WHUIR at the NTCIR-12 Temporal Intent Disambiguation Task

Sisi Gui
Wuhan University
Wuhan, Hubei, China
sgui0229@whu.edu.cn

Wei Lu
Wuhan University
Wuhan, Hubei, China
weilu@whu.edu.cn

ABSTRACT
WHUIR participated in the Temporal Intend Disambiguation (TID) Task of the Temporalia track at NTCIR-12. This paper describes our work of this specific subtask. Given a query, the task is to assign the probability value to four temporal classes i.e. Past, Recency, Future or Atemporal. Our overall strategy has been to rely on established off-the-shelf components (e.g., standard classifiers from LIBSVM and natural language processing methods from Stanford CoreNLP) and focus on feature discovering. We considered nineteen features in total from query itself. We used all the features for SVR in different parameter sets and chose the best three sets on the dry run data for the formal run. Results are presented and discussed in this paper.

Team Name
WHUIR

Subtasks
Temporal Intent Disambiguation (TID) Subtask (English)

Keywords
Temporal intention, SVR, Classification, Feature description

1. INTRODUCTION
Users express their temporal need through queries and several explicit research work on user log [1,2,3,4] showed that there is a proportion of queries have the temporal intent. Temporal query intent classification (TQIC) is the first step to research on temporal query studies and temporal query intent disambiguation (TID) is the upgraded task for TQIC. As a result, our work for attending NTCIR-12 TID subtask was mainly based on the research work or research findings of NTCIR-11 TID subtasks with the basic idea of classification.

Features are the main part of the classification and can be extracted from the query itself or the retrieved documents of the query. In this work, we only choose the features in query itself for the task. There are several reasons for it.

According to the overview[5] of the NTCIR-11 Temporalis task, teams studying on features in query itself outperformed than those team studying on features in documents. Yu et al. extracted time gap, core verb tense and name entity in query with logistic regression classifier to reach the highest average precision at 0.74 for four temporal class[6]. Shah et al. used query length, number of verbs, year information in the query and reached the highest precision for atemporal class[7].

We have some discussion for it. Of course it is hard to distinguish temporal intention in the query by merely studying words in the query and that is the main reason for studying the features in the retrieved the documents. However, most retrieved documents data is gathering by other search engines at other times which are different from real search scenario when the query dataset is constructed. Search engines these days may feature the function to rerank the retrieved documents according to relevance feedback and the retrieved results may vary from time to time. As a result, using external retrieved documents from other search engines are not an accurate way for this task.

The temporal intention will be classified more precise if user logs from the same search engine under the same search scenario is supported. By analyzing the user query click record, browsing history, we may get some hints for his or her temporal intention at that time. Nevertheless, it is hard to get the user logs.

Based on the facts and discussion listed above, we considered nineteen features in total from query itself only and used all the features for SVR in different parameter sets and chose the best three sets on the dry run data for the formal run.

The rest of this paper is organized as follows. Section 2 describes the features used in this work. Experiments and insights from three runs are the subject of Section 3. Finally we conclude in Section 4. Some acknowledgments are listed in Section 5.

2. METHODOLOGY
In this section, we present the framework used to temporal intent disambiguation subtask. We describe the procedures and the features used in this task.

2.1 SVR based approach
Given a topic, our goal is to output four probable values for each temporal class. As a result, this task can be regarded as the updated task for classification, except that this task has to output the real value for each class and classification is to output a class. Based on ideas upon, we completed the task with the basic idea of classification and manipulated it by Support Vector Regression algorithm. The whole procedure is as follows:

(1) data pre-processing

Compared to text, query is short and not much pre-processing need to done. In this work, the experiment mainly lowercase all the words, remove the punctuations, ultimately stored in a specific format to facilitate further feature extraction in use.

(2) feature extraction

We extract the corresponding features described in 2.2 on the processed data in (1) and organizes the feature sets in the corresponding format for classifier.

(3) classifier construction

Most classifier can only be divided into two categories, and this task is divided into four temporal categories. As a result, one-to-many method was applied to change the multiple classification analysis into binary classification. We established classifier for each category, that is to say, a past-other classifier will be trained if the probable value for past is needed for a given query.

(4) training and testing

Given the training data, the classifier will output the probable value for a given class and each query will have four values for four temporal classes. Then the normalization will be needed to ensure the four probably value be added to sum of 1. The normalized is done by dividing the sum of four values.

After getting all the results, cosine similarity and absolute loss will be used to evaluate the effectiveness of our system. The best training parameter sets will be used on test data for our three runs.

2.2 Features description

We extracted seven type features from the query itself and they can be expressed in nineteen detailed features for the classifier (Table 1).

2.2.1 name entity

Some name entities may have itself has time tendency. For example, the query "*when did Neil Armstrong die*" contains a name entity Neil Armstrong. Astronaut Neil Armstrong was born in 1930, died in 2012, so the retrieved documents about him should be more focus on the period between 1930 and 2012. Compared to queries do not contain name entity, queries containing entity can reflect temporal intention to some extent.

The feature *No_NER* counts the number of name entities in the query.

2.2.2 query length

Query length means the number of words in the query. In order to reflect the query temporal intention, users may use more words which is additional to describe the query content to express his or her time need. For example, for the query "*Martin Luther King day*", search engine can return a list of documents about the festival across many periods. But for the query "*Martin Luther King day 2013*", the user explicitly pointed out that he or she only need to find information related to the festival in 2013. The latter query is longer than the first one.

For calculating *Q_Len*, though stop-words will be removed before counting query length for common text processing process. However, we did not do stop-word remover and calculate the original length of the query because that query is very short in general.

2.2.3 numbers to express year

If the query contains some numbers to express year, the query has temporal intention definitely, e.g. "*Martin Luther King day 2013*", "*2013 calendar printable*", "*2012 movies*". This feature can explain the temporal intentions, but can not distinguish the temporal intention from past, present or future intentions.

Feature *isYear* outputs 0 or 1 to indicates whether the query contains some numbers to express year.

2.2.4 core verb tense

This feature was put forward by [6]. The most typical grammatical feature in English is the tense. Different tenses can distinguish between past, present and future. Again as English features clauses, a sentence may contain more than one verb. But only the verb tense in the main clause can reflect the really tense of the whole sentence. For example query "*when did Neil Armstrong die*" contains two verbs: did and die, one for the past and one for present. It is difficult to judge tense in the query only through the part of speech recognition. But people who are proficient in English can know did is real verb through the analysis sentence structure. As a result, a syntactic tree structure analysis is required before part of speech recognition.

Feature *No_CV* calculates the number of the verbs in the query. 0 or 1 will be assigned to Feature *P_CV*, *R_CV* and *F_CV* to point out whether the core verb is past tense, present tense or future tense respectively.

2.2.5 dominant keyword

The concept of dominant keyword were first discussed in the experiment discussion part of Shah et al.'s work[7] and we chose it as a strong feature for temporal intent disambiguation. Domain words refer to the most frequent words shown in certain category. By doing the post-analysis of the experiment results, they found some words are shown frequently in four temporal types. Thus we got the assumption to use the domain words as the feature.

The domain words detection is based on word listed in [7]. Feature *No_DW_P*, *No_DW_R*, *No_DW_F*, *No_DW_A* are the the number of domain words belonging to past, recency, future and atemporal categories, respectively.

2.2.6 time gap

Time gap is a frequent used feature in the similar tasks and it is refers to the time difference between the query submitted to search engines and the query contained in the time lag. For example, query "*Martin Luther King day 2013*" containing time of 2013, and assuming the query is submitted in 2015, the class of time gap is past obviously.

Featur *No_TG* counts the number of time gap in the query and feature *No_TG_P*, *No_TG_R* and *No_TG_F* count the number of the time gap in past, recency and future repectievely.

2.2.7 temporal words

This feature is based on an external dictionary TempoWordNet[8] to see if the query words belong to the past, present, and future or irrelevant category. TempoWordNet is based on WordNet[9] and word is given the probability value for the four temporal categories (past, recency, future and atemporal).

In this paper, if the value for a class is not 0 for a class, we think the word is in this temporal class. Feature *TWDic_P*, *TWDic_R*, *TWDic_F* and *TWDic_F* record the number of words in the query has the class for past, recency, future and atemporal repectively.

Table 1: Overall features considered for temporal query intent classification

Types	Features	Description
(1)	No_NER	number of name entity
(2)	Q_Len	query length
(3)	isYear	year information
(4)	No_CV	number of verbs

	P_CV	core verb presented in past tense
	R_CV	core verb presented in present tense
	F_CV	core verb presented in future tense
(5)	No_DW_P	number of past domain words
	No_DW_R	number of present domain words
	No_DW_F	number of future domain words
	No_DW_A	number of atemporal domain words
(6)	No_TG	number of time gap
	No_TG_P	number of time gap in past class
	No_TG_R	number of time gap in recency class
	No_TG_F	number of time gap in future class
(7)	TWDic_P	number of words in past based on TempoWordNet
	TWDic_R	number of words in recency based on TempoWordNet
	TWDic_F	number of words in future based on TempoWordNet
	TWDic_A	number of words in atemporal based on TempoWordNet

3. EXPERIMENT

3.1 Experimental setup

Many features are extracted on established off-the-shelf components. We mainly used Standord CoreNLP[1] to lemmatize words, to detect name entity, part of speech of words, standard time expression in the query and to parse sentence structure. Then we calculate name entity, query length, numbers to express year, core verb sense and time gap features based on Stanford CoreNLP results. Other features, e.g. domainant keywords and temporal words are extracted on word list[7] and TWnH-1.0[2] respectively.

We used SVR implementation in the toolkit LIBSVM[3] to do the feature selection and classification model training and testing. There are two parameters can be choose for SVR, namely svm type (epsilon-SVR, nu-SVR) and kernel type (linear, polynomial, radial basis, sigmoid). We used the parameter sets on the dry run data and chose the top three parameter sets for the formal run according to average per-class absolute loss and cosine similarity between the two probability vectors.

3.2 Results

Table 2 shows average per-class absolute loss and average cosine similarity between the two probability vectors. Three runs used the same feature sets but are different in parameters of LIBSVM.

Table 2 TID results

RUN ID	**svm type**	**kernel type**	**average absolute loss**	**average cosine similarity**
WHU-TID-E-1	epsilon-SVR	sigmoid	0.2662	0.6196
WHU-TID-E-2	nu-SVR	linear	0.2921	0.5225
WHU-TID-E-3	nu-SVR	polynomial	0.2520	0.6933

[1] http://nlp.stanford.edu/software/corenlp.shtml

[2] https://tempowordnet.greyc.fr/download_TWn.html

[3] https://www.csie.ntu.edu.tw/~cjlin/libsvm/

4. CONCLUSION

This paper describes our work for Temporal Intent Disambiguation subtask of NITCIR-12 Temporalia. Our work was mainly based on the research work of NTCIR-11 TID subtasks with the basic idea of classification. Based on the facts and discussion in the paper, we considered nineteen features in total from query itself only and used all the features for SVR in different parameter sets and chose the best three sets on the dry run data for the formal run. We submitted three runs in total.

Limiatation.

5. ACKNOWLEDGEMENTS

This work is supported by the National Natural Science Foundation of China (Grant No. 71173164).

6. REFERENCES

[1] Nunes, S., Ribeiro, C., & David, G. 2008. Use of temporal expressions in web search. *In Advances in Information Retrieval*. Springer Berlin Heidelberg. pp. 580-584.

[2] Campos, R., Dias, G., & Jorge, A. M. 2011. What is the temporal value of web snippets?. *Temporal Web Analytics Workshop*. ACM. 9-16.

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

[4] Metzler, D., Jones, R., Peng, F., & Zhang, R. 2009. Improving search relevance for implicitly temporal queries. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*. ACM. 700-701.

[5] Joho, H., Jatowt, A., Blanco, R., Naka, H., & Yamamoto, S. 2014. Overview of NTCIR-11 Temporal Information Access (Temporalia) Task. *In Proceedings of the NTCIR-12 Conference on Evaluation of Information Access Technologies*.

[6] Yu, H., Kang, X., & Ren, F. 2014. TUTA1 at the NTCIR-11 Temporalia Task. I*n Proceedings of the NTCIR-12 Conference on Evaluation of Information Access Technologies*.

[7] Shah, A., Shah, D., & Majumder, P. 2014. Andd7@ NTCIR-11 Temporal Information Access Task. *In Proceedings of the NTCIR-12 Conference on Evaluation of Information Access Technologies.*

[8] Dias, G. H., Hasanuzzaman, M., Ferrari, S., & Mathet, Y. (2014, April). Tempowordnet for sentence time tagging. *In Proceedings of the companion publication of the 23rd international conference on World wide web companion*. 833-838.

[9] Miller, G. A. 1995. WordNet: a lexical database for English. *Communications of the ACM*, 38(11), 39-41.