GTE service indicating atemporal nature. However, for atemporal queries like *"consumer economy", "sherlock holmes"*, the distance of the mean of the candidate years returned by GTE service indicates past, which causes a misclassification. Thus we should re-design the rule to also consider the spread of candidate years across time to understand the atemporal intent. Also the candidate years returned by the GTE service is based on search results returned in December 2015, which won't be optimal for queries with issuing time of 2013.

Queries like *"how long does the flu last", "the advantages of hosting the olympic games"* are considered to be either past or recent using the NGram and linguistic features, in these cases we need additional rules to help disambiguate the atemporal intent.

3. TEMPORALLY DIVERSIFIED RETRIEVAL

In the TDR subtask, we are given a topic, description and an indicative search question (subtopic) for each temporal class and we have to retrieve a list of relevant documents for each of these classes. We also have to return a list of documents that are temporally diverse for the same topic. Since the subtopic class information was not supposed to be used, we use a classifier to jointly classify the subtopics to the respective intents based on verb tense and dictionary features detailed in Section 3.1. Once we have a classified subtopic, we use it for retrieval which is based on a learning-to-rank approach using features extracted from verb tenses of sentences in documents, topical similarity of the document, textual relevance returned by statistical language model and temporal features based on the distribution of time references in the document with intent specific filters.

Documents provide temporal information in two forms, publication date and temporal expressions in the content. Thus in Section 3.2, we describe how to compute the temporal relevance score of a document using the temporal expressions in the content. We then shortly describe in Section 3.3, how this temporal relevance score is combined with the topical relevance score in a parameterized sum. Next, we focus on the features extracted from the document that are used in learning-to-rank approaches to build different ranking models for each temporal intent (Section 3.4). Finally, we describe in Section 3.5 the approach for producing the diversified set of documents across all temporal intents that uses the set of documents considered relevant for each temporal intent as candidates.

3.1 Subtopic Classification

For each search topic, the workload also contains subtopics which are indicative search questions for each of the temporal classes. We use a multiclass SVM classifier[2] to classify subtopics using the features extracted from it. Using this classifier there is a likelihood that subtopics belonging to the same topic will be classified to the same temporal intent. Thus we use the confidence score returned from the SVM classifier to jointly classify the subtopics for a given topic using a greedy approach.

In this approach we first pick the subtopic-intent pair that has the maximum confidence score, and then ignore the confidence scores of this intent for the remaining subtopics. Then we pick the next highest confidence score pair from the remaining set and proceed as above to determine all unique subtopic-intent pairs.

The features that are used for the multiclass classifier are listed below.

- We extract the tense of the subtopic similar to the approach described in section 2.1 which is used for the TID subtask.
- We identified certain words from the dry run queries which were frequently occurring for certain temporal intents and built a word dictionary. We also included synonyms of the above identified words and the class names as well.
- We compute the average expected distance (section 3.2) for the subtopic from the top 20 pseudo relevant documents that were retrieved.

3.2 Temporal Relevance Score of Document

The temporal relevance estimates the expected distance of a document in time, that is, the focus time using the temporal distribution of time expressions in the content. Thus in this section, we describe how we compute this temporal relevance using the annotated normalized time expressions $\tau(d)$. We map each time expression t_e in the document d to a time interval $[b, e)$ at day granularity (e.g., May 2014 is mapped to [01/05/2014, 31/05/2014]). Then we determine the temporal distribution of time references in a document at monthly granularity, that is, each document d is represented by a set $\tau_m(d)$ of monthly time intervals $[t_{mb}, t_{me})$.

[2]https://www.cs.cornell.edu/people/tj/svm_light/

The weight of each monthly time interval is calculated as follows

$$w([t_{mb}, t_{me})) = \sum_{t_e \in \tau(d)} \begin{cases} \frac{cnt(t_e)}{|\tau(d)|}, & t_e \in [t_{mb}, t_{me}) \\ \frac{cnt(t_e)}{|\tau(d)|} \cdot g, & [t_{mb}, t_{me}) \in t_e \in \{t_g\} \end{cases} \quad (3)$$

where $cnt(t_e)$ is the count of the number of occurrences of t_e in the document. The constant t_g is used to describe if a time interval is at a yearly granularity and the constant g is the weight contributed by this temporal expression to the time interval $[t_{mb}, t_{me})$(i.e.$\frac{1}{12}$).

The intent specific filter ($\mathcal{I}$) for recency ($\mathcal{R}$), past ($\mathcal{P}$) or future ($\mathcal{R}$) are chosen based on the subtopic classification. It is modeled as an exponential distribution using $dist$ as the distance between query issue time and time expression measured in months ($t_q - t_e$):

$$f(t_e) = \begin{cases} \lambda e^{-\lambda|dist|} \cdot \mathbb{1}(dist \geq 0), & if\ \mathcal{I} = \mathcal{R} \\ \lambda e^{\lambda|dist|} \cdot \mathbb{1}(dist > 0), & if\ \mathcal{I} = \mathcal{P} \\ \lambda e^{\lambda|dist|} \cdot \mathbb{1}(dist < 0), & if\ \mathcal{I} = \mathcal{F} \end{cases} \quad (4)$$

where $\mathbb{1}(dist \geq 0)$ is an indicator function whose value assumes 1 iff $dist \geq 0$. λ is a tunable parameter, which is set to 0.03 in our experiments. The intuition behind this approach is that temporal expressions close to t_q have a higher probability than older temporal expressions for the recency filter. While for the past and future filter, temporal expressions further away from t_q have a higher probabilty.

We then use the chosen intent specific filter to transform the temporal distribution of time references in a document.

$$h(t_e) = \begin{cases} w(t_e) \cdot f(t_e) \cdot |dist|, & if\ \mathcal{I}\ = \mathcal{P}\ or\ \mathcal{F} \\ w(t_e) \cdot f(t_e) \cdot \frac{1}{|dist|}, & if\ \mathcal{I}\ = \mathcal{R} \end{cases} \quad (5)$$

So the expected distance of the document with respect to the query hitting time, i.e., temporal relevance score is computed as follows

$$E(d) = \frac{1}{|\tau_m(d)|} \sum_{t_e \in \tau_m(d)} h(t_e) \quad (6)$$

3.3 Parameterized Sum Method

For all our experiments, we determine a set R of pseudo-relevant documents ($|R|$=1000) by employing a unigram language model with Dirichlet smoothing ($\mu = 2000$) [20]. We then re-rank the documents using scores obtained from the linear combination of the temporal relevance and topical relevance score, defined as follows

$$R_f = \lambda E(d) + (1 - \lambda)R_c, \quad 0 \leq \lambda \leq 1 \quad (7)$$

where λ is a tunable parameter and R_c is the relevance score of the language model.

3.4 Learning-To-Rank Features

In our approach we use a *listwise* learning-to-rank algorithm optimized for the evaluation measure $nDCG$@20. For a detailed description of the different approaches, refer to [12]. Feature selection is critical for learning-to-rank approaches, so in this section we describe the various features that we extract from the query-document pairs.

- **Verb Tense Features**: We take the noun terms of the search query into consideration and split the document into two sentence types: $\mathcal{S}_{noun}$ those sentences that contain atleast a noun search term and $\mathcal{S}_{non-noun}$ those that don't contain any noun search term. We determine if a sentence talks about the past, present or future by using the same approach as the linguistic feature for TID subtask (Section 2.1). We thus have 6 verb tense features:
  - the ratio of past, present and future tense w.r.t $\mathcal{S}_{noun}$,
  - the ratio of past, present and future tense w.r.t $\mathcal{S}_{non-noun}$.

  These features help determine the language of the document, how much of the text talks about the past, present or future which in turn helps match it to a particular temporal intent.

- **Topical Features**: These include 4 similarity based features using the jaccard similarity on a word level between:
  - search topic and document title,
  - search topic and document content,
  - search subtopic and document title,
  - search subtopic and document content.

  These features help determine the topical similarity between the document, search topic and subtopic. We also use the document relevance score between the search query and the document as a feature. The relevance score obtained from the unigram language model with Dirichlet smoothing is directly used.

- **Temporal Features**: These include *two* features based on the temporal expressions of the document.
  - temporal relevance score computed for a document as described in Section 3.2.
  - temporal density feature which is the ratio of the number of temporal expressions to the length of the document. This helps differentiate between atemporal and temporal documents.

3.5 Earth Mover's Distance for Diversification

The earth mover's distance ($\mathcal{E}$) is a measure of distance between two probability distributions, it is the minimum cost required to transform one probability distribution to another. In our case, we measure the $\mathcal{E}$ between the temporal distribution of time references from one document to another. We use $\mathcal{E}$ for diversification, so that we get a set of documents that have diverse temporal distributions which would in turn give a temporally diversified set. We consider sets R_i containing candidate documents from the top 100 documents retrieved for the specific intent ($i \in \{atemporal, recency, past, future\}$) using the above ranking approaches. We represent the diversified set of results as R_D, to which we add documents from sets R_i that have maximum $\mathcal{E}$ from the documents already present in R_D. During the initial step, we add a $rank$ 1 document from one of the sets R_i at

random and initialize $\mathcal{E}_0$ with a zero value. Then we compute the $\mathcal{E}$ between two temporal distributions A and B as follows

$$\begin{aligned} \mathcal{E}_{i+1} &= (w_A(te_i) + \mathcal{E}_i) - w_B(te_i) \\ \mathcal{E}_{Total} &= \sum_{i=1}^{|\tau_m(A) \cup \tau_m(B)|} |\mathcal{E}_i| \end{aligned} \tag{8}$$

We give preference to top ranked documents to be added to R_D by discounting the $\mathcal{E}_{Total}$ using the rank of the document, so the final score that we consider is

$$\mathcal{E}_f = \frac{1}{rank} * \mathcal{E}_{Total} \tag{9}$$

3.6 Experimental Setup

We used Lucene[3] to build the index for the "LivingKnowledge news and blogs annotated subcollection" corpus [8]. The unigram language model with Dirichlet smoothing implementation of Lucene was used to retrieve the top 1000 pseudo-relevant documents. The query is constructed from the title of the topic and the subtopic, and then searched against the title and content fields of the documents. The features described in section 3.4 are extracted from the tagged version of a pseudo-relevant document.

Training data. The 10 dry run topics and the 50 formal run topics of Temporalia-1 along with their *qrels*[4] are used to generate the training data for learning the ranking models. Each row in the training data is a query-document pair: the first column is the relevance judgement, the second is query id (qid) used to restrict the generation of constraints, the subsequent columns are feature/value pairs ordered by increasing feature number. We created separate training datasets for each temporal class, the data is prepared as follows: the query containing topic and subtopic is used to retrieve the top-1000 pseudo relevant documents using a language model (LM). The relevance judgements in the *qrels* are of the order 2 (*really relevant*), 1 (*relevant*) and 0 (*irrelevant*). From the pseudo relevant documents, we selected relevant and irrelevant documents in different ratios (1:1, 1:2, 1:3) in order to find the right balance of training examples. Finally we found that the ratio 1:2 (relevant: irrelevant) for preparing the training data performs best. Each temporal class-specific training data is then used to learn a ranking model so as to predict document ranking for a formal-run subtopic of the same temporal class. Besides, we also experimented by combining all the class-specific training data into one large training set, but the performance was lower than the class-specific training data.

Ranking Models. The RankLib[5] library is used to compare different *listwise* learning to rank approaches, such as, AdaRank [18], RankBoost [6] and LambdaMART [17]. We used the default learning algorithm specific parameters and optimize for the measure *nDCG*@20. For the final runs, we used the AdaRank learning algorithm.

3.6.1 Temporal Diversified Retrieval Runs

We submitted 3 runs for the TDR subtask. For all the runs the diversification of the results across all temporal intents is carried out using the earth mover's distance measure.

- L3S-TDR-E-1: Manual run with manually crafted queries using the topic and subtopic. The training and test data (formal runs) is generated from the pseudo-relevant documents retrieved using LM. The ranking model is learned based on the class-specific datasets.
- L3S-TDR-E-2: Automatic run in which the subtopics are classified using the joint classifier described in Section 3.1. The pseudo relevant documents are retrieved using LM and then re-ranked using the parameterized sum method (Section 3.3). The parameter λ is set to 0.3, giving more weightage to the textual relevance score.
- L3S-TDR-E-3: Automatic run in which the subtopics are classified using the joint classifier described in Section 3.1. The training and test data (formal runs) is generated from the pseudo-relevant documents retrieved using LM. The ranking model is learnt based on the class-specific datasets.
- LM: In this run documents are retrieved using LM.

3.6.2 Results and Discussion

Run	D#-NDCG@20	I-rec@20
L3S-TDR-E-1	0.8262	0.9850
L3S-TDR-E-2	0.6852	**0.9900**
L3S-TDR-E-3	**0.8423**	0.9850

Table 2: Diversified Results of TDR Formal Runs

The evaluation results in table 3 shows that (1) there is no significant difference in overall performance between the manual run (L3S-TDR-E-1) and automatic run with subtopic classification (L3S-TDR-E-3) which indicates that the joint classification approach for subtopic classification performs well. (2) The *nDCG@20* performance for the *atemporal* class is poor for parameterized sum method when compared to the learning-to-rank approach by about 19%. This shows that the topical features based on similarity measures along with the temporal density feature helps significantly retrieve more relevant atemporal documents. (3) The *nDCG@20* performance for the *future* intent is higher using parameterized sum method than using learning-to-rank approach by about 7%, this indicates that using topical features based on similarity measures and verb-tense features have a deterimental effect than only using the temporal relevance score when retrieving documents for future intent. (4) 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, indicating the importance of textual relevance in retrieving such documents. Our learning-to-rank models outperform the baseline in *nDCG@20* for the *atemporal* and *recency* class, highlighting the significance of temporal features such as temporal relevance and temporal density for these classes.

The overall *ERR* measure is high, indicating that the 1st and 2nd ranked document is relevant for most topics across all runs. The topic "The right to be forgotten" performs poorly for *ERR* and *nDCG@20* measures across all subtopics since we apply stopword removal to the topics before forming a query (e.g."right forgotten"), which when used

[3]https://lucene.apache.org/core/
[4]http://research.nii.ac.jp/ntcir/permission/ntcir-11/perm-en-Temporalia.html
[5]https://sourceforge.net/p/lemur/wiki/RankLib/

Run	NDCG@20					P@20				
	Atemporal	Future	Past	Recency	All	Atemporal	Future	Past	Recency	All
L3S-TDR-E-1	0.7264	0.6511	0.7005	**0.7151**	0.6983	**0.7960**	0.7360	0.7710	**0.7970**	**0.7750**
L3S-TDR-E-2	0.6109	0.6932	0.7127	0.6758	0.6731	0.7330	0.7790	**0.8000**	0.7760	0.7720
L3S-TDR-E-3	**0.7299**	0.6508	0.6998	0.7116	0.6980	0.7960	0.7360	0.7700	0.7930	0.7737
LM	0.7052	**0.7151**	**0.7297**	0.6865	**0.7076**	0.7690	**0.7850**	0.7940	0.7580	0.7416

Table 3: Per-Class results for all TDR Runs. For every temporal class, the highest value is indicated in bold.

doesn't retrieve the most relevant documents for this topic highly enough.

The performance of the diversified results is measured using *D#-nDCG@20* that combines intent recall (*I-rec@20*), and *D-nDCG@20* which is a form of nDCG measure where the gain is replaced with a global gain value that takes into consideration the per intent graded relevance for a document. Our diversification approach performs well across learning to rank runs (table 2), since the intent recall is high across all topics as we consider the top 100 relevant documents returned for each temporal class and since our *nDCG* values are relatively high for the individual temporal classes as well. The earth mover's distance works well in our case since we diversify the candidate documents by selecting those documents that are highly ranked in the individual lists and those that are most diverse in the temporal distribution of the time references. Our method could be improved, if instead of a random selection for selecting the first document, we use a method to choose the most confident document from across the temporal classes. Also, considering only the top 20 relevant documents from each intent could improve the *D#-nDCG@20* measure.

4. CONCLUSIONS

In this paper we discussed our approaches for solving the TID and TDR subtask part of the Temporalia-2 task. For the TID subtask, we used a rule-based voting method that is comprised of intent specific rules which use query-specific features such as verb tense of the query, temporal expression identified from the query and the multinomial distribution of n-grams in the query. The evaluation results show that incorporating the linguistic and temporal features helps improve the estimation of the temporal intent of a query. Adding a rule, based on dictionary words related to future could help improve the estimation of queries that have a higher probability for future intent. For atemporal queries that have candidate years returned from the GTE service, rules which determine the spread of years across time is needed for disambiguation.

In the TDR subtask, we jointly classified all the subtopics together to get unique subtopic-intent pairs. We then built separate learning to rank models for each temporal intent using features extracted from the document. The temporal relevance score helped improve the performance measures for future and past intents. The overall performance could be improved by mining other indicative search questions for an intent, as well as using the named entity tags to improve content relevance and to determine important time expressions.

The temporal diversification of results using earth mover's distance was an effective approach exploiting the temporal distribution of time references in the documents. However, this could be improved by experimenting with smaller candidate sets of relevant documents from each individual intent.

5. REFERENCES

[1] O. Alonso, J. Strötgen, R. Baeza-Yates, and M. Gertz. Temporal information retrieval: challenges and opportunities. In *Proceedings of TWAW'11 associated with WWW'11*, 2011.

[2] K. Berberich, S. Bedathur, O. Alonso, and G. Weikum. A Language Modeling Approach for Temporal Information Needs. In *Proceedings of ECIR*, 2010.

[3] R. Campos, G. Dias, A. Jorge, and A. Jatowt. Survey of Temporal Information Retrieval and Related Applications. *ACM Computing Surveys (CSUR)*, 47(2):15, 2014.

[4] R. Campos, G. Dias, A. Jorge, and C. Nunes. Gte: A distributional second-order co-occurrence approach to improve the identification of top relevant dates in web snippets. In *Proceedings of CIKM*, pages 2035–2039, 2012.

[5] A. X. Chang and C. D. Manning. Sutime: A library for recognizing and normalizing time expressions. In *Proceedings of 8th International Conference on Language Resources and Evaluation*, 2012.

[6] Y. Freund, R. Iyer, R. E. Schapire, and Y. Singer. An efficient boosting algorithm for combining preferences. *Journal of Machine Learning Research*, pages 933–969, 2003.

[7] Y. Hou, Q. Chen, J. Xu, Y. Pan, Q. Chen, and X. Wang. HITSZ-ICRC at the NTCIR-11 temporalia task. In *Proceedings of the 11th NTCIR Conference on Evaluation of Information Access Technologies*, 2014.

[8] H. Joho, A. Jatowt, R. Blanco, H. Yu, and S. Yamamoto. Overview of the NTCIR-12 Temporal Information Access (Temporalia-2) Task. In *Proceedings of the 12th NTCIR Conference on Evaluation of Information Access Technologies*, 2016.

[9] N. Kanhabua, R. Blanco, and K. Nørvåg. Temporal information retrieval. *Foundations and Trends in Information Retrieval*, 9(2):91–208, 2015.

[10] N. Kanhabua and K. Nørväg. Determing Time of Queries for Re-Ranking Search Results. In *Proceedings of ECDL*, pages 13–25, 2010.

[11] X. Li and W. Croft. Time-based language models. In *Proceedings of CIKM*, pages 469–475, 2003.

[12] T.-Y. Liu. Learning to Rank for Information Retrieval. In *Springer*, 2011.

[13] D. Metzler, R. Jones, F. Peng, and R. Zhang. Improving Search Relevance for Implicitly Temporal Queries. In *Proceedings of SIGIR*, 2009.

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

2008.

[15] R. Socher, J. Bauer, C. D. Manning, and A. Y. Ng. Parsing with compositional vector grammars. In *Proceedings of ACL conference*, 2013.

[16] K. Toutanova, D. Klein, C. D. Manning, and Y. Singer. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of HLT-NAACL*, pages 252–259, 2003.

[17] Q. Wu, C. Burges, K. Svore, and J. Gao. Adapting boosting for information retrieval measures. *Journal of Information Retrieval*, 2007.

[18] J. Xu and H. Li. AdaRank: a boosting algorithm for information retrieval. In *Proceedings of SIGIR*, 2007.

[19] H.-T. Yu, X. Kang, and F. Ren. TUTA1 at the NTCIR-11 temporalia task. In *Proceedings of the 11th NTCIR Conference on Evaluation of Information Access Technologies*, 2014.

[20] C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to Ad Hoc information retrieval. In *Proceedings of SIGIR*, 2001.