DUT-NLP-CH @ NTCIR-12 Temporalia Temporal Intent Disambiguation Subtask

Jiahuan Pei[1], Degen Huang[2], Jianjun Ma[3],
Dingxin Song, Leyuan Sang
Department of Computer Science and Technology
Dalian University of Technology
Dalian 116023, Liaoning, P.R. China
p_sunrise@mail.dlut.edu.cn[1], huangdg@dlut.edu.cn[2], majian@dlut.edu.cn[3]

ABSTRACT

This paper details our participation in Temporal Intent Disambiguation Subtask of NTCIR-12. In this paper, we take the subtask as a problem of classification and our major job is to find some distinguishable features to feed machine learning classifiers to estimate a distribution of four temporal intent classes for a given query. Considering the lack of the temporal information in queries, we expand it by two types of features: explicit features extracted from the context of queries, and implicit time gap features from time series analysis using Google Trends. In addition, some textual features, such as word-based probability distribution, temporal trigger words and center word of query, have been added to classifiers with probability estimation. Finally, we select the best three classifiers to get the submitted runs, where our best result is 0.8135 of AvgCosin measure and 0.1710 of AvgAbsLoss measure. After the submission of the formal run, further research has been done with a better result as 0.8886 of AvgCosin measure and 0.1286 of AvgAbsLoss measure.

Team Name

DUT-NLP-CH

Subtask

Temporal Intent Disambiguation (Chinese)

Keywords

Temporal Intent Disambiguation, Temporal Feature Extraction, Temporal Query Classification

1. INTRODUCTION

With the development of time-aware search engines in recent years, the increasing interest in Temporal Information Retrieval has heightened the need for user's temporal intent behind queries [1, 2] .

Temporal query is a query that contains a search user's time-related information as is reported by Joho et al [3]. Temporal query intent classification (TQIC) task was first proposed by Jones and Diaz [4] to categorize queries into temporally unambiguous, temporally ambiguous and atemporal queries and NTCIR-11 expended this idea to get more detailed categories as past, recency, future and atemporal. Vlachos et al [5] divided temporal queries into three patterns, that is, periodic, seasonal and large peak queries. Kulkarni et al. [6] categorized temporal queries by features extracted from query population and its changes. By analyzing queries' temporal intent distributions and searching frequency over time record in web query log, Ren et al [7] investigated the automatic detection of web queries and categorized users' temporal intents into hierarchical temporal classes. Parikh and Sundaresan [8] detected the shape and interval of the bursts to classify time-related queries. Kanhabua and Nørvåg [9] proposed a new query modeling method based on analysis of top-k retrieved documents. Overall, extensive research has been done on temporal queries classification.

The DUT-NLP-CH team participated in Chinese Temporal Intent Disambiguation (TID) Subtask [10]. Comparing to TQIC subtask at NTCIR-11, queries in this subtask are more likely to be ambiguous with few or even without explicit temporal expressions. Moreover, a query refers to each temporal intent with a certain probability.

In this task, there are mainly four challenges: (1) *Lack of explicit temporal information.* According to the analysis of the log of Excite Search Engine, a search query only contains an average of 2.4 words [11], let alone explicit temporal information as time expression or temporal trigger words. (2) *No query log available.* Many other query intent analyzing tasks usually mine essential features from log files, but in this task, only query string and its submission time are provided. (3) *Temporal intent may change over time.* Intent of some seasonal queries, such as "法国网球公开赛", may differ in terms of different submission time. Therefore, besides static features extracted from queries themselves, some dynamic features should also be taken into consideration. (4) *Temporal intent ambiguities.* As is known, query strings are presented as a sequence of keywords, which are generally incomplete in grammar, so one search query may contain more than one meaning. For instance, the intent of the query "廉价夏令营" is most likely to refer to Future, because people usually make preparation before an activity. Therefore, this paper will focus on the challenges mentioned above and propose a novel approach to exploit users' temporal intent behind queries with only submission time and query strings by estimating the distribution of four temporal classes, that is, **Past**, **Recency**, **Future** and **Atemporal**.

2. DATASETS

For this task, we achieved data from different resources and processed them as follows:

DATA1: 52 dry run quires released by NTCIR-12 organizer with textual representation of queries, their submission date and distribution of four temporal intent classes.

DATA2: 300 formal run data from NTCIR-11 TQIC subtask [12]. Given that the queries of TQIC task are presented in English and their tagging result is one of temporal classes, we take some measures to make them available for our task. First, we used Google Translation API to translate English query strings into Chinese and then an artificial correction was used to make the translation clear and fluent. Next, some ambiguous queries like "10 year old found alive" cannot be mapped exactly into Chinese, so we kept them as foreign strings in Chinese search engines. Finally, we converted one class tagging into distributional tagging as one of {1, 0, 0, 0}, {0, 1, 0, 0}, {0, 0, 1, 0} and {0, 0, 0, 1}.

DATA3: 503 time-sensitive search data extracted from SogouQ log data. We mapped *time of the click event* and *user query* into *query_string* and *query_issue_time* for this task respectively. Then the data were annotated independently by three members based on their temporal intent and kappa value was used to make a consistency test.

TESTDATA: We accessed 300 testing data from NTCIR-12 TID subtask. We regarded the distribution of four temporal intent classes for a query as a probability vector and analyzed the dataset by getting the weighted average of the vectors. Table 1 shows the result.

Table 1. Average distribution of four temporal intent classes

Past	Recency	Future	Atemporal	Total
0.13	0.16	0.07	0.64	1

3. APPROACH

We regard this task as a classification problem with probability output since each query has a distributional tagging vector of four temporal classes. Our overall method relies on well-designed features and well-established classifiers.

First, we employed Chinese Segmentation, POS tagging, Name Entity Recognizer and Parser in Stanford Corenlp tookit [13] to preprocess the search queries, and recognized Temporal Expression in query texts by a Chinese temporal tagging tool named HeidelTime [14]. Then, we extracted designed features from preprocessed results and sent them to ten classifiers provided by scikit-learn, a machine learning module in Python. Finally, according to two TID evaluation indexes, we selected top-3 models for testing submission runs. In the development stage, we combined **DATA2** with **DATA3** as train set and chose **DATA1** as test set. In the formal run testing process, we used features, classifiers and parameters designed in the development stage.

4. FEATURE DESIGN

4.1 Time Gap Features

Temporal expressions in queries indicate the user's temporal information directly. As Yu et al. [15] stated explicit time gap features are ideal features for temporal intent classification task, defined as the gap between the user's search intent time point and the submission time.

We used HeidelTime to recognize the temporal expression and then made some rules to map their value to time gap features. The time gap features in this paper refers to "PAST_REF", "RECENCY_REF", "FUTURE_REF" and "IMPLICIT_REF". Value of some temporal expressions can be directly mapped into time gap feature. For example, "近期" in the query "近期 油价上涨" was tagged with value "FUTURE_REF". However, in more general cases, we should manage to evaluate the time point which query intent refers to and then calculate the gap between two time points. For example, in the query "4 月 工作汇报","4 月" was identified as a temporal expression with value of "2012-04" by setting the query submission time "2012-07-31" as reference time； therefore， it refers to "PAST_REF" with past temporal intent. And also in some queries like "2013 年 父亲节", we can get the temporal expression but hardly estimate the time gap without specific time point. We represented these features as "IMPLICIT_REF" to indicate that the query is time-sensitive but its intent needs further confirmation.

4.2 Word-based Probability Distribution Features

In the previous work, some salient words were explored to indicate the intent of query. For instance, the word "历史" in "排球 的 历史" emphasizes "past" events about volleyball with the long passage of time referring to ***Past*** intent, and the word "股票" in "fb 股票价格" is expected to be timely and up to date thus containing more intent about ***Recency***. However, it's a difficult process to select these trigger words manually and a word may refer to different temporal intents with a probability rather than one certain temporal intent. Therefore, we introduced word-based probability distribution vector as:

$$\vec{v} = \left(P_{past}, P_{recency}, P_{future}, P_{atemporal}\right)$$

which was automatically constructed by accumulating the "contribution" of all the words in a query. First, we took a query as unigram model and presented a query string as bag-of-words and used Naive Bayes define p_i as:

$$P(C_i|\text{Query}) = \frac{P(C_i)\prod_{w_j \in dict} P(w_j \mid C_i)^{TF(w_j, Query)}}{\sum_{k=1}^{N} P(C_k)\prod_{w_j \in dict} P(w_j \mid C_k)^{TF(w_j, Query)}}$$

where N is the number of temporal intent categories, which is set to 4 in this paper, $P(C_i)$ is the probability of the queries categorized into class C_i, $TF(w_j, Query)$ is the frequency of w_j occurring in this query and we use Laplace distribution define $P(w_j|C_i)$ as:

$$P\left(w_j|C_i\right) = \frac{1 + TF\left(w_j, C_i\right)}{|dict| + \sum_{t \in Query} TF\left(w_t, C_i\right)}$$

where $TF(w_j, C_i)$ is the frequency of w_j occurring in the queries of class C_i and $|dict|$ is the size of the dictionary which constructed from training data. In order to get a reasonable dictionary, we processed words in corpora by filtering stop words and less distinctive words if they satisfy the formula:

$$\sum_{i=1}^{N} (p_i - P(C_i))^2 \le \varepsilon$$

In our experiment, we designed $\varepsilon = 0.01$ to remove the words without obvious distinction among the four types of temporal intent.

4.3 Temporal Trigger Word Features

Tense of verbs has a natural advantage over temporal intent representation in English queries, but for Chinese, there are typical temporal trigger words rather than conjugations of verbs that refer to temporal information. We employed a simple method to exploit trigger words. First, we extracted the word from vocabulary of training set as a trigger candidate if $P(w_j|C_i) \geq 0.7$. Then, we manually selected the words that strongly indicate the temporal categories from candidates and got the four trigger word sets. And then we expanded word sets by word clustering using syntactic word vectors, which we trained from 1.2GB Wikipedia dump file[1] by Word2Vec toolkit [16] and manually filtered the words again. Table 2 shows some examples of temporal trigger words. We finally designed P_TIGGER, R_TIGGER, F_TIGGER and A_TIGGER features to represent the number of the four types of temporal trigger words.

Table 2. Temporal trigger words

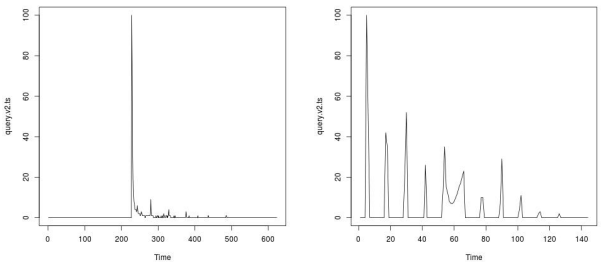
Past	Recency	Future	Atemporal
以往	近况	未来	博客
往届	金价	预测	歌词
回顾	招聘	即将	计算器

4.4 Other Explicit Textual Features

In addition to the three major time features, there are some other textual features that can be extracted directly from the preprocessed results. We defined them as ***CenterWord***, ***posOfCenterWord***, ***validQueryLength***, ***numOfNER***, ***numOfNotChWords*** and ***isNounFreg***. The feature ***CenterWord*** is the word which is parsed as a root in the dependency tree by Stanford Parser and the feature ***posOfCenterWord*** indicates the Part-of-speech of the center word. The feature ***validQueryLength*** refers to the number of words without *stopwords* and ***numOfNER*** represents the number of Name Entities in a query filtered stopwords. The ***numOfNotChWords*** means the number of words which do not belong to Chinese, such as "fb" in the query "fb 股价". The binary feature ***isNounFrag*** means whether the query has a Noun Phrase or Noun Fragment structure in the parsing results.

4.5 Time Gap Features from Google Trends

As we have mentioned above, explicit temporal information is absolutely rare in user's search strings, so in this paper we focus on mining potential temporal information beyond the query strings by analyzing data from Google Trends, a public website based on Google Search providing users' search-volume weekly or monthly. We used the data to estimate users' temporal intent mainly based on the hypothesis that query volume can reflect users' interest in a query. For example, search volume of the query "汶川地震" bursts at the 228th points and its corresponding time intervals is "2008-05-11 - 2008-5-17" which indicates the event time. Another typical example of this is that , volume of the query "法国网球公开赛" bursts periodically because the French Open, a major tennis tournament, is held annually between late May and early June. Figure 1 shows the search volume of the two examples.



(a) 汶川地震 (b)法国网球公开赛

Figure 1. Search Volume Accessed from Google Trends

We adopted four time-related features named GT_{past}, $GT_{recency}$, GT_{future} and $GT_{atemporal}$ to indicate the temporal intent via information of search volume. And we extracted them by the following processes,

(1) ***Preprocessing*** First, we filtered the temporal expressions in the original query strings and sent the remaining parts to Google Trends and downloaded their csv files. We removed temporal expressions mainly because Google Trends calculated search volume of the whole query. Second, we extracted only sampling time points in the format of "YYYY-MM-DD - YYYY-MM-DD" or "YYYY-MM" and their corresponding search-volume values. Then, we resampled the data by month and formatted the time as "YYYY-MM".

(2) ***Extraction*** Inspired by the idea from Ren et al. [8], we tackled these features extraction task as a time series classification problem. We extracted the 11 features mentioned in Ren's paper, and used the SVC model with Gaussian kernel they provided to predict the categories. Then, for each query we got the probability output of four classes as P_{QoT}, P_{OQ}, P_{AMQ} and P_{PMQ}. The probability of the *QoT* can be directly mapped into $GT_{atemporal}$ value, because *QoT* equals to *Atemporal*. The remaining problem is to propose an approach to divide the 1-P_{QoT} into three parts for feature GT_{past}, $GT_{recency}$ and GT_{future}.

In this task, the data before query submission time is reliable but volume after that is actually invisible. Therefore, we use Auto-Regressive and Moving Average (ARMA) [17] model to predict the "future" volume after submission time. And then we adopted the average absolute length of interest points in area A, B, C to linear SVC classifier to learn the feature GT_{past}, $GT_{recency}$ and GT_{future}. With reference to Figure 2 we can see that all the interest points are located in the left time interval and the feature GT_{past} get all the remaining probability, that is $GT_{past} = 1 - P_{QoT}$.

[1] http://dumps.wikimedia.org/zhwiki/20151123/zhwiki-20151123-pages-articles-multistream.xml.bz2

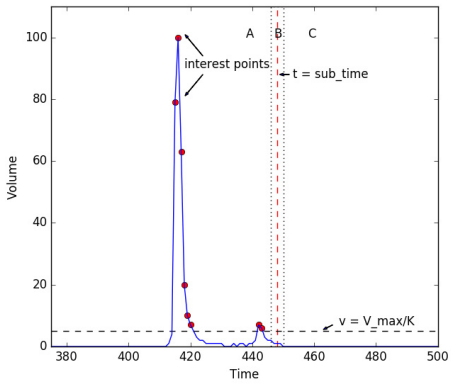


Figure 2. An example the query "新亮剑"

We listed all the features that we mentioned in Section 4 in Table 3.

Table 3. Feature list

Group name	Feature No.	Feature name	meaning or value
Time Gap	f1-f4	PAST_REF/RECENCY_REF/ FUTURE_REF/IMPLICIT_REF	0 or 1
Word-based Probability Distribution	f5-f8	$P_{past}/P_{recency}/P_{future}/P_{atemporal}$	$0 \le P \le 1$ sum(P)=1
Temporal Trigger Word	f9-f12	P_TIGGER/R_TIGGER/ F_TIGGER/A_TIGGER	1,2,...,N
Other textual Features	f13	CenterWord	1,2,...,N
	f14	posOfCenterWord	1,2,…,33
	f15	validQueryLength	1,2,…,N
	f16	numOfNER	1,2,…,N
	f17	numOfNotChWords	1,2,…,N
	f18	isNo,unFrag	0 or 1
Google Trends' Time Gap	f19-f22	$GT_{past}/GT_{recency}/GT_{future}/GT_{atemporal}$	$0 \le GT \le 1$ sum(GT)=1

5. RESULTS

5.1 Formal Run Results

We now describe and analyze our experimental results for formal runs and after their submission. Table 4 shows the results we submitted as formal runs.

Table 4. Results of formal runs

Run	AvgCosin	AvgAbsLoss
1	**0.8135**	0.1728
2	0.8066	0.1854
3	0.8116	**0.1710**

Run1 is the best result of our formal runs in terms of the average value of the cosine similarity (**AvgCosin)** by C-Support Vector Classification (SVC) with a linear kernel function and a default C-value, which is based on libsvm. Run3 is the result of Logistic Regression (LR) model with *l*1 penalty, which represents lower value of averaged per-class absolute loss (**AvgAbsLoss**) than Run1. Run2 is the results of Random Forest (RF) model with balanced class weights. Detailed information about features that we used in the experiment is stated in Section 5. We combined **DATA2** and **DATA3** as training set, and **DATA1** is the developing set.

5.2 A posteriori improvement

For further research after our submission, we made some comparative experiments on different models as follows: C-Support Vector Classification (SVC), Nu-Support Vector Classification (NuSVC) Random Forest (RF), Logistic Regression (LR), Gaussian Naive Bayes (GNB), Multinomial Naive Bayes (MNB), Linear Discriminant Analysis (LDA), and Decision Tree (DT). Some models are stable because the distribution of four temporal intent classes will not change over times run while others change slightly. Table 5 shows the results of different models with default parameter settings based on two basic features: time gap features and temporal trigger word features. **DATA1** and **DATA2** were combined as training data, and 300 formal run data treated as testing data.

Table 5. Comparison among different models

Model		AvgCosin	AvgAbsLoss
SVC	linear	0.8639±0.0011	0.1640±0.0003
	rbf	0.8658±0.0016	0.1637±0.0005
	poly	0.6657±0.0043	0.2543±0.0013
NuSVC	linear	0.8692±0.0008	0.1635±0.0005
	rbf	**0.8724**±0.0020	**0.1611**±0.0006
	poly	0.8280±0.0056	0.1875±0.0028
RF	balanced	0.8544±0.0062	0.1700±0.0038
	unbalanced	**0.8862**±0.0019	**0.1262**±0.0014
LR	*l*1	0.8623	0.1674
	*l*2	0.8453	0.1782
GNB		0.8433	0.1606
MNB		0.7416	0.2145
LDA		**0.8812**	**0.1380**
DT		0.8517	0.1702

From Table 5, some findings we have gotten are as follows:

(1) The result of linear SVC model, LR model with penaly *l*1, RF model with balanced class weight outperforms Run1, Run2, Run3 respectively. It is mainly because **DATA1** and **DATA2** are adopted as our training data while formal run training set is composed of **DATA2** and **DATA3**. We selected queries from SogouQ corpora for **DATA3** without considering the balance of each temporal class. And we annotated the queries under an inconsistent temporal intent with the organizers, which is most likely caused by the ambiguity of search queries.

(2) We found that class weight is an important factor for this task. In TQIC task of NTCIR-11, all the classes share the same weight as 0.25, but in this task, each class has different weights. The adjustment of class weight is an essential technique. For example, RF model with class weight of {Past: 0.13, Recency: 0.16, Future: 0.07, Atemporal: 0.64} performs much better than its balanced model, whose class weight is {Past: 0.25, Recency: 0.25, Future: 0.25, Atemporal: 0.25}.

(3) LR model, GNB model, LDA model and DT model are stable, that is, they will not change their distributional probability

of prediction. Based on this phenomenon, we chose LDA model, which is both well-performed and stable, to do feature selection and got the result as shown in Table 6.

Table 6. Feature Selection based on LDA model

Run	Composition	AvgCosin	AvgAbsLoss
4	baseline	0.8812	0.1380
5	baseline+f7	0.8825	0.1343
6	baseline+f7+f20	0.8831	0.1339
7	baseline+f7+f14	0.8841	0.1335
8	baseline+f7+f14+f13	**0.8886**	**0.1286**

6. CONCLUSIONS

This paper details the approach DUT-CH group addressed for Temporalia task at the NTCIR-12. We participate in TID subtask (Chinese) aiming at predicting the distribution of four temporal intents. For TID Chinese subtask, the formal run results were produced by adopting all the designed features to linear SVC model, Logistic Regression model with *l*1 penalty and Random Forest model with class weight balanced. After the submission of the formal run, we did further experiments to compare different models and feature composition and finally got a better and more stable result by LDA model and selected time gap, word-based probability distribution vector, temporal trigger word, Google Trends' time gap, center word and its Part-of-speech as good features.

In the future work, we will exploit more implicit features by analyzing time-series data, Google Trends and the documents returned by search engines like Google Search and Baidu Search.

7. ACKNOWLEDGMENTS

This work has been supported by the National Nature Science Foundation of China (No. 61173100 61272375) and National Social Science Foundation of China (No. 15BYY175).

8. REFERENCES

[1] R. Campos, G. Dias, AM. Jorge and A. Jatowt. Survey of temporal information retrieval and related applications. *ACM Computing Surveys (CSUR)*. 47(2):15, 2015.

[2] K. Nattiya, R. Blanco, and K. Nørvåg. Temporal Information Retrieval. *Foundations and Trends in Information Retrieval*, 9(2): 91-208, 2015.

[3] H. Joho, A. Jatowt, and B. Roi. A Survey of Temporal Web Search Experience. In *Proceedings of the 22nd International Conference on World Wide Web (Companion)*. pages 1101–1108, 2013

[4] R. Jones and F. Diaz. Temporal profiles of queries. *ACM Transactions on Information Systems (TOIS)*, 25(3): 14, 2007.

[5] Vlachos, M., Meek, C., Vagena, Z., Gunopulos, D.: Identifying similarities, periodicities and bursts for online search queries. In *Proceedings of the 2004 ACM SIGMOD international conference on Management of data*, pages 131-142, 2004

[6] 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, 2011

[7] P. Ren, Z. Chen, X. Song, B. Li, H. Yang and J. Ma. Understanding temporal intent of user query based on time-based query classification. In: *Natural Language Processing and Chinese Computing*, pages 334-345, 2013

[8] N. Parikh and N. Sundaresan. Scalable and near real-time burst detection from ecommerce queries. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 972-980, 2008

[9] N. Kanhabua and K. Nørvåg. Exploiting time-based synonyms in searching document archives. In *Proceedings of the 10th annual joint conference on Digital libraries*, pages 79-88, 2010

[10] 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.

[11] A. Spink, D. Wolfram, MBJ. Jansen, and T. Saracevic, Searching the web: The public and their queries. *Journal of the American society for information science and technology*, 52(3): 226-234, 2001.

[12] H. Joho, A. Jatowt, R. Blanco, H. Naka and S. Yamamoto. Overview of NTCIR-11 Temporal Information Access (Temporalia) Task. In *Proceedings of the 11th NTCIR conference*, pages 429-437, 2014.

[13] 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.

[14] H. Li, J. Strötgen, J. Zell and M. Gertz. Chinese Temporal Tagging with HeidelTime. In *EACL*, pages 133-137, 2014.

[15] H. Yu, X. Kang, and F. Ren. TUTA1 at the NTCIR-11 Temporalia Task. In *Proceedings of the 11th NTCIR Conference*, pages 461-467, 2014.

[16] T. Mikolov, K. Chen, G. Corrado and J. Dean. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781, 2013.

[17] W. McKinney, J. Perktold and S. Seabold. Time series analysis in Python with statsmodels. *Jarrodmillman. Com*, pages 96-102, 2011.