Automatic Temporal Expression Normalization with Reference Time Dynamic-Choosing

Xujian Zhao, Peiquan Jin, and Lihua Yue
School of Computer Science and Technology
University of Science and Technology of China
nonozxj@mail.ustc.edu.cn, {jpq,llyue}@ustc.edu.cn

Abstract

Temporal expressions in texts contain significant temporal information. Understanding temporal information is very useful in many NLP applications, such as information extraction, documents summarization and question answering. Therefore, the temporal expression normalization which is used for transforming temporal expressions to temporal information has absorbed many researchers' attentions. But previous works, whatever the hand-crafted rules-based or the machine-learnt rules-based, all can not address the actual problem about temporal reference in real texts effectively. More specifically, the reference time choosing mechanism employed by these works is not adaptable to the universal implicit times in normalization. Aiming at this issue, we introduce a new reference time choosing mechanism for temporal expression normalization, called reference time dynamic-choosing, which assigns the appropriate reference times to different classes of implicit temporal expressions dynamically when normalizing. And then, the solution to temporal expression defuzzification by scenario dependences among temporal expressions is discussed. Finally, we evaluate the system on a substantial corpus collected by Chinese news articles and obtained more promising results than compared methods.

1 Introduction

Temporal expression normalization is very important for temporal information processing because it is in charge of transforming temporal expressions in surface texts to temporal information behind surface texts. Temporal information is defined as the knowledge about time or duration, which can be abstracted into some objects defined as temporal attributes in TIMEX2 Standard [Ferro et al., 2005]. Human being can take temporal relation reasoning and anchor events on the time line with this information. Meanwhile, temporal expressions are defined as chunks of texts which convey explicit or implicit temporal information. So TERN evaluation plan[1] gives the task of temporal expression normalization that is annotating the appropriate temporal attributes for each temporal expression in texts. For example, a simple temporal expression, "May 1, 2009", can be normalized as <TIMEX2 VAL = "2009-05-01"> May 1, 2009 </TIMEX2>.

Unfortunately, temporal expressions in real texts are more complicated because they contain a large number of Implicit Times besides Explicit Times. Here,

(1) *Explicit Time*: Explicit Time can directly be laid in the timeline. Basically, it is a direct entry in the timeline and need not to be transformed. E.g., "May 1, 2009".

(2) *Implicit Time*: Implicit Time can be mapped as an entry in the timeline with help of real contexts and some predefined knowledge and need to be transformed. E.g., "May 1", "tomorrow" and "two day ago".

Consequently, temporal expression normalization is mainly aiming at Implicit Times that

[1] http://timex2.mitre.org/tern.html

Coling 2010: Poster Volume, pages 1498–1506,
Beijing, August 2010

need to be transformed with referring to some specific times. However, the previous works on temporal expression normalization which basically adopt two mechanisms for choosing reference time, static time-value [Mani and Wilson, 2000; Wu et al., 2005; Wu et al., 2005] and static choosing-rules [Vozov, 2001; Jang et al., 2004; Lin et al., 2008], are not compatible with the real texts. The static time-value mechanism refers to taking the report time or publication time of the document as the fixed reference time for the whole text when normalizing. And the static choosing-rules mechanism means that the machine always uses fixed rules by contexts to choose reference time for each Implicit Time whatever its temporal semantics is. The rule based on the nearest narrative time [Lin et al., 2008] is the most typical and effective one, which uses the nearest narrative time in text above as the reference time all the while. But actually the context-free assumption or the rote operation is unsuitable for universal Implicit Times. For example, a news report is as Figure 1 shows:

> (*Beijing, May 6, 2009*) B company took over A company totally on March 8, 2000. After one week, B company listed in Hong Kong, and became the first listed company in that industry. However, owing to the decision-making mistakes in the leadership and the company later poor management, B company got into debt for several hundred million dollars, and was forced to announce bankruptcy this Monday.

Figure 1. Example of news reports

For these two Implicit Times in the text, “after one week” and “this Monday”, obviously there will be critical conflicts when using these two mechanisms referred above to choose reference time. The static time-value is unsuited for the “after one week”, and “this Monday” makes mistakes when taking the nearest narrative time (i.e., “after one week”) as the reference time to normalize according to the static choosing-rules.

Motivated by this issue, we propose a new reference time choosing mechanism for temporal expression normalization. Firstly, we segment the Implicit Time into two parts, modifier and temporal noun, and then train a classifier with referential features of these two parts to classify Implicit Times. As a result, we choose the corresponding reference time for each temporal expression depending on its class when normalizing. Meanwhile an acceptable defuzzification solution is introduced to normalize fuzzy times in our method. And the contributions of this paper are:

(1) We introduce a simple but effective reference time choosing method, called dynamic-choosing mechanism, which can choose the appropriate reference times automatically for universal Implicit Times as well as be compatible with the dynamically changeable contexts.

(2) Going beyond traditional normalization approaches, we develop a new way to deal with the defuzzification in order to figure out the *fuzzy reference time* (the reference time is vague or has imprecise start and end in timeline), which makes the normalization robust and improve the accuracy of reference times.

The rest of this paper is organized as follows. Section 2 discusses related works. In section 3 we describe the reference time dynamic-choosing mechanism. The temporal expression normalization is presented in section 4. Section 5 gives the description about experiments and evaluations. Finally, conclusion and future work are presented in section 6.

2 Related Work

In general, several research works on normalizing temporal expressions, which are involved in English [Mani and Wilson, 2000], French [Vozov, 2001], Spanish [Saquete et al., 2002], Korean [Jang et al., 2004] and Chinese [Wu et al., 2005; Lin et al., 2008], have been reported in recent years. Among them, the handcrafted rules-based methods [Saquete et al., 2002; Schilder and Habel, 2001; Mani and Wilson, 2000] can deal with various temporal expressions, but the procedure to build a robust rules system is quite time-consuming. With regard to the machine learning for normalization [Jang et al., 2004; Wu et al., 2005; Vicente-Diez et al., 2008], the potential task is the classification which is deciding one explanation of a temporal expression from several alternatives.

However, these works on temporal expression normalization do not give an effective reference time choosing method for Implicit Times in real texts. More specifically, the pioneer work by Lacarides [1992] investigated various contextual effects on different temporal-reference relations. Then Hitzeman et al. [1995] discussed the reference-choosing taking into account the effects of tense, aspect, temporal adverbials and rhetorical relations. Dorr and Gaasterland [2002] presented the enhanced one in addition considering the connecting words. But they are theoretical in nature and heavily dependent on languages. Currently, the static time-value mechanism [Mani and Wilson, 2000; Wu et al., 2005; Wu et al., 2005] and the static choosing-rules mechanism [Vozov, 2001; Jang et al., 2004; Lin et al., 2008] for reference time choosing are applied into some systems widely. Nevertheless, as the discussion in section 1, these two ways are not adaptable to universal Implicit Times. In addition, Vicente-Diez et al. [2008; 2009] discussed the reference date for relative times, but the alternative rules are not effective in experiments. Lin et al. [2008] considered the condition that there is no report time or publication time when choosing reference time.

Referring to the defuzzification, TIMEX2 Standard [Ferro et al., 2005] takes the X placeholder to express fuzzy times' value, so the related works [Jang et al., 2004; Lin et al., 2008; Vicente-Diez and Martinez, 2009] almost follow this vague expressing way. However, this method can not address the actual situation that the fuzzy time is referred to by other times. Based on the human cognitive psychology, Anderson et al. [1983] presented a classical scenario-time shifting model that discussed the time includes the fuzzy time is the clue to scenario shifting when people reading. Inspired by this issue and based on our experiments, we find all times in a same scenario own strong dependences in temporal granularity, which can effectively help us determine granularity in defuzzification. And more details are discussed in section 4.2.

Aiming at solving these challenges above, we establish a temporal expression normalization system for real texts, which improves the accuracy of temporal reference normalization remarkably by the dynamic-choosing mechanism.

3 Reference Time Dynamic-choosing Mechanism

3.1 Referential feature in Implicit Time

In this paper, we define the Implicit Time consists of the modifier and the temporal noun which is modified by modifiers. And here we extend the modifier based on the TIMEX2 Standard, which include verb, conjunction, adverb and preposition that quantify or modify temporal nouns. For example, "ten days" is a temporal noun, but "ten days ago" is modified after adding the modifier "ago".

Meanwhile we find no matter how long or how many modifiers modify the temporal noun, the whole temporal expression holds the original temporal reference inferred from the temporal noun. Moreover, the key point of normalizing temporal expressions is choosing the appropriate reference time according to the real context rather than deciding the right direction or computing the measurable offset. For instance, with regard to these two Implicit Times in Figure 1, "after one week" and "this Monday", we can achieve the referential direction easily from the modifiers through some mapping rules. Meanwhile, the offsets are able to be understood directly by machine with pattern matching. But for the reference time, we must build the context-depending reference reasoning to trace it. The reference link is described as Figure 2 shows.

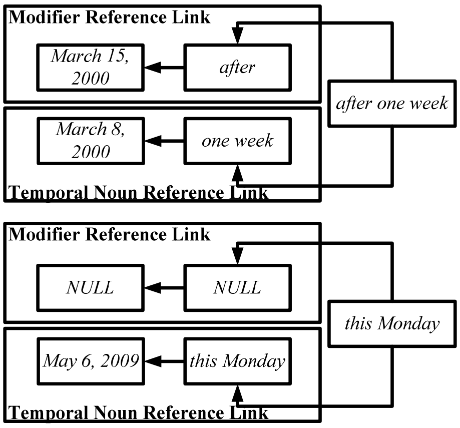

Figure 2. Example of reference link

From the reference reasoning, we can see the full temporal reference comes from two parts: modifier reference and temporal noun reference. Because the former is inferred from the latter, the temporal noun reference reasoning plays more important roles in normalization. In other words, the reference reasoning of the whole Implicit Time strongly depends on the temporal noun. Furthermore, in the practical operation, we indeed take the report time or the nearest narrative time in text above as the reference time of the temporal noun when normalizing a whole Implicit Time. Therefore, we consider the classification of the Implicit Time based on the classes of temporal noun's reference time. Basically we tag the Implicit Time as the same class as its temporal noun's under classifying temporal nouns into two classes according to the referential feature.

(1) *Global Temporal Noun*: Global Temporal Noun takes the report time or publication time of the document as the reference time when normalizing. Basically, it is independent of the local context.

(2) *Local Temporal Noun*: Local Temporal Noun makes reference to the nearest narrative time in text above in normalization due to depending on the local context.

Table 1 and 2 give some examples of Global Temporal Noun and Local Temporal Noun in real texts.

Consequently, here we denote the Implicit Time consists of the Global Temporal Noun and the modifier(s) by *Global Time* or GT, and accordingly the Local Temporal Noun corresponds to *Local Time* or LT.

Class	Sub-class	Examples
Global Temporal Noun	year	*last year*
	month	*next month*
	day	*this Friday*
	hour	*tonight*
	fuzzy	*lately*[2]

Table 1. Common Global Temporal Noun expressions

Class	Sub-class	Examples
Local Temporal Noun	year	*that year*
	month	*October*
	day	*the second day*
	hour	*morning*
	fuzzy	*then*
	duration	*one month*

Table 2. Common Local Temporal Noun expressions

3.2 Naïve Bayesian Classifier

A variety of machine learning classifiers are designed to resolve the classification problem, such as SVM classifier, ME classifier and the Decision Tree family. But the performance of these classifiers is greatly depending on the features selection. Based on the observation and analysis in our experiments, we find the referential feature holds in the temporal noun is hard to express with some explicit denotations. For example, "that year" and "this year" are nearly identical in surface feature, but the former is locally context-depending while the latter is locally context-free. So the Naïve Bayesian Classifier that assumes independence among feature denotations is suitable to be applied to our method.

We take the single word in the temporal noun as the object attribute x_i after removing the Explicit Time in the whole text. Given the class label c, the classifier learns the conditional probability of each attribute x_i from training data. Meanwhile, achieving the practical instance of X, classification is then performed by applying Bayes rules to compute the probability of c, and then predicting the class with the highest posterior probability.

$$c_o = \arg\max_c grade(c \mid x_1, x_2, \cdots, x_n) \quad x_i \in X \qquad (1)$$

$$grade(c \mid x_1, x_2, \cdots, x_n) = \frac{p(c \mid x_1, x_2, \cdots, x_n)}{p(\bar{c} \mid x_1, x_2, \cdots, x_n)} \qquad (2)$$

Applying Bayes rules to (2), we have:

$$grade(c \mid x_1, x_2, \cdots, x_n) = \frac{p(x_1, x_2, \cdots, x_n \mid c)p(c)}{p(x_1, x_2, \cdots, x_n \mid \bar{c})p(\bar{c})} = \frac{\prod_{i=1}^{n} p(x_i \mid c)p(c)}{\prod_{i=1}^{n} p(x_i \mid \bar{c})p(\bar{c})} \qquad (3)$$

Actually, we estimate $p(x_i \mid c)$ and $p(x_i \mid \bar{c})$ by Maximum Likelihood Estimation (MLE) from training data with Dirichlet Smoothing method [Li et al., 2004].

$$p(x_i \mid c) = \frac{num(x_i, c) + \mu}{\sum_{j=1}^{n} num(x_j, c) + \mu \cdot n} \qquad (4)$$

$$p(x_i \mid \bar{c}) = \frac{num(x_i, \bar{c}) + \mu}{\sum_{j=1}^{n} num(x_j, \bar{c}) + \mu \cdot n} \qquad (5)$$

3.3 Reference Time Choosing

In our approach, there is a reference time table is used to hold full reference times for the whole text, and we need to update and maintain it dynamically after each normalizing processing.

[2] Some single temporal adverbs are taken as temporal noun, e.g. recently, currently and so on.

The time table consists of two parts: Global Reference Time and Local Reference Time.

(1) *Global Reference Time*: Global Reference Time (GRT) is a type of reference time which is referred to by the Global Time. Specifically, it is the report time or the publication time of the document.

(2) *Local Reference Time*: Local Reference Time (LRT) is made reference to by the Local Time. It will be updated dynamically after each normalizing.

Figure 3 shows a sample of the interaction between reference times and target times.

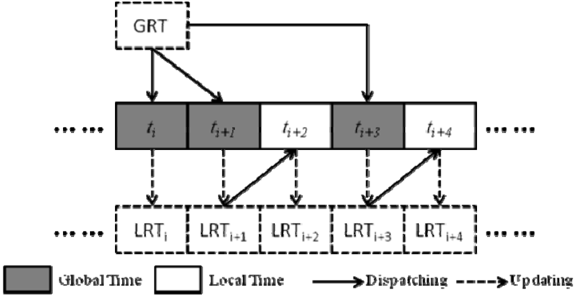

Figure 3. Interaction between reference times and target times

In Figure 3, we notice that different classes of time dynamically and automatically choose references based on their respective classes rather than do it using the fixed value or the inconsiderate rule under the static mechanism. And the reference time table is updated in real time finishing each normalizing, which makes the temporal situation compliable with dynamically changeable contexts.

4 Temporal Expression Normalization

4.1 Basic Normalizing Algorithm

In the beginning, we need to achieve the report time (RT) or the publication time (PT) of the document to initialize the GRT and LRT. Additionally, the fuzzy time can be referred to by other times in the normalization, but we must solve the defuzzification problem before taking it as the reference time. With respect to this issue, we will discuss it in the next section. Consequently, the practical normalizing algorithm is as follows.

```
Algorithm: TimeNormalize
Input: temporal expression $t_i$ in text
Output: regular time list TList
Begin
  //initialize the GRT and LRT with RT or PT of this document
  GRT ← Initialize (RT|PT)
  LRT ← Initialize (RT|PT)
  for each $t_i$ in text do
    //segment $t_i$ into modifier and temporal noun
    $t_i$'← SegmentTemporal ($t_i$)
    if IsExplicitTime ($t_i$) is true
      //update the time table with $t_i$
      LRT ← UpdateTime ($t_i$)
      //insert $t_i$ into regular time list directly
      TList ← InsertList ($t_i$)
    else
      if IsLocalTime ($t_i$') is true
        //retrieve the latest LRT from time table and then normalize $t_i$'
        $T_i$ ← RegularizeTemporal ($t_i$', LRT)
      else
        //retrieve GRT from time table and then normalize $t_i$'
        $T_i$ ← RegularizeTemporal ($t_i$', GRT)
      LRT ← UpdateTime ($T_i$)
      end if
    TList ← InsertList ($T_i$)
    end if
  return TList
End Begin
```

4.2 Temporal Expression Defuzzification

In general, the defuzzification for fuzzy times faces two problems: deciding granularity and choosing offset. Here we introduce some knowledge on the human cognitive psychology and the empirical method to figure out these two issues respectively. Based on the scenario-time shifting model referred in related works, we get the conclusion that once the scenario is shifting, the time is shifting. More specifically, the time shifting is reflected in the temporal granularity between two different scenarios. So referring to writers, they will choose a few temporal expressions own the same granularity to render the coherent temporal dimensionality in one scenario in order to avoid generating improper scenario shifting for readers. Figure 4 describes the variation process of the temporal granularity between two different scenarios through scenario-time shifting.

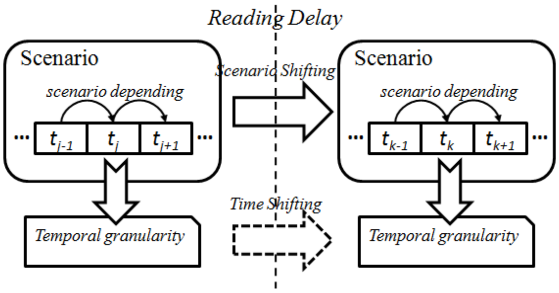

Figure 4. Variation process of temporal granularity

As conveyed in Figure 4, temporal expressions in the same scenario are constrained by the scenario depending. Hence fuzzy times should keep pace with scenario-correlative times in granularity. For example, two sentences in different scenarios:

"He was in Hong Kong *yesterday*, but now he is in Beijing."

"He was in Hong Kong *last year*, but now he is in Beijing."

Obviously, the first "now" means "today" in that scenario, and it has the same temporal granularity with "yesterday". Meanwhile it will be more appropriate for the second "now" choosing "year" as the temporal granularity than choosing "day" because of the dependence to the scenario-correlative time. In narrative, the paragraph is normally considered as the minimum unit of the scenario, so scenario-correlative relations should stand on the one paragraph at least.

But for the first temporal expression in the paragraph, we need to think about two specific conditions: when it appears in the first paragraph and in the non-first paragraph if it is a fuzzy time. Because there is no scenario shifting to the first paragraph, we employ a dictionary to initialize the algorithm when the first time included in the first paragraph is fuzzy time. The defuzzification process is outlined as follows.

```
Algorithm: TempGranuarityDefuzzify
Input: temporal expression $t_i$ in text
Output: precise-granularity time $t_i'$
Begin
  //obtain the granularity of $t_i$
  granularity ← GetGranularity ($t_i$)
  //decide whether $t_i$ is a fuzzy time
  if granularity is not null
    $t_i'$ ← $t_i$
  else
    if $t_i$ is not first temporal expression in current paragraph
      //assign the former's granularity to the $t_i$
      granularity ← GetGranularity ($t_{i-1}$)
    else
      if $t_i$ is not included in first paragraph
        //decide which granularity is assigned to $t_i$ between $t_{i-1}$ and $t_{i+1}$
        if IsSameGranularity (GetGranularity ($t_{i-1}$), GetGranularity ($t_{i+1}$)) is not true
          granularity ← GetGranularity (CoarserCompare ($t_{i-1}$, $t_{i+1}$))
        else
          granularity ← GetGranularity ($t_{i+1}$)
        end if
      else
        //retrieve default granularity from dictionary
        granularity ← FindGranularityInDict ($t_i$)
      end if
    end if
    //retrieve default offset from dictionary
    offset ← FindOffsetInDict ($t_i$)
    //update and intact all temporal attributes of $t_i$
    $t_i'$ ← ModifyTimeAttribute ($t_i$, granularity, offset)
  end if
  return $t_i'$
End Begin
```

It's possible for the first temporal expression to correlate with forenamed times in last paragraph in real texts, so we choose the coarser granularity for the fuzzy time when appearing conflicts in granularity between the last temporal expression and the next temporal expression. Additionally, an empirical fuzzy time dictionary is constructed as the default in order to figure out the offset problem. For example, "lately" is denoted in dictionary as below.

Lately
Common synonyms: *recently*, *latterly*, *late*, *of late*
Default granularity: *day*
Default offset: *7 units*

Finishing the defuzzification for the whole text, the basic normalizing algorithm is evoked then. In the experiments, we find that the temporal expression defuzzified can clearly improve the accuracy of reference times besides discovering the implicit temporal information much more.

5 Evaluation

5.1 Setup

Because the normalization for temporal expressions is independent of the language [Wilson et al., 2001], we take the formal Chinese news[3] as

[3] People's Daily news corpus (January, 1998), supported by Institute of Computational Linguistics (ICL), Peking University.

the experimental corpus, which consist of 3148 Chinese news articles. The data collection contains 2,816,612 characters/967,884 words and 21,176 manually annotated temporal nouns. Among this corpus, 2518 articles (80%) include 13,835 temporal expressions are used as training data for the classification, and the rest (20%) as test data. Then the whole corpus is tested for the normalization. Event-anchored expressions are relevant with a specific event and it is hard to represent the exact meaning of them, so in our system, event-anchored expressions are not normalized.

5.2 Results

Results on Implicit Times classification: We firstly choose some temporal expressions classified in advance by crafted, and manually extend them in expressing patterns as the original training samples. For example, "last month" will extend to "this month" and "next month", which all belong to Global Times. Actually there are only 16,104 temporal expressions in our experiment because integrated temporal expressions in corpus are segmented into several parts, and we combine them together again before operating. Using classifier trained by training data, we get 2,264 Global Times and 998 Local Times from testing collections, where there are 1,705 Global Times and 804 Local Times are correct respectively by manual statistics. Table 3 gives the details of classification.

From the experiment data, we find the precision and the recall almost below 80%, and the classification performance is not expected. The reason is that we do not consider some special application situations beforehand, which result in classifying errors. For example, the Global Time should be taken as the Local Time when it appears in the dialog or speech that marks boundaries by a pair of quotation marks. So we introduce some revising patches shown in Table 4 to deal with this issue. Here the second and the fourth patches make corresponding temporal expressions be treated as non-target times that need not be processed. In addition, Time Set and Non-Specific are taken as the other classes except the Implicit Time and the Explicit Time. The final results with revising patches are shown in Table 5. Obviously, revising patches make the classification be more adapted for the real texts, and the performance evaluation is promising.

Class	**#Correct**	**Precision (%)**	**Recall (%)**	**F-measure (%)**
Global Time	1705	75.31	78.64	76.94
Local Time	804	80.56	79.45	80.00
Sum/Average	2509	77.94	79.05	78.47

Table 3. Results of classification

ID	**Patch Type**	**Patterns**	**Operations**
1	Dialog/Speech	"XXX"	Time → LT
2	Book/Movie	《XXX》 *XXX*	Be omitted
3	Time Set	quantifier + XXX	Time → others
4	Proper Noun	e.g. October Revolution	Be omitted
5	Non-Specific	e.g. childhood	Time → others

Table 4. Revising patches for classification

Class	**#Correct**	**Precision (%)**	**Recall (%)**	**F-measure (%)**
Global Time	1879	88.69	86.67	87.67
Local Time	918	90.55	90.71	90.63
Sum/Average	2797	89.62	88.69	89.15

Table 5. Results of classification with revising patches

Results on temporal expression normalization: For evaluating our algorithm objectively, we compare the experiment result with other two methods on the same testing corpus. The first compared method which is adopted in many traditional systems [Li et al., 2004; Wu et al., 2005] applies the static time-value mechanism to determine the reference time. The nearest narrative time [Lin et al., 2008; Vicente-Diez and Martinez, 2009] that represents the static choosing-rules mechanism is taken as the second compared method. Table 6 presents the results.

Method	**Average referent updating/ article**	**Accuracy (%)**	**Errors**	
			Referent (%)	**Others (%)**
STVM	0	68.42	22.84	8.74
SCRM	7.8	76.19	11.25	12.56
Our method	4.2	83.55	7.33	9.12

*STVM: Static Time-Value Mechanism
SCRM: Static Choosing-Rules Mechanism

Table 6. Results of normalization

The data shows that our method exceeds the compared ones evidently. The accuracy increases by 15.13% at most, and the errors by referent decreases by 3.92% at least. In contrast to the SCRM, we avoid the limitation that SCRM only concentrates on the nearest distance for choosing referent. Meanwhile, because the SCRM pays no attention to the normalization for fuzzy temporal expressions, the error by others (e.g. granularity) is greater than ours. Additionally, the STVM method applies the report time or the publication time of the document as the reference time for the whole text, so there is no referent updating in process. We mark all errors as referent errors as long as they involve with false reference time in results analysis, therefore, the STVM gets the highest referent errors ratio.

With respect to the defuzzification, we evaluate it on fuzzy times separately. All defuzzified fuzzy times are assessed by human, and then decided whether they are acceptable to the context. The evaluation results are shown in Table 7.

Type	**#Acceptable**	**Acceptable ratio (%)**	**As referent (%)**
Global Time	687	80.14	18.39
Local Time	159	92.61	6.43
Sum/Average	846	86.38	12.41

Table 7. Evaluations on temporal expression defuzzification

For the fuzzy temporal expression in Local Time, it is much fewer and easier than the one in Global Time in number and expression respectively, so the defuzzification in Local Times achieves more expected results. On the other hand, the fuzzy time in Global Time is often the first temporal expression in the first paragraph, and the corresponding dictionary-based method certainly affects the experiment results. According to the percentages that the temporal expressions defuzzified successfully account for in the all reference times, it demonstrates that the defuzzification makes contributions to the referential normalization besides discovering the internal temporal information in the fuzzy time.

6 Conclusion

In this paper, we present an approach to automatically normalizing temporal expressions under the reference time dynamic-choosing mechanism. The referential feature in temporal nouns is applied to classify Implicit Times. Based on this, different classes of times can be normalized according to their respective classes. Meanwhile, we introduce the scenario-time shifting model to deal with the defuzzification problem. The experiment shows that our approach achieves more promising evaluation results, and makes the automatic normalization more adaptable to real texts than the prior works. However, the neglect on the event-anchored expression certainly restricts the whole system in applications, so the event-anchored expression will be our research focus in future.

Acknowledgement This work is supported by the National Natural Science Foundation of China under the grant no. 60776801 and 70803001, the Open Projects Program of National Laboratory of Pattern Recognition (no.20090029), the Key Laboratory of Advanced Information Science and Network Technology of Beijing (no. xdxx1005), the National High Technology Research and Development Program ("863" Program) of China (Grant No. 2009AA12Z204).

References

Anderson, A., Garrod, S.C. and Sandford, A.J. 1983. The Accessibility of Pronominal Antecedents As a Function of Episode Shifts in Narrative Text. *Quarterly Journal of Experimental Psychology*, 35a, pp. 427-440.

Dorr, B. and Gaasterland, T. 2002. Constraints on the Generation of Tense, Aspect, and Connecting Words from Temporal Expressions. *Technical Report CS-TR-4391,UMIACS-TR-2002-71, LAMP-TR-091*, University of Maryland, College Park, MD, 2002.

Ferro, L., Gerber, L., Mani, I., et al. 2005. TIDES 2005 Standard for the Annotation of Temporal Expressions [EB/O L]. (2005-09)
http: //timex2.mitre.org.

Hitzeman, J., Moens, M. and Grover, C. 1995. Algorithms for Analyzing the Temporal Structure of Discourse. In *Proceedings of the 7th European Meeting of the Association for Computational Linguistics*, pp. 253-260.

Jang, S.B., Baldwin, J. and Mani, I. 2004. Automatic TIMEX2 Tagging of Korean News. *ACM Transactions on Asian Language Information processing* 3(1), 51-65.

Lascarides, A., Asher, N. and Oberlander, J. 1992. Inferring Discourse Relations in Context. In *Proceedings of the 30th Meeting of the Association for Computational Linguistics*, pp. 1-8.

Li, W.J., Wong, K.F., Cao, G.H. et al. 2004. Applying Machine Learning to Chinese Temporal Relation Resolution. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, pp. 582-588.

Lin, J., Cao, D.F. and Yuan, C.F. 2008. Automatic TIMEX2 tagging of Chinese temporal information. *Journal of Tsinghua University* 48(1), 117-120.

Mani, I. and Wilson, G. 2000. Robust Temporal Processing of News. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, pp. 69-76.

Saquete, E., Martinez-Barco, Patricio and Munoz, R. 2002. Recognizing and Tagging Temporal Expressions in Spanish, in *Proceedings of Workshop on Annotation Standards for Temporal Information in Natural Language*, pp. 44-51.

Schilder, F. and Habel, C. 2001. From Temporal Expressions to Temporal Information: Semantic Tagging of News Messages. In *Proceedings of the ACL 2001 Workshop on Temporal and Spatial Information Processing*, pp. 65-72.

Vazov, N. 2001. A System for Extraction of Temporal Expressions from French Texts based on Syntactic and Semantic Constraints. In *Proceedings of the ACL Workshop on Temporal and Spatial Information Processing*, pp. 96-103.

Vicente-Díez, M.T., Samy, D. and Martínez, P. 2008. An Empirical Approach to a Preliminary Successful identification and Resolution of Temporal Expressions in Spanish News Corpora. In *Proceedings of the Sixth International Language Resources and Evaluation*, pp. 2153-2158.

Vicente-Díez, M.T., Martinez, P. 2009. Temporal Semantics Extraction for Improving Web Search. In *Proceedings of the Workshop on Database and Expert Systems Application*, pp. 69-73.

Wilson, G., Mani, I., Sundheim, B. et al. 2001. A Multilingual Approach to Annotating and Extracting Temporal Information. In *Proceedings of the Workshop on Temporal and Spatial Information Processing*, pp. 1-7.

Wu, M.L., Li, W.J., Lu, Q. and Li, B.L. 2005. CTEMP: A Chinese temporal parser for extracting and normalizing temporal information. In *Proceedings of International Joint Conference on Natural Language Processing*, pp. 694-706.

Wu, M.L., Li, W.J., Chen, Q. and Lu, Q. 2005. Normalizing Chinese Temporal Expressions with Multi-label Classification. In *Proceedings of Natural Language Processing and Knowledge Engineering*, pp. 318-323.