KDETM at NTCIR-12 Temporalia Task: Combining a Rule-based Classifier with Weakly Supervised Learning for Temporal Intent Disambiguation

Abu Nowshed Chy, Md Zia Ullah, Md Shajalal, and Masaki Aono
Department of Computer Science & Engineering
Toyohashi University of Technology
Toyohashi, Aichi, Japan
[nowshed, arif, shajalal]@kde.cs.tut.ac.jp and aono@tut.jp

ABSTRACT

Web is gigantic and being constantly update. Everyday lots of users turn into websites for their information needs. As search queries are dynamic in nature, recent research shows that considering temporal aspects underlying a query can improve the retrieval performance significantly. In this paper, we present our approach to address the Temporal Intent Disambiguation (TID) subtask of the Temporalia track at NTCIR-12. Given a query, the task is to estimate the distribution of four temporal intent classes including Past, Recency, Future, and Atemporal based on its contents. In our approach, we combine a rule-based classifier with weakly supervised classifier. We define a set of rules for the rule-based classifier based on the temporal distance, temporal reference, and POS-tag detection, whereas a small set of query with their temporal polarity knowledge are applied to train the weakly supervised classifier. For weakly supervised classifier, we use the bag-of-words feature and TF-IDF score as a feature weight. Experimental results show that our system reaches the competitive performance among the participants in Temporalia task.

Team Name

KDETM

Subtasks

Temporal Intent Disambiguation (TID) Subtask.

Keywords

Temporal Information Access, Temporal Information Retrieval, Temporal Intent, Weakly-supervised Learning.

1. INTRODUCTION

The Web is rapidly moving towards a platform for mass collaboration in content production and consumption, and the increasing number of people are turning to online source for daily information needs. Web search queries are dynamic in nature and temporally sensitive. Temporally sensitive means that the intent of user queries changes over time, with some queries occasionally spiking in popularity (e.g., earthquake) and others remaining relatively constant (e.g., youtube) [8]. Many queries may only be answered accurately if their underlying temporal orientations are correctly identified. That is why; recognizing the temporal intent behind users' queries must account for in order to improve the performance of information retrieval (IR) system. For example, we can consider a situation, where a journalist is going to prepare an investigation report about "Paul Walker's accident". To find out details about the accident, he or she turns to search news from several different aspects including the accident's major facts, reactions from eye-witness, commentary from road and transportation authority, etc. Here, the intent of the journalist is to trace the most recent and relevant news about the accident; but any search that contains the paul walker's name will bring up the news from different points in time. We can think of another situation, where the journalist is going to write a report about the impact of social media after the general election in Japan. In that case, he or she also searches news in a time frame. For exploring such kinds of search behaviors and boosting the retrieval performance by extracting underlying temporal intents of user queries, NTCIR-11 was first introduced the Temporal Information Access (Temporalia) task in 2014 [5]. Based on the achievements at NTCIR-11, this year NTCIR-12 set new technical challenges of Temporalia-2 [6] that involves two subtasks to address temporal information access technologies including, Temporal Intent Disambiguation (TID) Subtask and Temporally Diversified Retrieval (TDR) Subtask.

In this paper, we present our participation to the Temporal Intent Disambiguation (TID) subtask. In this subtask, given a query, systems are required to estimate the distribution of four temporal intent classes including Past, Recency, Future, and Atemporal. Search queries are atemporal when they do not have a temporal intent. Therefore the corresponding search results are in principle not expected to change due to the passing of time. On the other hand, search results for past, recency, and future queries are related to time. Recency queries refer to present events, future queries refer to predictions or scheduled events, and past queries are related to events already happened [2]. The TID task is challenging mainly because there is little input from users. In our approach, we combine a rule-based classifier with weakly supervised classifier. We define a set of rules for the rule-based classifier based on the temporal distance, temporal reference, and POS-tag detection, whereas a small set of query with their temporal polarity knowledge are applied to train the weakly supervised classifier. For training the weakly supervised classifier, we use the bag-of-words feature and TF-IDF score as a feature weight.

The rest of the paper is structured as follows: **Section 2** describes the state-of-the-art of temporal query intent retrieval. Next, we will introduce our proposed framework in **Section 3**. **Section 4** includes experiments and evaluation to show the performance of our proposed method. Some concluded remarks and future directions of our work described in **Section 5**.

2. RELATED WORK

Temporal intent extraction plays a crucial role in the field of Information Retrieval (IR). That is why; temporal intent extraction has become a research focus in recent years. At NTCIR-11, several researchers tried to address such kind of problem by extracting various linguistic level features, bag-of-words features, difference of query issuing time feature, verb features, etc. along with the ensemble of several machine learning algorithm as well as combined with the rule-based classifier [4] [11] [2] [3].

3. OUR APPROACH

In this section, we describe the details of our proposed framework. Given a query, the goal of our proposed temporal intent disambiguation subtask is to estimate the distribution of four temporal intent classes including Past, Recency, Future, and Atemporal based on its contents. The overview of our proposed framework depicted in Fig. 1.

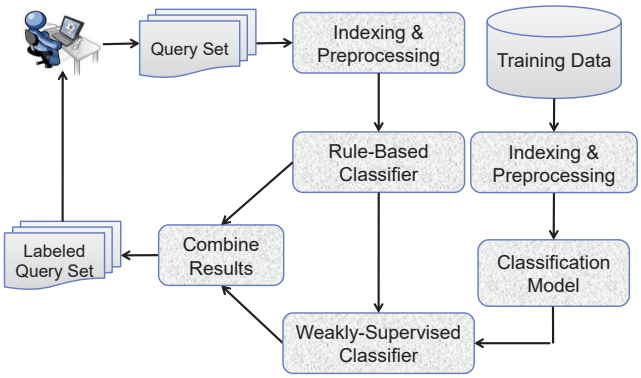

Figure 1: Temporal Intent Disambiguation (TID) System

At first, our system fetches the set of queries and indexed them for further processing. In the preprocessing stage, we perform the tokenization, stop-word removal, single-letter word removal, and special character removal. Next, our proposed rule-based classifier is applied to classify the queries temporal intent as past, future or unknown. After that, each query is considered as a bag-of-words and classified by using weakly supervised Naive-Bayes (NB) classifier and Support Vector Machine (SVM) classifier. Results of both rule-based classifier and weakly supervised classifiers are then combined and set of labeled queries are returned.

3.1 Rule-Based Classifier

In rule-based classifiers, we usually construct a set of rules that determine a certain combination of patterns, which are most likely to be related to the different classes. Each rule consists of an antecedent part and a consequent part. The antecedent part corresponds to a word patterns and the consequent part corresponds to a class label. We can define a rule as follows:

$$R_j : if\ x_1\ is\ A_{j1}\ and\ .....\ x_n\ is\ A_{jn}$$
$$then\ Class = C_j, \quad j = 1, ..., N$$

where R_j is a rule label, j is a rule index, A_{j1} is an antecedent set, C_j is a consequent class, and N is the total number of rules. Our unsupervised rule-based classifier casts the Temporal Intent Disambiguation (TID) Subtask problem as a multi-class classification problem and labeled each query as past, future or unknown. To achieve this, we defined the following set of rules based on the temporal distance, temporal reference, and part-of-speech (POS) tag:

3.1.1 Temporal Distance

There are some queries such as "Madden 2014 Release Date", "Game of Thrones Movie 2012" etc., which contain a digital year strings. That is why; its temporal intent is easy to judge based on the extracted time from query and query issuing time. For this, we make use of SUTime component [1] of Stanford CoreNLP to extract the temporal expressions from the query. Based on the extracted time from query and query issuing time, we define two rules. The first one is, if the extracted time from query earlier than query issuing time, we assign the highest probability value to the past class for that query. The second one is, if the extracted time from query later than query issuing time, we assign the highest probability value to the future class for that query.

3.1.2 Temporal Reference

Sometimes queries may contain some common holiday names such as "Fathers Day", "Mothers Day", "Thanksgiving" etc. For this we make use of default holiday list of SUTime to extract the respective temporal expression. Based on the extracted time of the query we use the two rules described in 3.1.1 to assign the probability value of respective class.

3.1.3 Part-of-Speech (POS) Tag Based Rule

Every query is run through Stanford CoreNLP [9] to perform part-of-speech (POS) tagging. From the resulting annotations, we define a rules. If a query start with s wh-pronoun such as WR, WP, etc. following an inflected verb such as VBD, then we assign the highest probability value to the past class for that query.

3.2 Weakly Supervised Learning Approach

Weakly supervised classifier uses the prior documents with their temporal polarity knowledge, where a small number of seed documents with known polarity are used to infer the polarity of a target document. The weakly supervised learning process described in detail as follows:

3.2.1 Data Preprocessing

The data preprocessing step is initiated with tokenization, which is the process of forming tokens from an input stream of characters. Sometimes query may contain emoticons and other special characters. But meaningful English words do not contain these characters. So, we remove these from queries as well as removing the single-letter word. Moreover, stop-word removal also performed in this stage. For stop-word removal, we applied the refined form of Indri's standard stop-list[1].

[1]http://www.lemurproject.org/stopwords/stoplist.dft

3.2.2 Naive-Bayes (NB) Classifier

The basic idea of Naive-Bayes (NB) classifier is to use the joint probabilities of words and categories to estimate the probabilities of categories for a given query. The naive part of such a model is the assumption of word independence, which makes the computation of this classifier far more efficient than the exponential complexity of non-Naive-Bayesian approaches. By using Naive-Bayes classifier, given a query (Q), the temporal probability distribution of each temporal class is estimated as follows:

$$P(C_k|Q) = P(C_k) \cdot P(Q|C_k)$$

where $P(C_k|Q)$ is the probability that a given query, Q belongs to a class, C_k. Assuming uniform priors over query documents and term independence:

$$P(Q|C_k) \ = \ \prod_{i=1}^{|Q|} P(w_i|C_k)$$

where $|Q|$ is the number of words in the query documents and $P(w_i|C_k)$ is the probability that the i-th word of a given document occurs in a document from category, C_k. When the size of the training set is small, the relative frequency estimates of probabilities, $P(w_i|C_k)$, will not be reasonable. If a word never appears in the given training set, its relative frequency estimate will be zero. To overcome this limitation, the Laplace law of succession is applied to estimate $P(w_i|C_k)$ as follows:

$$P(w_i|C_k) = \frac{N_{ct} + \lambda}{N_c + \lambda V}$$

where N_{ct} is the number of times the word occurs in that category, C_k. N_c is the number of words in category, C_k. V is the vocabulary size. λ is the positive constant and we set it 0.5 to avoid zero probability.

3.2.3 Support Vector Machine (SVM) Classifier

We use the multiclass SVM implementation from [10]. It uses the multi-class formulation described in [7]. For a training set $(x_1, y_1)...(x_n, y_n)$ with labels y_i in $[1..k]$, it finds the solution of the following optimization problem during training:

$$min 1/2 \sum_{i=1..k} w_i * w_i + C/n \sum_{i=1..n} \xi_i$$
s.t. for all y in $[1..k]$:
$[x_1 * w_{yi}] >= [x_1 * w_y] + 100 * \triangle(y_i, y) - \xi_1$
. . . . . .
s.t. for all y in $[1..k]$:
$[x_n * w_{yn}] >= [x_n * w_n] + 100 * \triangle(y_n, y) - \xi_n$

where C is the usual regularization parameter that trades off margin size and training error. We estimate the optimal value of C using cross-validation. $\triangle(y_n, y)$ is the loss function that returns 0 if y_n equals y, and 1 otherwise. To solve this optimization problem, $SVM^{multiclass}$ uses an algorithm based on Structural SVMs. For training SVM. we use the bag-of-words feature and TF.IDF score as a feature weight.

Term Frequency (TF):
Term frequency (TF) means frequency of a term/keyword in a document. The higher the TF, the higher the importance(weight) for the document.

Let a document, d_1 and $d_1 = w_1, w_2, ..., w_k$ words with frequency $f_1, f_2, ..., f_k$ respectively. Then,

$$\text{Term Frequency } TF_i = \frac{f_i}{\sum_k f_k}$$

Inverse Document Frequency (IDF):
The Inverse Document Frequency is a measure of the general importance of the term (obtained by dividing the total number of documents by the number of documents containing the term, and then taking the logarithm of that quotient).

$$\text{Inverse Document Frequency } IDF_i = log \frac{|D|}{|d : t_i \ \in \ d|}$$

where, $|D|$: total number of documents in the corpus. $|d : t_i \ \in \ d|$: number of documents where the term t_i appears. If the term is not in the corpus, this will lead to a division-by-zero. It is therefore common to use $1 + |d : t_i \ \in \ d|$.

Finally, $TF.IDF$ is used to produce a composite weight for each term in each document.

$$TF.IDF = TF_i * IDF_i$$

3.3 Combining the Classifiers

After developing our proposed rule-based classifier and training the supervised classifiers including Naive-Bayes and SVM, we combine them to estimate the temporal probability of each class for a given query. At first, our rule-based classifier is applied to classify the query as past, future or unknown. Query that are classified as the past or future, we assign the highest probability value to that class and the probability of the remaining class assigned the zero probability. Next, for the query that are classified as unknown by the rule-based classifier, we consider the predictions of supervised classifier as the final temporal probability of each class.

4. EXPERIMENTS AND EVALUATION

4.1 Dataset Collection

The Temporal Intent Disambiguation (TID) subtask at NTCIR-12 provides a dry run dataset of ninety three (93) queries along with their respective temporal distribution of classes and issuing time and three hundred (300) queries along with their issuing time were released as formal run data. To train our weakly supervised classifier, we use the dry run and formal run query set of Temporal Query Intent Classification (TQIC) subtask at NTCIR-11, and dry run dataset of TID subatask at NTCIR-12

4.2 Evaluation Measure

For a specific query q, let $P = p_1, p_2, p_3, p_4$ denote its standard temporal class distribution, and $W = w_1, w_2, w_3, w_4$ denote the temporal class distribution from a system. The classification loss for a single query will be measured using the following two ways.

Metric-1: Averaged per-class absolute loss, i.e.,

$$\frac{1}{4} \sum_{i=1}^{4} |w_i - p_i|$$

Metric-2: Cosine similarity between the two probability vectors P and W, i.e.,

$$cos\theta(P,W) = \frac{P \cdot W}{|P||W|} = \frac{\sum_{i=1}^{4} |p_i * w_i|}{\sqrt{\sum_{i=1}^{4} p_i^2}\sqrt{\sum_{i=1}^{4} w_i^2}}$$

The final performance is the averaged value across all test queries.

4.3 Submitted Runs and Results

In this section, we describe the run configuration of our submitted runs and the evaluation results of formal runs for our submitted runs as shown in table 1.

KDE_Run1: At first, rule based classifier is applied to estimate the temporal probability distribution of each query as described in section 3.3. Next, for the query that are classified as unknown by the rule-based classifier we estimate the probability score of each temporal class, P(C) as follows:

$$P(C) = P(NB) + P(SVM) - P(NB) \cdot P(SVM)$$

where $P(NB)$ is the probability score from Naive-Bayes classification model and $P(SVM)$ is the probability score from SVM classification model. After estimating the score for each temporal class, rounded percentage value of each score is assigned as a final probability score.

KDE_Run2: At first, rule based classifier is applied to estimate the temporal probability distribution of each query as described in section 3.3. Next, for the query that are classified as unknown by the rule-based classifier, we applied the Naive-Bayes classification model to estimate the temporal probability distribution of each class.

KDE_Run3: At first, rule based classifier is applied to estimate the temporal probability distribution of each query as described in section 3.3. Next, for the query that are classified as unknown by the rule-based classifier, we applied the SVM classification model to estimate the temporal probability distribution of each class.

Table 1: Performance of Our Submitted Runs

Method	Avg. Cosine	Avg. Loss
KDE_Run1	0.6578	0.2342
KDE_Run2	0.6972	0.2173
KDE_Run3	0.4454	0.2706

Our rule-based classifier works fine to estimate the temporal probability distribution of each class. However vocabulary mismatch problem affects the KDE_Run3 severely due to small training set and lacks of technique to handle the zero probability score for a feature never seen in our training data.

5. CONCLUSIONS

In this paper we presented our approach to the Temporal Intent Disambiguation (TID) Subtask of Temporal Information Access (Temporalia-2) task in the NTCIR-12 challenge. We tackled the problem by combining a rule-based classifier with weakly supervised machine learning approach. We submitted three runs based on a rule-based classifier and two different machine learning classifiers (Naive-Bayes and SVM). Among our submitted runs KDE_Run2 achieved the best performance(AvgCosin = 0.6972 and AvgLoss = 0.2173). There is much room left to further improve our methods in TID subtask. Shortage of training dataset for our machine learning approach is the main problem. In future, we have a plan to overcome this limitation by incorporating more training samples for training i.e.; more annotated user queries extract from search engine query log. We also have a plan to incorporate more rules with complex semantics and explore more temporal features.

6. REFERENCES

[1] A. X. Chang and C. D. Manning. Sutime: A library for recognizing and normalizing time expressions. In *LREC*, pages 3735–3740, 2012.

[2] M. Filannino and G. Nenadic. Using machine learning to predict temporal orientation of search engines' queries in the temporalia challenge. In *NTCIR*, pages pages 438–442, 2014.

[3] M. Hasanuzzaman, G. Dias, and S. Ferrari. Hultech at the ntcir-11 temporalia task: Ensemble learning for temporal query intent classification. In *The 11th NTCIR Conference on Evaluation of Information Access Technologies*, pages p–478, 2014.

[4] Y. Hou, C. Tan, J. Xu, Y. Pan, Q. Chen, and X. Wang. Hitsz-icrc at ntcir-11 temporalia task. In *NTCIR*, pages pages 468–473, 2014.

[5] H. Joho, A. Jatowt, R. Blanco, H. Naka, and S. Yamamoto. Overview of NTCIR-11 temporal information access (temporalia) task. In *Proceedings of the NTCIR-11 Conference on Evaluation of Information Access Technologies, Tokyo, Japan, December 9-12, 2014, pp. 429-437,*. Citeseer, 2014.

[6] 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, June 7-10, 2016, Tokyo, Japan*, 2016.

[7] K. Krammer and Y. Singer. On the algorithmic implementation of multi-class svms. *Proc. of JMLR*, pages pages 265–292, 2001.

[8] A. Kulkarni, J. Teevan, K. M. Svore, and S. T. Dumais. Understanding temporal query dynamics. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 167–176. ACM, 2011.

[9] C. D. Manning, M. Surdeanu, J. Bauer, J. R. Finkel, S. Bethard, and D. McClosky. The stanford corenlp natural language processing toolkit. In *ACL (System Demonstrations)*, pages 55–60, 2014.

[10] I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun. Support vector machine learning for interdependent and structured output spaces. In *Proceedings of the twenty-first international conference on Machine learning*, page 104. ACM, 2004.

[11] H. Yu, X. Kang, and F. Ren. Tuta1 at the ntcir-11 temporalia task. pages pages 461–467, 2014.