ParsTime: Rule-Based Extraction and Normalization of Persian Temporal Expressions

Behrooz Mansouri [1,2], Mohammad Sadegh Zahedi [1,2], Ricardo Campos[3], Mojgan Farhoodi[1] and Maseud Rahgozar[2]

[1] Information Technology Faculty, Iran Telecommunication Research Center, Tehran, Iran
[2] Database Research Group, Control and Intelligent Processing Center of Excellence, School of Electrical and Computer Engineering, University of Tehran
[3] Polytechnic Institute of Tomar, LIAAD – INESC TEC, Tomar, Portugal
{b.mansouri,s.zahedi,farhoodi}@itrc.ac.ir,
rahgozar@ut.ac.ir,ricardo.campos@ipt.pt

Abstract. Extraction and normalization of temporal expressions are essential for many NLP tasks. While a considerable effort has been put on this task over the last few years, most of the research has been conducted on the English domain, and only a few works have been developed on other languages. In this paper, we present ParsTime, a tagger for temporal expressions in Persian (Farsi) documents. ParsTime is a rule-based system that extracts and normalizes Persian temporal expressions according to the TIMEX3 annotation standard. Our experimental results show that ParsTime can identify temporal expressions in Persian texts with an F1-score 0.89. As an additional contribution we make available our code to the research community.

Keywords: Temporal Tagger, Time Normalization, Pattern Matching

1 Introduction

Extracting temporal information from text plays an important role in natural language processing and information retrieval tasks such as text summarization and temporal query classification [1]. It can also be used by search engines for tasks like query auto completion, result ranking or query classification [2], to name but a few.

The first step to extract temporal information from text, is to recognize temporal expressions and to convert them into a standard annotation. To conduct this, temporal taggers are usually used. Over the last few years, a considerable number of different time taggers were proposed for the English language. GUTime [3] was developed by the Georgetown University as part of TARSQI toolkit, with the purpose to improve question answering systems towards temporally-based questions. HeidelTime [4], introduced by Strötgen and Gertz, is probably one of the most well-known approaches and the best performing system in task A for English of the TempEval-2 challenge (http://semeval2.fbk.eu). Another well-known system is that of Chang and Manning, who presented SU-Time [5] as part of Stanford CoreNLP. Besides English, temporal taggers have been addressed for only a few other languages. Li et al. [6] for example,

proposed, a Chinese temporal parser for extracting and normalizing temporal information using HeidelTime architecture. Strötgen et al. [7] in turn, introduced the temporal tagger for Arabic, Italian, Spanish and Vietnamese using the same architecture. Many researches in NLP and information retrieval have developed research considering the Persian language [8,9,10]. However, no one so far, has developed a temporal tagger devoted to this important Indo-European language, which is one the most dominant in the Middle East, spoken in several countries like Iran, Tajikistan and Afghanistan. In this paper, we propose a first attempt on this matter, by introducing a temporal tagger for detecting Persian temporal expressions within a text. ParsTime is a rule-based temporal tagger that can identify and normalize different Persian temporal expressions within a text with high precision. These expressions may refer to different types such as date, time, duration or set. Besides the expression type, the calendar type (Georgian, Hijri or Jalali) is also identified by ParsTime, an additional challenge when compared to Western Languages which only refer to a single calendar type. ParsTime was evaluated under two different datasets achieving an F1-score of 0.89, which is in line with the results obtained by other temporal taggers in a diversity of languages. As a further outcome of our research, we also make available an implementation of our method in Java which we made publicly available on GitHub[1].

2 Method Description

2.1 Types of Temporal Expressions

TIMEX3 [13] is a well-known annotation scheme for temporal expressions which is usually used in this kind of task. In TIMEX3, each temporal expression has two attributes, Type and Value. A Type can be one of the four types: Date, Time, Duration and Set. A Value, instead, corresponds to a normalized temporal value, which is a normalized way of referring to a temporal expression. For example, for the Time "یک ربع به هفت" (a quarter to seven) a normalized value would be 2017-10-25T06:45. (With reference date as 2017-10-25). Like in many other languages, Persian also contains a huge variety of temporal expressions. In this work, we consider the following four types defined in TIMEX3 [13]:

- Date: This expression points to a calendar time. It may point to a day, week, month, season or year. ParsTime is able to extract both explicit temporal expressions (e.g., "اکتبر ۲۰۱۷" (October 2017)), as well as relative temporal expressions (e.g., "روز گذشته" (yesterday)). While extracting and normalizing the first type is straightforward, for the latter we need the reference time. As a rule-of-thumb we will consider document publish time, or the last date extracted from the text, whenever the first one cannot be found.
- Time: This kind of expression refers to a time of the day. It might refer to an exact time, such as "۹ صبح ۱ اکتبر" (9:00 am October 1), or an approximate time, such as, "صبح شنبه" (Saturday morning).

[1] https://github.com/BehroozMansouri/ParsTime

- Duration: This type of expression describes a duration (interval). The start and end points of duration might be exactly mentioned, as in expression "از اکتبر تا نوامبر" (From October to November). The duration itself, may be exact or an approximation. For example, "طی این هفته" (during this week) has implicit boundaries, while the expression "تا ساعاتی دیگر" (until next few hours) does not have a certain end point.
- Set: This expression represents periodic temporal sets. In other words, it refers to a temporal signal that occurs on a regular basis, such as "هر دوشنبه" (every Monday). Table 1 provides an example of some temporal expressions for each of these four types with normalized TIMEX3 value.

Table 1. Persian temporal expression example with translation and a ref. time of 2017-10-25

Type	Example	Translation	Normalization Value
Date	هفده سپتامبر ۲۰۱۷	September seventeenth 2017	2017-09-17
Time	یک ربع به هفت	a quarter to seven	2017-10-25T06:45
Interval	چهار روز آینده	the next four days	P4D
Set	عصر هر سه‌شنبه	every Tuesday afternoon	XXXX-WXX-2TAF

2.2 ParsTime Architecture

In this section, we introduce the architecture underlying the ParsTime method. In summary, it takes a text as an input and split its sentences. Each sentence is then temporally annotated. It is then matched against a list of predefined temporal patterns and normalized if it matches a pattern, before outputting it with TIMEX3 annotation. Fig. 1 indicates the workflow of ParsTime. In the following we describe each of these steps in more detail.

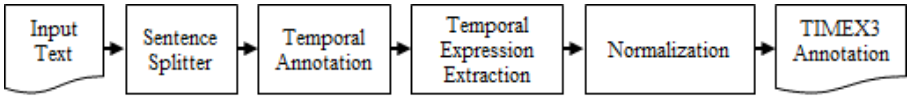

Fig. 1.ParsTime workflow.

- Pre-Processing: First, a preprocessing step is needed to prepare the input text for the coming stages. Like in many other NLP systems, the input text is first tokenized. In our method we resort to the ParsiPardaz toolkit [11].
- Temporal Annotation: Before matching the input text against predefined patterns, tokenized text coming from the previous section, is first temporally annotated with 12 predefined tags (including day, month, season, numbers, etc.) in a similar fashion as a standard PoS tagger. For instance, "ژانویه ۲۰۱۸" (January 2018) is annotated as "Month Number";"ژانویه" annotated as a "Month", and 2018 is annotated as "Number". This is an important stage, as temporal patterns will be defined (in the next stage) based on these temporal tags notations. To conduct this process, we resort to define a set of expression resources for each temporal tag. For example, for the temporal tag "Month", a list of months from "January" to "December" (in addition to a

list of months in Hijri and Jalali calendar) is defined. Each term in the input text is matched against the related list for each temporal tag.

- Temporal expression extraction: After the input text is temporally annotated, annotated tokens are matched against a list of predefined temporal expression patterns to identify temporal expressions. These patterns are defined based on the temporal tags introduced in the previous step. For instance, the defined pattern "Month+Number", can detect temporal expressions such as "ژانویه ۲۰۱۸" (January 2018). As mentioned in Section 2.1, four types of temporal expressions can be recognized by ParsTime. For each of these categories, a considerable number of patterns is defined: 149 for Date, 74 for Time, 97 for Duration and 26 for Set, totalizing 346 temporal expression patterns. This number is considerably higher than other languages due to the complexity of the Persian language. For instance, HeidelTime [4] has defined a reduced number of 248 patterns for the English language. It should be noticed that some of the patterns are subset of others, such that, when ParsTime recognizes the shorter pattern, it continues to check if the text contains a longer pattern or not. If the longer pattern is not recognized, then the shorter one is extracted. For example, for the expression "فردا صبح" (Tomorrow morning), the pattern "RD" (Relative day) is recognized, but as we also have the pattern "RD+PD" (Relative Day + Part of Day), ParsTime continues and identifies the longer pattern.
- Normalizing: The final step of this workflow is to normalize the extracted temporal expression. Every extracted temporal expression E_i has two attributes and can be viewed as a two-tuple $E_i = \langle V_i, T_i \rangle$, where V_i is the normalized value that indicates the temporal semantic of an expression as defined by TIMEX3 standard, and T_i the type of temporal expression (Date, Time, Duration or Set).

3 Evaluation

To evaluate our method, we resorted to the development of two new datasets as no previous standard ground-truth could be found. Our aim, is to understand whether there is any difference in the effectiveness of our method, upon different types of input (more formal or more informal temporal expressions). For the first one, we relied on a news dataset, a kind of dataset that is usually used for this kind of tasks. For this purpose, we selected Hamshahri [12], a news dataset which covers a wide range of news in Persian language, including politics, entertainment and sports from a ten-year period, spanning from 1996 to 2006. We then randomly selected 2000 news articles domain from this dataset and asked 4 students to tag the temporal expressions using TIMEX3 annotations. An inter-rater reliability analysis using the Fleiss Kappa statistics was performed to determine consistency among the editors. Overall, the annotators obtained about 0.82 of agreement level, which represents a high agreement between editors. For the second dataset, we relied on search engine query logs, to capture a more informal nature of temporal expressions. Query logs are a useful resource of this kind of data, as search engine users usually tend to be more relaxed when specifying their temporal intents. To this purpose, we asked the very same 4 students to select 250 unique queries (totalizing 1000 queries) containing temporal expressions from the query log records of a Persian search engine, Parsijoo. For each student, a unique period of query log records was

provided and they were asked to select and annotate 250 queries (using TIMEX3 annotations) that contains at least one temporal expression. Both of these datasets are publicly available[2]. For the task of evaluating the results, we computed the precision, recall and F1-scores for the extraction of temporal expressions. For the type and value attributes, only the correctly identified temporal expressions are considered (the ratio of correct guesses for both type and value). Table 2 reports the results for each dataset.

Table 2. ParsTime performance on Hamshahri corpus and Parsijoo query log records.

	Extraction			Attribute	
Dataset	Precision	Recall	F1-score	Value	Type
Hamshahri	0.92	0.86	0.89	0.84	0.93
Parsijoo	0.91	0.85	0.88	0.82	0.92

The results achieved by ParsTime are quite satisfying and in line with the results reported by other temporal taggers for different languages (see Table 3). For both formal and informal temporal expressions, the results of ParsTime performance was nearly the same. To better understand the results we looked at the errors and noticed that, wrong input format was the main reason that affects the recall. Wrong spelling, wrong punctuation (and spacing) and the rare words used in queries were the main causes that ParsTime was unable to detect some of temporal expressions. For example, in the expression "غروبگاه شنبه" (Saturday evening), only day patterns is recognized, as the term "غروبگاه" (uncommon word meaning evening) is a rare term and was not defined in patterns. Also, our error analysis reveals that, one of the main reasons that may affect the precision of ParsTime, may be related to ambiguous words which are often common in Persian. For instance, the word "بهمن" is a name of a Persian Month but also the name of Football team. Considering this ambiguousness, in the expression "تیم بهمن سال ۹۳" (Team Bahman 93), a temporal pattern "Month+Number" is wrongly recognized.

Table 3. Effectiveness of ParsTime and other temporal taggers.

	Extraction			Attribute	
Temporal Tagger	Precision	Recall	F1-score	Value	Type
ParsTime	0.92	0.86	0.89	0.84	0.93
SuTime (English)	0.88	0.96	0.92	0.82	0.92
HeidelTime (English)	0.90	0.82	0.86	0.85	0.96
HeidelTime (Spanish)	0.96	0.84	0.90	0.85	0.87
HeidelTime (Arabic)	0.95	0.83.8	0.89	-	-

4 Conclusion

In this paper, we presented ParsTime, the first temporal tagger for extracting and normalizing Persian temporal expression from texts. ParsTime is a rule-based system that can extract different types of temporal expressions including date, time, duration and set. Our experimental results, over two newly created TIMEX3 annotated datasets,

[2] http://dbrg.ut.ac.ir/ParsTime

show that ParsTime achieved high F1-score. As an additional contribution to the research community we also make available a Java version of our method. This will enable researchers to use our system despite guaranteeing the reproducibility of our research. In future work, we plan to provide resources for detecting implicit temporal expressions, such as “Rio Olympics”, which implicitly refer to 5-21 August 2016.

Acknowledgement

This research was supported by Persian native search engine program from Iran Telecommunication Research Center (ITRC).

References

1. Campos, R., Dias, G., Jorge, A. M., & Jatowt, A. (2015). Survey of temporal information retrieval and related applications. ACM Computing Surveys (CSUR), 47(2), 15.
2. Mansouri, B., Zahedi, M. Rahgozar,M., Oroumchian, F., Campos, R. (2017). Learning Temporal Ambiguity in Web Search Queries. In ACM Conference (CIKM), pp-6-10.
3. Verhagen, M., Mani, I., Sauri, R., Knippen, R., Jang, S. B., Littman, J., ... & Pustejovsky, J. (2005). Automating temporal annotation with TARSQI. In Proceedings of the ACL 2005 on Interactive poster and demonstration sessions. Association for Computational Linguistics. pp. 81-84.
4. Strötgen, J., & Gertz, M. (2010). Heideltime: High quality rule-based extraction and normalization of temporal expressions. In Proceedings of the 5th International Workshop on Semantic Evaluation. Association for Computational Linguistics. pp. 321-324.
5. Chang, A. X., & Manning, C. D. (2012). Sutime: A library for recognizing and normalizing time expressions. In LREC .Vol. 2012, pp. 3735-3740.
6. Li, H., Strötgen, J., Zell, J., & Gertz, M. (2014). Chinese Temporal Tagging with HeidelTime. In EACL Vol. 2014, pp. 133-7.
7. Strötgen, J., Armiti, A., Van Canh, T., Zell, J., & Gertz, M. (2014). Time for more languages: Temporal tagging of Arabic, Italian, Spanish, and Vietnamese. ACM Transactions on Asian Language Information Processing (TALIP), 13(1), 1.
8. Zahedi, M., Mansouri, B., Moradkhani, S., Farhoodi, M., & Oroumchian, F. (2017). How questions are posed to a search engine? An empiricial analysis of question queries in a large scale Persian search engine log. In Web Research (ICWR), 2017 3th International Conference on). IEEE pp. 84-89.
9. Zahedi, M., Aleahmad, A., Rahgozar, M., Oroumchian, F., & Bozorgi, A. (2017). Time sensitive blog retrieval using temporal properties of queries. Journal of Information Science, 43(1), pp.103-12.
10. Mansouri, B., Zahedi, M., Rahgozar, M., & Campos, R. (2017). Detecting Seasonal Queries Using Time Series and Content Features. In Proceedings of the 2017 ACM on International Conference on the Theory of Information Retrieval ACM. pp. 279-300.
11. Sarabi, Z., Mahyar, H., & Farhoodi, M. (2013). ParsiPardaz: Persian Language Processing Toolkit. In Computer and Knowledge Engineering (ICCKE), 3th International eConference on IEEE. pp. 73-79.
12. AleAhmad, A., Amiri, H., Darrudi, E., Rahgozar, M., & Oroumchian, F. (2009). Hamshahri: A standard Persian text collection. Knowledge-Based Systems, 22(5), pp.382-387.
13. Pustejovsky, J., Castano, J. M., Ingria, R., Sauri, R., Gaizauskas, R. J., Setzer, A., ... & Radev, D. R. (2003). TimeML: Robust specification of event and temporal expressions in text. New directions in question answering, 3,pp. 28-34.