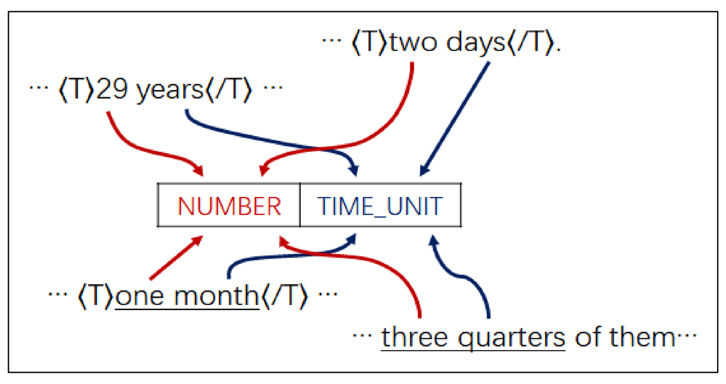
A Pattern-based Approach to Recognizing Time Expressions

Wentao Ding(wtding@smail.nju.edu.cn), **Guanji Gao**, **Linfeng Shi** and **Yuzhong Qu**(yzqu@nju.edu.cn)
National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China

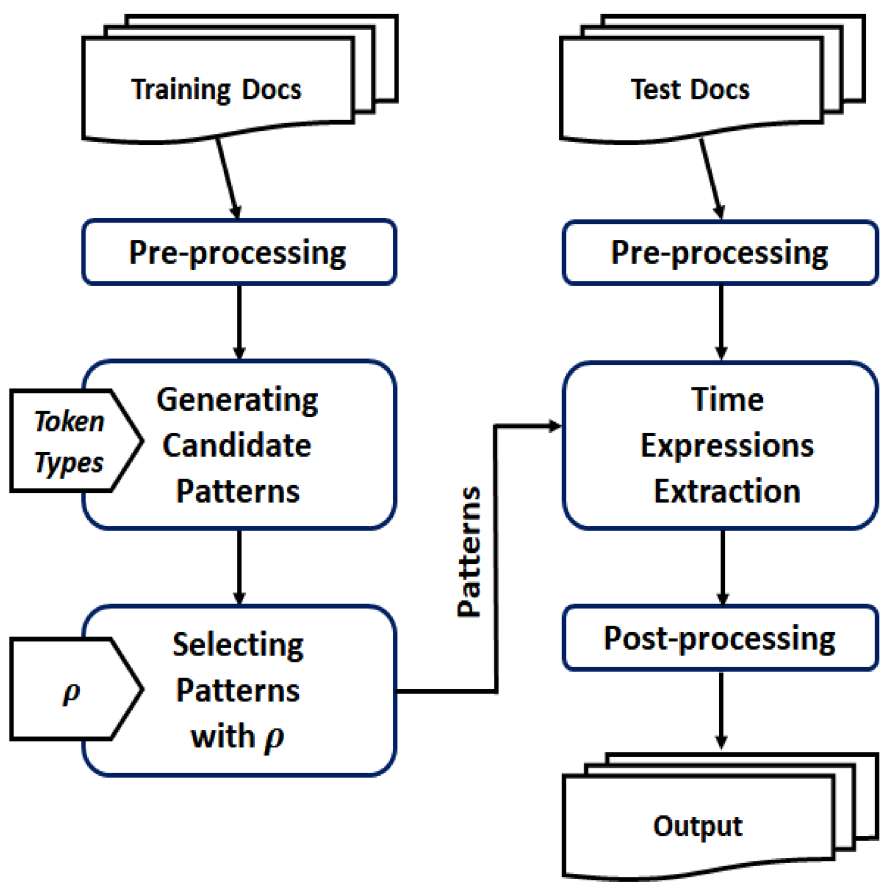
Introduction

- Newest state-of-the-art time expression recognizing approaches are mainly **black-boxed** or based on **heuristic rules**, which leads to the difficulty in understanding.
- Classic rule-based approaches rely on **deterministic rules hand-engineered by experts**.
- Previous work has shown the power of token types in recognizing.
  - Sequential type patterns can be used for extracting time expressions.
  - But the generality of token types also bring mistakes.



*Is it possible to **select an appropriate subset of all generated patterns**, to achieve a **good performance on recognizing** time expressions, and meanwhile provides an **adjustability on limiting the total mistakes** for fitting different precision-recall demands of various applications?*

Framework



- **Pre-processing**

Transforming documents to lemma/pos-tag token sequences, dealing with some special cases like "5days", "Valentine Day".

- **Generating Candidate Patterns**

Given the training documents $D_{training}$, automatically generate a pattern set P by abstracting each token to corresponding types.

- **Selecting Patterns with ρ**

Selecting an appropriate subset Q from P to maximize the correct token strings matched by Q while limiting the number of total mistakes caused by Q. A parameter ρ is introduced to loosely bound the total mistakes.

- **Extracting Time Expressions**

Use selected patterns to extract all matching strings from D_{test}.

- **Post-processing**

Merging adjacent and overlapped expressions, recognizing time ranges which depend on nearby expressions (e.g. "1957 and 58").

Pattern Generation

- **Candidate Pattern Generation**

we collect the set E of time expressions from $D_{training}$, then replacing each token in by its corresponding token type to get sequential patterns.

Candidate Pattern Set： $P = \{pattern(e) | e \in E\}$

- **Token Types**

Our type system contains 32 fine-grained types classified from the perspective of POS-tags and semantic functions. Most of the types and their corresponding regexs are collected from SUTime and SynTime.

  - **Untyped Tokens**: we let the untyped tokens remain unchanged. In other words, we dynamically create one-token types for them.

Name	Content
DEFINITE_DET.	(the | this | that | these | those)/W?DT
EARLY_LATE	(earlier | later)/RBR? |(before | after | ago | early | late)/RB
TIME_UNIT	(second | minute | hour | ... | century | era)/NNP?S?

Pattern Selection

Problem Statement

Given the candidate pattern set P, training documents D, time expression set E and a control parameter ρ, **Select a subset $\boldsymbol{Q} \subseteq \boldsymbol{P}$ to maximize $\boldsymbol{Gain(Q)}$ with a constraint $\boldsymbol{Cost(Q) \leq B}$.**

- The gain and cost is measured by strings matched by each pattern.

$$S_D(p) = \{str_{\{i,l,r\}} \mid \wedge_{k=0}^{r-l} p_k = type(token_{\{i,k\}})\}$$

- **The gain of Q** is defined as a coverage function on the time expression set E.

$$Gain(Q) = \sum_{e \in E} \max_{p \in Q} Cov(p,e) \qquad Cov(p,e) = \begin{cases} \frac{|p|}{|e|} & \exists s \in S_D(p) \text{ is a substring of } E \\ 1 & \text{otherwise} \end{cases}$$

- **The cost of Q** is defined as summing up the mistakes caused by each pattern.

$$Cost(Q) = \sum_{p \in Q} Cost(p) \qquad Cost(p) = \sum_{s \in S_D(p)} \begin{cases} 0 & \exists e \in E,\ s \text{ is a substring of } e \\ 1 & \text{otherwise} \end{cases}$$

- The total cost should not exceed a bound $B = |E| \cdot (1 - \rho)$, where $\rho \in [0,1]$

Optimization

We apply an greedy algorithm which has been proved to have a approximation ratio ~0.35 and a time complexity $O(|P|^2|E|)$.

Algorithm 1: Algorithm for Pattern Selection

Input: the candidate pattern set P, time expression set E and the functions Cov, $Cost$
Output: A subset $Q \subseteq P$ denotes the selected patterns
$Q_1 \leftarrow \text{GreedySelect}(\emptyset)$;
$p^* = \text{argmax}_{p \in P}\{Gain(\{p\})\}$;
if $Cost(p^*) > |E| \cdot (1 - \rho)$ **then**
  return Q_1;
$Q_2 \leftarrow \text{GreedySelect}(\{p^*\})$;
if $Gain(Q_1) > Gain(Q_2)$ **then**
  return Q_1;
else
  return Q_2;

Algorithm 2: GreedySelect

Input: Initial selected patterns I, and all of the input of algorithm 1
Output: A subset $Q \subseteq P$ denotes the selected patterns
$R \leftarrow P - I$;
$Q \leftarrow I$;
repeat
  $p_i \in R \leftarrow \text{argmax}\{\frac{(Gain(Q \cup \{p_i\}) - Gain(Q))}{Cost(p_i) + \varepsilon}\}$;
  if $Cost(Q \cup \{p_i\}) \leq |E| \cdot (1 - \rho)$ **then**
    $Q \leftarrow Q \cup \{p_i\}$;
  $R \leftarrow R \setminus \{p_i\}$;
until $R = \emptyset$;
return Q

Evaluation

Dataset

- **TempEval-3**: corpus of newswire text consists 183(train)+22(test) documents.
- **WikiWars**: 17(train)+5(test) Wikipedia history articles about war.
- **Tweets**: 742(train)+200(test) tweets of which each contains at least one time expression.

Evaluation Metrics

- **Strict Match F1 Score (SM F1)**: the F1 value in terms that the extracted timex *strictly matches* the gold timex
- **Relaxed Match F1 Score (RM F1)**: the F1 value in terms that the extracted timex *overlaps with* the gold timex

Comparison Methods

- **Rule-based approaches:** HeidelTime SUTime SynTime
- **Black-box learning approaches**: ClearTK UWTime TOMN

Experimental Results

TempEval-3

Method	SM F1	RM F1
HeidelTime	81.34%	90.30%
SUTime	79.57%	90.32%
ClearTK	82.70%	90.23%
UWTime	83.10%	91.40%
SynTime	**92.09%**	**94.96%**
TOMN	***91.58%***	***94.51%***
PTime*	84.25%	91.58%

WikiWars

Method	SM F1	RM F1
HeidelTime	83.10%	90.30%
SUTime	76.64%	92.55%
ClearTK	***83.82%***	93.56%
UWTime	83.00%	92.30%
SynTime	80.11%	92.29%
TOMN	82.47%	***94.25%***
PTime*	**87.21%**	**96.37%**

Tweets

Method	SM F1	RM F1
HeidelTime	82.05%	86.71%
SUTime	78.50%	89.77%
ClearTK	80.54%	89.59%
UWTime	78.59%	87.06%
SynTime	91.74%	***95.87%***
TOMN	***92.56%***	95.45%
PTime*	**93.50%**	**98.53%**

Research Support: National Natural Science Foundation of China (No. 61772264).
Project Homepage: http://ws.nju.edu.cn/ptime