An Open Source Library for Semantic-Based Datetime Resolution

Aurélie Merlo, Denis Pasin
Jam, 80 rue de Cléry, 75002 Paris, France
{firstname}@hellojam.fr

Abstract

In this paper, we introduce an original Python implementation of datetime resolution in French, which we make available as open-source library. Our approach is based on Frame Semantics and Corpus Pattern Analysis in order to provide a precise semantic interpretation of datetime expressions. This interpretation facilitates the contextual resolution of datetime expressions in timestamp format.

1. Introduction

Jam is an artificial intelligence supervised by humans answering to questions of young French people (18-30 years old) on activities to do during their spare time (sport, movies, restaurant, travel…). Adapted to French, the AI classifies the user messages sent by text message among a list of needs. Each need is associated with a list of metadata. A metadata is a qualitative data in an incoming message. This data provides relevant informations (datetime, location, price…) in order to search for contents (restaurant, movie, air ticker…) through APIs (Booking, Yelp).

Our users communicate in natural language. We need a system able to identify and translate various possible formulations of a metadata into a data understandable by an API. For example, the system has to be able to make all these datetime constructions interoperable in timestamp format: *departure on August 26, a nice evening for Halloween, departure about August 26, I don't know when, only on September 2*.

TIMEX3 (Pustejovsky & al., 2010) and libraries using this standard annotation (GUTime (Verhagen et al., 2005), HeidelTime (Strötgen & Gertz, 2010), SUTime (Chang & Manning, 2012)) fail to provide a satisfactory contextual resolution outside explicit datetimes. For example in (1), TIMEX3 does not account for approximative datetimes:

(1) *départ vers le 8 mai* "departure about May 8"
départ vers <TIMEX3 tid="t1" type="DATE" value="XXXX-05-08">le 8 mai</TIMEX3>

TIMEX3 fails also to define a datetime against an event (*two days before Christmas*) and does not account for compositionnality (Bethard & Parker, 2016). For example in (2), datetime expression should be interpreted as a date associated with a time and not as two TIMEX3 tags:

(2) *vendredi 10 avril à 15h* "Friday April 10 at 3PM"
<TIMEX3 tid="t3" type="DATE" value="XXXX-04-10">vendredi 10 avril</TIMEX3> à <TIMEX3 tid="t6" type="TIME" value="XXXX-XX-XXT15:00">15h</TIMEX3>

The relevance of the content that we send to our users depends heavily on the understanding we have of their needs. Faced with the lack of precision and the incompleteness of TIMEX3, we developed a python library for identification, interpretation and contextual resolution in timestamp format of French datetime constructions. Datetime resolution is not a new task in NLP. Our approach is original because it is based on the works of the Frame Semantics (Fillmore 1976) and the Corpus Pattern Analysis (CPA) (Hank 2004). Our library provides a rich semantic interpretation of datetime constructions which enables to resolve them in timestamp format. The library and its documentation are available under MIT license and on https://github.com/blackbirdco/time_extract.

2. Semantic-Based Approach

2.1. Frame Semantics and Corpus Pattern Analysis (CPA)

In order to take into account datetime compositionnality, we based our approach on Frame Semantics. According this semantic theory, the meaning of a word or an expression is determined above all by its context. A frame consists of Frame Elements (FEs) and a list of lexical units which call a frame. In FrameNet, we identified two frames that describe datetime:
- Calendric_Unit : Unit FE (*Tuesday*), Whole FE (*Tuesday of next week)* and Relative_time FE (*next Tuesday*)
- Time_vector : Direction FE and Distance FE (*in two days, two days ago*).
Frame Semantics does not offer a complete description of all datetime constructions and it only lists the FEs without explain how they are combined. Furthermore, Relative_time FE lacks precision for contextual resolution. *Next Tuesday* cannot be resolved similar to *Last Tuesday* in timestamp format.

CPA, influenced by Frame Semantics, is a method of corpus analysis for lexicographic purpose. Meaning is viewed as a pattern. Each pattern consists of ontological categories called semantic type. A semantic type is a general category representing a shared property by concepts. With CPA it is possible to describe datetime constructions using patterns. For example, we can describe the frame timeModified corresponding to a timeline modification using this pattern:

datetime expressions: *before Monday, after Monday*
frame: timelineModified
pattern: before or after dayOfWeek

dayOfWeek is a semantic type because all of its concepts (Monday, Tuesday, Wednesday…) share the property to be modified by a timeline modifier (before, after, around, not, only).

2.2. Datetime Ontology

We extracted from our database 1200 incoming messages for annotation. The purpose of annotation was to identify frames and patterns for datetime expressions. We show succinctly the highest frames and patterns of our datetime ontology[1]:

- datetimeGeneralUsageTerm
  - datetimeElementTerm
    - dateTerm
      - explicitDateTerm
        - pattern: dayOfWeekItem (*Monday)*
        - pattern: dayOfWeekItem monthValueItem dayValueItem (*Monday August 28*)
        - pattern: dayOrdinalItem dayOfWeekItem monthValueItem YearValueItem (*second Monday of January 2017*)
        - …
      - modifiedDateTerm
        - beforeDateTerm (*before Monday*)
        - afterDateTerm (*after Monday*)
        - approximativeDateTerm (*around Monday*)
        - constraintDateTerm (*only on Monday*)
        - negativeDateTerm (*not on Monday*)
      - vectorDateTerm
        - futureVectorDateTerm (*in two days*)
        - pastVectorDateTerm (*two days ago*)
      - urgencyDateTerm (*it's urgent*)
      - indeterminateDateTerm (*I don't know when*)
      - anyDateTerm (*no matter when*)
    - hourTerm
      - explicitHourTerm (*at 3 PM*)
      - modifiedHourTerm (*before 3 PM*)

[1] The complete ontology is in the library documentation.

        - vectorHourTerm (*in two hours*)
        - indeterminateHourTerm
        - anyHourTerm
    - datetimeGroupElementTerm (*August 7 and 10*)
    - datetimeAlternativeElementTerm (*August 7 or in two days*)
- datetimeTravelUsageTerm
    - departureDatetimeTerm (*departure on Monday*)
    - returnDatetimeTerm (*return before August 10 2016*)

3. Implementation

3.1. Semantic interpretation module

Intepretation module takes an user message in French (Il y a quoi au ciné lundi vers 15h? “what are the movies on Monday around 3PM?”). The message is cleaned (addition of missing spaces, multi-word expressions tagging) and lemmatized with TreeTagger. Lemmas in datetime semantic lexicon are tagged with a semantic type (Monday [firstDayOfWeekItem]). The semantic tagged message is used as an input for a semantic rule-based chunker. We created a semantic grammar which represents frames and patterns above. The output of interpretation module is a semantic tree with an extracted datetime expression:

```
(metadata
 (datetimeTerm
  (datetimeGeneralUsageTerm
   (datetimeElementTerm
    (dateTerm
     (unmodifiedDateTerm
      (calendarDateTerm
       (dayOfWeekTerm lundi “Monday”/firstDayOfWeekItem))))
    (hourTerm
     (modifiedHourTerm
      (approximativeHourTerm
       vers “around”/approximativeItem
       (unmodifiedHourTerm
        (numericHourTerm 15/numeralItem heure “hour”/hourUnitItem)))))))))
```

Our approach has several advantages. It reports the semantic compositionnality of datetime expressions. It allows to infer implicit informations (August 7 or 8 (August implicitly)). Our approach is quite flexible to give a semantic interpretation even the most complex datetime expressions. The contextual resolution in timestamp format is facilitated by the richness of tree analysis. Finally, the library adaptation to other languages can be done only by translation of semantic lexicons, the grammar being language-independant.

3.2. Contextual resolution module

As shown in the introduction, talking to other machines, services or customers require a standard format that is understandable by them. You can't really use "next Monday" to talk to them but "1468250471" will work. That's why we introduced a parser from information tree to JSON. So Monday is stored as the timestamp of Monday:

```
{"text": "next Monday", "timestamp": 1468250471}.
```

We also added multiple fields to enhance this timestamp and make it more "precise". It's quite easy to do when the date is well expressed ("September 12, 2017") but, with oral-like text it's almost never the case.

Relative datetime case: *I’m looking for a bar this evening*. When does the evening start exactly? We solve the problem of relative datetime by adding a "approximate" filed and a "radius" one. We also fixed times for each kind of "special" times like evening which is 9pm. Regarding to the 'time-object' granularity we have different radiuses. For example, this evening is hourly based so the radius is 2 hours around the decided time:

{"text": "this evening", "timestamp": 1468250471 (the time of today at 9pm), "approximate": true, radius: 7200}.

Group and alternative datetime case: *I'm free this Monday or next one*. We should look for something either this Monday or one week later? We solve the problems of group and alternative datetimes by adding an id to each "time" object. Then the script returns a super object containing an array of time object and a logical string:

{"times":
[{"id": 1, "text": "this Monday", "timestamp": 1468250471 (the time of next monday), "approximate": false},
{"id": 2, "text": "next one", "timestamp": 1468855271 (the time of next Monday), "approximate": false}],
"logic": "1 || 2"}.

Urgency datetime case. There's sometimes notion of urgency in datetimes: *I need a cab right now*. We forward through a particular field:

{"times": [{"id": 1, "text": "right now", "timestamp": 1468250471 (the time of next Monday), "approximate": false, urgent: true}], "logic": "1"}.

4. A brief evaluation

We evaluated our approach with a corpus of 639 incoming messages extracted from our database. The messages contain datetime expressions. The aim of the evaluation was to measure semantic interpretation efficacy checking frames and patterns in analysis tree. For the evaluation, the measures of precision, recall and F1 are used. We obtained a high recall (0,994), precision (0,814) and F1 score (0,895) in interpreting datetime expressions.

5. Conclusion

We presented an open-source library for datetime resolution in French. In future work, we plan (i) to add new temporal notions as period, duration and set, (ii) to make a more detailed evaluation and (iii) to translate the semantic lexicons in order to adapt library for other languages.

References

Bethard S. and Parker J. (2016). "A Semantically Compositional Annotation Scheme for Time Normalization". In: *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*.

Chang A. X. and Manning C. D. (2012). "SUTIME: A Library for Recognizing and Normalizing Time Expressions". In: *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC 2012)*.

Fillmore C. (1976). "Frame semantics and the nature of language". In: *Annals of the New York Academy of Sciences: Conference on the Origin and Development of Language and Speech*.

Hank P. (2004). "Corpus Pattern Analysis". In: *Proceedings of the 11th Euralex*.

Pustejovsky J., Lee K., Bunt H., and Romary L. (2010). "ISO-TimeML: An International Standard for Semantic Annotation". In: *Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC 2010)*.

Strötgen J. and Gertz M. (2010). "Heideltime: high quality rule-based extraction and normalization of temporal expressions". In: *Proceedings of the 5th International Workshop on Semantic Evaluation (SemEval 2010)*.

Verhagen M., Mani I., Sauri R., Knippen R., Jang S. B., Littman J., Rumshisky A., Philipps J. and Pustejovsky J. (2005). "Automating temporal annotation with TARSQI". In: *Proceedings of the 43th Annual Meeting of The Association for Computational Linguistics (ACL 2005)*.