DUT-NLP-CH @ NTCIR-12 Temporalia Task

Chinese Temporal Query Disambiguation

Jiahuan Pei*, Degen Huang, Jianjun Ma,
Dingxin Song, Leyuan Sang

Dalian University of Technology

Introduction

- Motivation

  Extension of TQIC task

- Main Challenges

1. Lack of explicit temporal information
2. No query log available
3. Temporal intent may change over time
4. Temporal intent ambiguities

Datasets

- Training Data
  1. 52 dry run quires released by NTCIR-12
  2. 300 formal run quires from NTCIR-11 TQIC subtask
  3. 503 queries extracted from SogouQ log data
- Testing data

  300 testing queries from NTCIR-12 TID subtask

Past	Recency	Future	Atemporal	Total
0.13	0.16	0.07	0.64	1

Table 1. Average distribution of four temporal intent classes

Approach

- Overview
  1. A classification problem with probability output since each query has a distributional tagging vector of four temporal classes.
  2. Our overall method relies on well-designed features and well-established classifiers.

- Basic Steps
  1. Chinese Segmentation, POS tagging, Name Entity Recognizer and Parser by Stanford Corenlp tookit, Temporal Expression Recgnition by HeidelTime
  2. Feature selection based on preprocessed results
  3. Classification provided by sklearn, a machine learning module in Python

Feature List

- Explicit Time Gaps
- Word-based Probability Distribution
- Temporal Trigger Word Features
- Others Explicit Textual Features
- Implicit time Gaps from Google Trends

Explicit Features

Implicit Features

Feature List

- Explicit Time Gaps
  - Why?
    - Indicating the user’s temporal information directly
  - How?
    1. HeidelTime to recognize the temporal expression (TE)
    2. Designing Rules to map TE value to time gap features
  - Examples
    - <TE value=‘FUTURE_REF’> 近期 </TE> 油价 上涨 → FUTURE_REF
    - <TE value=‘2012-04’> 4月</TE> 工作汇报 → PAST_REF

Feature List

- Word-based Probability Distribution
  - Why?
    - Difficulty in selecting trigger words manually
    - Trigger word temporal intents diversity
  - How?
    - Vector representation $\vec{v} = \left(P_{past}, P_{recency}, P_{future}, P_{atemporal}\right)$
    - Conditional probability for the i^{th} class

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

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

Feature List

- Temporal Trigger Word Features
  - Why?
    - Typical temporal trigger words rather than conjugations of verbs that refer to temporal information
  - How?
    - Constructing trigger candidate *T* from training words if $P\left(w_j|C_i\right) > thres$ （following eq. same as previous page）

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

    - Manually selecting basic trigger sets *T'* from *T*
    - Extending *T' to T''* via word vector clustering
    - Manually filtering T'' again

Feature List

- Others Explicit Textual Features
  - centerWord
  - posOfCenterWord
  - validQueryLength
  - numOfNER
  - numOfNotChWords
  - isNounFreg

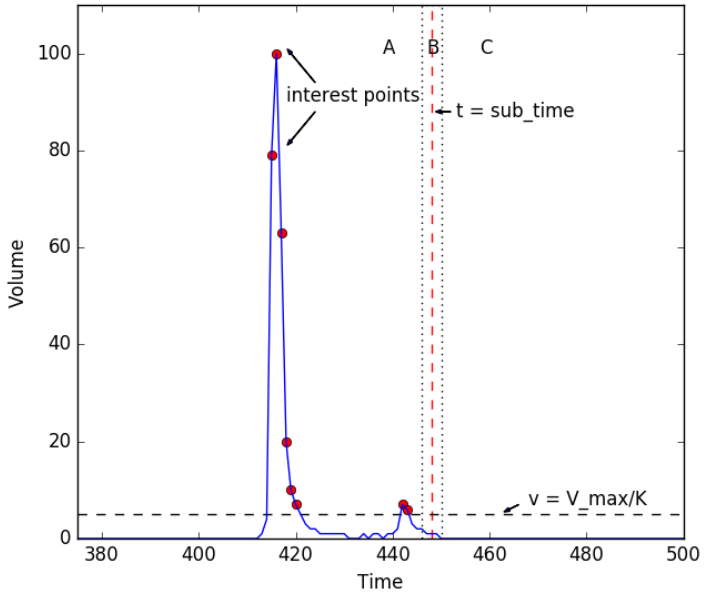
Feature List

- Implicit Time Gap Features
  - Why?
    - Explicit temporal information is absolutely rare
  - How?
    - Preprocessing
      - Downloading & resampling *Google Trends* data
    - Implicit time gap extraction
      - Time-series prediction→ ARMA model
      - Classification via Rens' model → P_{QoT}, P_{OQ}, P_{AMQ} & P_{PMQ}
        - QoT: Query without Time Intent
        - OQ: Query with One Time Interval Intent
        - AMQ: Query with Aperiodic Time Intervals Intent
        - PMQ: Query with Periodic Time Intervals Intent

Feature List

- Implicit Time Gap Features
  - Classification Mapping function →

$$\begin{cases} Gt_{atemporal} = P_{QoT} \\ Gt_{past}, Gt_{recency}, Gt_{future} = \text{linear SVC (Rens' features+avg(len))} \end{cases}$$



Formal Run Results

- C-SVC
  C-Support Vector Classification with a linear kernel function and a default C-value
- LR
  Logistic Regression model with $l1$ penalty
- RF
  Random Forest model with balanced class weights

Run	Model	AvgCosin	AvgAbsLoss
1	C-SVC	**0.8135**	0.1728
2	LR	0.8066	0.1854
3	RF	0.8116	**0.1710**

Table 2. Formal run results based on C-SVC, LR and RF model

Models Comparison

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

Table 3. Comparison among different models based on time gap features and temporal trigger word features

Feature Selection

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**

Table 4. Feature Selection based on LDA model

Conclusion

- Summary
  - Two types of features
  - Three models for formal run
  - Further comparison among models & features
- Our best run
  - LDA model
  - Features
    - Time gap
    - Word-based probability distribution vector
    - Temporal trigger word
    - Google Trends' time gap
    - Center word and its Part-of-speech.
- Future work
    - More features based time-series
    - Features from retrieval documents
    - Query embedding

Thanks for attention!