Temporal expression normalisation in natural language texts

Michele Filannino
School of Computer Science
The University of Manchester, UK
filannim@cs.man.ac.uk

Abstract

Automatic annotation of temporal expressions is a research challenge of great interest in the field of information extraction. In this report, I describe a novel rule-based architecture, built on top of a pre-existing system, which is able to normalise temporal expressions detected in English texts. Gold standard temporally-annotated resources are limited in size and this makes research difficult. The proposed system outperforms the state-of-the-art systems with respect to TempEval-2 Shared Task (value attribute) and achieves substantially better results with respect to the pre-existing system on top of which it has been developed. I will also introduce a new free corpus consisting of 2822 unique annotated temporal expressions. Both the corpus and the system are freely available on-line.

Keywords: information extraction, temporal expression, text mining, natural language processing

1 Introduction

In many domains, the possibility of using and interpreting temporal aspects and events is important in order to organise information. Temporal knowledge allows people to filter information and even infer temporal flows of events. Furthermore, it permits an improving of intelligence for question answering, information retrieval and information filtering systems.

A temporal expression [7], also called timex, refers to every natural language phrase that denotes a temporal entity such as an interval or an instant. For example, in a sentence like “Italian prime minister Mario Monti said yesterday that the reform has been very successful.” the phrase “yesterday” is actually a temporal expression. Timexes elicit a binding between the natural language domain and the time domain because it is always possible to represent such expressions as a time point, interval or set using ISO 8601 standard. Temporal expressions could be of three different types [1]: fully-qualified, deictic and anaphoric.

Fully-qualified A temporal expression is fully-qualified with respect to the binding when all the information required to infer a point in the time domain are fully included inside the expression. In this category the following expressions falls: March 15 2001, 21st July 1985 or 31/04/2011. Fully-qualified expressions are the easiest to detect because of their rigid lexical form.

Deictic In this case, inferring the binding with the time domain necessarily requires to take into account the time of utterance (when the document has been written or when the speech has been given). Deictic expressions could not be properly associated to a precise time without that information. Typical deictic temporal expressions are: today, yesterday, last Sunday and two months ago.

Anaphoric These expressions can be mapped to a precise point in the time domain only taking into account temporal expressions previously mentioned in the text or during the speech. Examples of this category are: March 15, the next week, Saturday. The only difference between deictic and anaphoric expressions is the location of the temporal reference: for deictic expressions it is the time of utterance or publication, for anaphoric expressions it is a time previously evoked in the text or speech. Anaphoric expressions constitute a future challenge for the scientific research in this field.

For the sake of completeness, another kind of categorisation [13] is also adopted in the field. It identifies

---

1http://www.cs.man.ac.uk/~filannim/

2http://www.w3.org/TR/NOTE-datetime
the possible shapes of timexes with respect to their semantics and admits the following types: time or date references, time references that anchor on another time, durations, recurring times, context-dependent times, vague references and times indicated by an event.

In my taster project I focussed on the normalisation of fully-qualified and deictic temporal expressions. I will not use the last categorisation because of the fuzziness of boundaries among types.

2 Background

The idea of annotating temporal expressions automatically from texts appeared for the first time in 1998 [3]. This topic aroused an increasing interest with the proposal of a proper temporal annotation scheme [12]. The original aim was to make the annotation phase easier with respect to the previous scheme in order to collect annotated data and use the temporal information to enhance performances of question answering systems [12]. All the most recent systems [19, 20] proposed for the temporal expressions extraction task go through two different steps: identification and normalisation. This dichotomy has become universally accepted by the research community because it makes the extraction phase easier to approach [1].

In the identification phase the effort is concentrated on how to detect properly the sub-expressions that are real temporal expressions in natural language texts. This step is usually done by using machine learning techniques. Ahn et al. [1] firstly used Conditional Random Fields [10] showing better performances with respect to a previous work [2] in which they used Support Vector Machines [4]. Poveda et al. [13] introduced a sophisticated Bootstrapping technique enhancing the recognition of temporal expressions while Mani et al. [12] used rules learned by a decision tree classifier, C4.5 [15], and Ling and Weld [11] tried Markov Logic Network [6].

The second step is the normalisation. In this phase the main goal is to interpret the expression, extract the temporal information and represent it in a proper pre-defined format. The universally accepted standard for temporal expressions annotation is TimeML [14]. It provides a specification for the representation of temporal expressions and also events. In this work the normalisation task aims at producing the proper TimeML code that correctly represents the temporal information (see Figure 1). This step is usually accomplished using rule-based approaches. Grover et al. [9] used a rule-based approach on top of a pre-existing information extraction system, whereas Strötgen and Gertz [16] produces a small set of hand-crafted rules with an ad-hoc selection algorithm. Uözaman and Allen [17] produced a rule-based normaliser focussing just on TYPE and VALUE attributes of TIMEX3 tag (the one used to represent timexes).

3 Method

The contribution of my short taster project is twofold. Firstly, I will illustrate a temporal expression corpus explicitly designed for the normalisation phase. Then I will describe the software architecture of a new normaliser built on top of a pre-existing one.

3.1 Temporal expressions corpus

Gold-standard temporally-annotated resources are very limited in general domain [5], and even less in specific ones like medical, clinical and biological [8]. In the last decade, different sources of annotated temporal expressions have been developed. Because of the rapid evolution of this research field, usually the sources differ even with respect to the annotation guidelines. This leads to the existence of different corpora not entirely compatible to each other.

The main difference among them consists in the tag used to annotate temporal expressions: TIMEX2 against TIMEX3. These two tags reflect totally different way of annotating the same temporal expressions leading to the impossibility of using both corpora at the same time.

I created a corpus of temporal expressions collecting all TIMEX3 tags in four different corpora: AQUAINT³, TimeBank 1.2⁴, WikiWars⁵ and TRIOS TimeBank v0.1⁶. I extracted from each document all the possible temporal expressions and for each one I also saved the related document creation time, the type (DATE, TIME, SET or DURATION) and the normalisation provided by the human annotators. Then I compacted the corpus removing possible duplicates. With the expression duplicates I refer to completely identical tuples, i.e. same text, same normalisation, same utterance time and same type.

I obtained a corpus of 2822 unique annotated temporal expressions. The Table 2 shows an excerpt of the corpus. Further information about the distribution of temporal expression types in it is provided in Table 1.

The corpus is freely available⁷ in CSV format using a tabulation character as delimiter.

³http://www.ldc.upenn.edu/Catalog/docs/LDC2002T31/
⁴http://www.timexportal.info/corpora-timebank12
⁵http://www.timexportal.info/wikiwars
⁶http://www.cs.rochester.edu/u/naushad/trios-timebank-corpus
⁷http://www.cs.man.ac.uk/~filannim/timex3s_corpus.csv
Italian prime minister Mario Monti said that the reform has been very successful.

Figure 1: Example of TimeML code. In the sentence there is a deictic temporal expression; “yesterday” can be correctly annotated only taking into account the document creation time (DCT).

<table>
<thead>
<tr>
<th>Temporal expression</th>
<th>Type</th>
<th>Value</th>
<th>Utterance</th>
</tr>
</thead>
<tbody>
<tr>
<td>more than two years</td>
<td>DURATION</td>
<td>P2Y</td>
<td>2011-09-26</td>
</tr>
<tr>
<td>much of 2010</td>
<td>DATE</td>
<td>FUTURE_REF</td>
<td>2011-09-26</td>
</tr>
<tr>
<td>nearly a month</td>
<td>DATE</td>
<td>P1M</td>
<td>2011-09-26</td>
</tr>
<tr>
<td>nearly an hour</td>
<td>DURATION</td>
<td>PT1H</td>
<td>1991-02-25</td>
</tr>
<tr>
<td>nearly forty years</td>
<td>DURATION</td>
<td>P40Y</td>
<td>1919-08-01</td>
</tr>
<tr>
<td>nearly four years ago</td>
<td>DATE</td>
<td>1994</td>
<td>1998-02-27:081300</td>
</tr>
<tr>
<td>nearly three years</td>
<td>DURATION</td>
<td>P3Y</td>
<td>1989-10-30</td>
</tr>
<tr>
<td>nearly two months</td>
<td>DURATION</td>
<td>P2M</td>
<td>1998-03-06:131900</td>
</tr>
<tr>
<td>nearly two months afterwards</td>
<td>DATE</td>
<td>FUTURE_REF</td>
<td>2011-09-26</td>
</tr>
<tr>
<td>nearly two weeks ago</td>
<td>DATE</td>
<td>1989-WXX</td>
<td>1989-10-30</td>
</tr>
<tr>
<td>nearly two years</td>
<td>DURATION</td>
<td>P2Y</td>
<td>1998-03-01:141100</td>
</tr>
<tr>
<td>next day</td>
<td>DATE</td>
<td>2011-09-27</td>
<td>2011-09-26</td>
</tr>
</tbody>
</table>

Table 2: Brief excerpt of the corpus.
Table 1: Distribution of TIMEX3 tags in the corpus.

<table>
<thead>
<tr>
<th>Timex type</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATE</td>
<td>2307</td>
</tr>
<tr>
<td>DURATION</td>
<td>416</td>
</tr>
<tr>
<td>TIME</td>
<td>71</td>
</tr>
<tr>
<td>SET</td>
<td>28</td>
</tr>
<tr>
<td>TOTAL</td>
<td>2822</td>
</tr>
</tbody>
</table>

3.2 Temporal expressions normaliser

I built a new normaliser on top of the one freely available from University of Rochester\(^8\): TRIOS. It is a rule-based normaliser and it has been proved to provide the second best performance in TempEval-2 Shared Task [17]. All the rules are in the form of regular expressions in a switch architecture: the activation of one of them excludes the activation of all the others.

I introduced a top layer with three new kinds of rules: extension, manipulation and post-manipulation rules.

The extension rules are just new rules that cover non-expected cases and are checked immediately before the pre-existing rules. If a temporal expression do not activate any of the extension rule, it goes into TRIOS. For example, some of these rules are used to normalise expressions of festivities dates such as “Thanksgiving day” or “Saint Patrick’s day”.

The manipulation rules have been introduced to turn particular well-known expressions into an easier form before TRIOS processes them. Once one of these rules is activated, the original temporal expression is transformed into a reduced one that is easier to normalise properly for the pre-existing set of rules. After the transformation, the new temporal expression is taken in input by TRIOS for the normalisation task.

Lastly, I used the post-manipulation rules to solve some deficiencies in the normaliser by adding further information lost by TRIOS and finally improving the performance. In this case the temporal expression is evaluated through the extension rules or the original set. At the end of the normalisation process the result is enriched with further information. For example, I used these rules to add information about seasons which are not considered in TRIOS at all.

In the end, I introduced 32 new regular expression patterns: 16 extension rules, 12 manipulation rules and 4 post-manipulation rules. The entire system is freely available online\(^9\) under GNU licence\(^10\).

4 Evaluation

I evaluated the normalisation system using the new corpus previously described as a training set and then I measured the performances with respect to the TempEval-2 Shared Task test set. This offered me the possibility of comparing my normaliser with all the others evaluated in that challenge.

In order to measure the difference between TRIOS and my extension I also tested both of them by using the new corpus. It is important to notice that TRIOS has been trained on the same data provided in the new corpus. For this reason a comparison between these systems is legitimate.

In both cases, the evaluation procedure is based on counting. Because the normalisation task is aimed at providing the right TYPE attribute and the right VALUE attribute, the evaluation is carried out by counting how many times the system provides the same value with respect to the human ones. It is important to emphasise that every value provided by the system that differs form the human one for at least one character is considered error.

If this method is quite reasonable for TYPE attribute, it might be too restrictive for VALUE attribute. Some practical examples could be of help to explain the problem.

- The human annotation of a certain timex is \{type: "DATE", value: "FUTURE_REF"\} whereas the system provides a a the more specific annotation \{type: "DATE", value: "2013-09-XX"\}.

- The system provides an annotation that is less specific than that provided by humans. For example, it happens when the human-annotation is \{type: "DATE", value: "2011-04-18"\} and the system provide \{type: "DATE", value: "2011-04-XX"\}.

In all these cases the annotations are considered completely wrong. Even when the system provides a partially wrong annotation, e.g. \{type: "DATE", value: "2011-04-23"\} for a human annotation of \{type: "DATE", value: "2011-04-18"\}, considering it a complete wrong result may be too strict because year and month are correct however. This fact has justified the investigation of other measurement metrics [18].

4.1 Results

The normalisation results with respect to TempEval-2 Shared Task are shown in Table 3. The new TRIOS

\(^8\)http://www.cs.rochester.edu/u/naushad/temporal
\(^9\)http://www.cs.man.ac.uk/~filannim/timex_normaliser.zip
\(^10\)http://www.gnu.org/licenses/gpl.html
### Table 3: Results obtained from TempEval-2 test set.

<table>
<thead>
<tr>
<th></th>
<th>type</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Edinburgh</td>
<td>0.84</td>
<td>0.63</td>
</tr>
<tr>
<td>HeidelTime</td>
<td>0.96</td>
<td>0.85</td>
</tr>
<tr>
<td>KUL</td>
<td>0.91</td>
<td>0.55</td>
</tr>
<tr>
<td>TERSEO</td>
<td>0.98</td>
<td>0.65</td>
</tr>
<tr>
<td>TipSem</td>
<td>0.92</td>
<td>0.65</td>
</tr>
<tr>
<td>TRIOS</td>
<td>0.94</td>
<td>0.76</td>
</tr>
<tr>
<td>TRIOS extension</td>
<td>0.95</td>
<td>0.86</td>
</tr>
</tbody>
</table>

### Table 4: Results obtained from the corpus.

<table>
<thead>
<tr>
<th></th>
<th>type</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRIOS</td>
<td>0.8572</td>
<td>0.6257</td>
</tr>
<tr>
<td>TRIOS extension</td>
<td>0.8853</td>
<td>0.7170</td>
</tr>
</tbody>
</table>

extension outperforms each system in the normalisation of **VALUE** attributes and performs competitively in the normalisation of **TYPE** attributes.

The table already shows that the normalisation of **VALUE** attributes is slightly harder than that of **TYPE** attributes. The extension of TRIOS outperformed the original system of 2.81% for **TYPE** attribute and 9.13% for **VALUE** attribute.

I randomly sub-sampled (400 temporal expressions) the original corpus 10 times and I measured the performances with TRIOS and my extension. I conducted a statistical analysis on the results and I proved that the difference is statistically significant (Willcoxon test), respectively $\rho = 0.00586$ and $\rho = 0.0001621$.

The normalisation results with respect to the new corpus are shown in Table 4.

#### 4.2 Error analysis

The original TRIOS normaliser made 1023 **VALUE** mistakes and 402 **TYPE** mistakes while its extension respectively made 779 and 323. Through an accurate analysis of the errors, I found plenty of human annotations that seemed to be wrong at first impression. Once I analysed the same annotations taking into account the entire sentence from which each expression had been extracted, I found that the human annotations were actually right. Some examples are shown in Table 5.

This leads to the conclusion that further improvements are possible only if I consider also the resolution of anaphoric expressions. To do this, it will be necessary to consider a wider window for each temporal expression that takes into account at least the entire sentence in which each temporal expression is located.

#### 5 Conclusions

I introduced a new rule-based normaliser of temporal expressions and I showed that it resulted in better performances than the current state-of-the-art system with respect to TempEval-2 Shared Task. I also illustrated the corpus of temporal expressions for normalisation and its purpose. I made both, the normaliser and the corpus, freely available online (GNU public licence apply).

#### 5.1 Future work

The work presented in this report is the product of a preliminary study in the field of information extraction. The results presented in this report clearly show the necessity of coping with anaphoric temporal expression to substantially enhance the performances of normalisation phase. Currently, the normalisation task takes into account only the temporal expressions, without considering a wider window, such as the entire sentence or a pre-defined number of words after and before the expression. This is required in order to cope with anaphoric expressions.

My long-term goal is to develop novel temporal expressions extraction techniques and use them in clinical domain. Because of the lack of pre-annotated clinical data, I will explore the use of semi-supervised machine learning approaches for the identification phase.

#### 5.2 Acknowledgements

I would like to thank Naushad UzZaman from the University of Rochester to have shared his normaliser with the scientific community. I would also like to acknowledge the support of UK Engineering and Physical Science Research Council in the form of doctoral training grant.

### References

Table 5: Some errors made by the normaliser.

<table>
<thead>
<tr>
<th>human</th>
<th>system</th>
</tr>
</thead>
<tbody>
<tr>
<td>25</td>
<td>1999-04-25</td>
</tr>
<tr>
<td>last year</td>
<td>1988-Q2</td>
</tr>
<tr>
<td>three years before</td>
<td>FUTURE_REF</td>
</tr>
<tr>
<td>the summer of 1862</td>
<td>FUTURE_REF</td>
</tr>
<tr>
<td>the weekend</td>
<td>P2D</td>
</tr>
</tbody>
</table>


