Improved Recognition and Normalisation of Polish Temporal Expressions

Jan Kocoń
Wrocław University
of Science and Technology
Wrocław, Poland
jan.kocon@pwr.edu.pl

Michał Marcińczuk
Wrocław University
of Science and Technology
Wrocław, Poland
michal.marcinczuk@pwr.edu.pl

Abstract

In this article we present the result of the recent research in the recognition and normalisation of Polish temporal expressions. The temporal information extracted from the text plays major role in many information extraction systems, like question answering, event recognition or discourse analysis. We proposed a new method for the temporal expressions normalisation, called Cascade of Partial Rules. Here we describe results achieved by updated version of Liner2 machine learning system.

1 Introduction

Temporal expressions tell us *when* something happens, *how long* something lasts or *how often* something occurs. Recognition of temporal expressions is a sequence labelling task and *normalisation* of temporal expressions is a process of their interpretation. It is often used in many natural language processing tasks, *e.g.,* question answering (Pustejovsky et al., 2005b), text summarisation (Daniel et al., 2003) or event recognition (Andersen et al., 1992; Llorens et al., 2010).

One of the most important thing is the ability to share the temporal information across many languages and systems. A serious problem would be to combine the information from different information extraction systems, *e.g.,* to analyse data from multi-lingual newspapers, where the expected result is the unification of metadata.

A widely used markup language for temporal and event expressions is TimeML (Saurí et al., 2006). At the beginning it was prepared for English, in the context of TERQAS[1] workshop, as a part of the ARDA-funded program AQUAINT[2] in order to improve the performance of question answering systems (Pustejovsky et al., 2005a). One of the most widely used rule-based system *HeidelTime*[3] (Strötgen and Gertz, 2013; Strötgen et al., 2013) which uses the TIMEX3 annotation standard, currently supports 13 languages: English, German, Dutch, Vietnamese, Arabic, Spanish, Italian, French, Chinese, Russian, Croatian, Estonian and Portuguese.

PLIMEX (Kocoń and Marcińczuk, 2015) is the adaptation of TIMEX3 specification with annotation guidelines presenting how to describe temporal expressions in Polish text documents. It is based on TIDES Instruction Manual for the Annotation of Temporal Expressions (Ferro, 2001), which describes TIMEX2 annotation format. The TIDES manual is also the core of the TIMEX3 annotation format, used in the TimeML specification (Saurí et al., 2006). Both documents present how to use the special Standard Generalized Markup Language tags to annotate temporal expressions, by inserting them directly into the text. We adapted types of temporal expressions from TIMEX3: DATE, TIME, DURATION and SET.

In this work we would like to propose a new method for the temporal expressions normalisation, called Cascade of Partial Rules, which significantly improved the quality of the normalisation system.

2 Related Works

Previous works in this area concerned preparation of a broad description of Polish temporal expressions with annotation guidelines, based in the state-of-the-art solution for English, mainly TimeML specification. The final product, called

[1] Time and Event Recognition for Question Answering Systems. An Advanced Research and Development Activity Workshop on Advanced Question Answering Technology

[2] http://www.informedia.cs.cmu.edu/aquaint/index.html

[3] https://code.google.com/p/heideltime/

Proceedings of Recent Advances in Natural Language Processing, pages 387–393,
Varna, Bulgaria, Sep 4–6 2017.
https://doi.org/10.26615/978-954-452-049-6_051

PLIMEX (Kocoń et al., 2015), was extended with the solution to capture the local semantics of temporal expressions, by adapting LTIMEX (Mazur, 2012) specification to Polish. Temporal description also supports further event identification and extends event description model, focusing at anchoring events in time, ordering events and reasoning about the persistence of events. PLIMEX was designed to address these issues. All documents in Polish Corpus of Wroclaw University of Technology (KPWr) were annotated using PLIMEX annotation guidelines with adapted types of temporal expressions (timexes) from TIMEX3: *Date, Time, Duration* and *Set*. In the following examples of timexes, the extent of the annotation in text (if needed) is marked with square brackets. All English translations of Polish examples are given in parentheses. These examples are presented in (Kocoń and Marcińczuk, 2017), and a broad description of each timex type is presented in (Kocoń et al., 2015).

2.1 Types of Temporal Expressions

Date is a type of *timex* which denotes a point on a timeline, i.e., a unit of time greater than or equal to the day. The key question is *when*:

(1) *[poniedziałek, 16 marca 1985 roku] (on [Monday, 16th March 1985])*

Time describes *timexes* which refer to the time of day. The key question is also *when*. For example *Smith wrócił (Smith returned)*:

(2) *[dwadzieścia po dwunastej] (at [twenty past twelve])*

Duration, in contrast to *Date*, has two points on a timeline associated with it – a start and an end point. A different name used in literature is *period* (Saquete et al., 2003). The key question is *how long*. For example *Smith był tutaj (Smith stayed there)*:

(3) *[dwa miesiące] (for [two months])*

Set is relating to more than one instance of a time unit – either a point or a period. The key question is *how often*. Examples – *Jan wraca pijany (John comes back drunk)*:

(4) *co dwa dni (every two days)*

2.2 Global Semantics

In order to describe *timexes* we adopted normalisation format from TimeML. The attribute VAL (Value) attached to the temporal expression is important in the normalisation context. It is a textual representation of a *timex*, which is assigned using guidelines described in the ISO-8601 standard. In order to describe it, the first letters of the English names that specify time are used (e.g. **Y**ear, **M**onth, **D**ay, **h**our, **m**inute, **BC** – before Christ, **AD** – Anno Domini, **T**ime, **MO**rning, **H**alf, **Q**uarter, **SU**mmer, **P**eriod). Each *timex* can be determined with respect to the proper type, using the coding proposed in the ISO-8601 standard, e.g. calendar date (YYYY-MM-DD), week of the year (YYYY-Wxx), hour (hh:mm:ss), date & hour (YYYY-MM-DDThh:mm:ss), duration (PxW).

Table 1 shows example values of the VAL attribute and the semantic meaning (Kocoń and Marcińczuk, 2017).

VAL	Meaning (EN)	Meaning (PL)
1992	year 1992	rok 1992
1992-SU	summer of 1992	lato 1992 roku
BC0346	year 346 BC	346 rok p.n.e.
P2Y	2 years	2 lata
P3W	3 weeks	3 tygodnie
PT8H2M	8 hours and 2 min.	8 godz. i 2 min.
P2DE	2 decades	dwie dekady

Table 1: Examples of VAL attributes and the semantic meaning of *timexes*. VAL values can be assigned manually during annotation or by automatic system.

Normalisation in the TimeML standard involves the determination of the global semantics for a *timex*. There is no indirect form of notation of the local semantics. The introduction of the intermediate stage of determining the global semantics is reasonable from the normalisation point of view. It can be seen in systems that recognise *timexes* in the English language (e.g. HeidelTime [4], (Strötgen and Gertz, 2013)), which often use their own intermediate standard of normalisation. For that purpose LTIMEX standard was adapted (Kocoń and Marcińczuk, 2017). It can be used to determine the local semantics of *timexes*.

[4] https://code.google.com/p/heideltime/

2.3 Local Semantics

LTIMEX standard was designed to be compatible with existing annotation schemes, especially TIMEX2 (and TIMEX3). It is beneficial for both design and evaluation to recognise the semantics of the expression with no context involved, what is called *local semantics*, representing the partial and underspecified context-free meaning of temporal expressions (Mazur, 2012). The compatibility with the existing schemes has two purposes:

- It is human-readable and requires minimum effort to use for annotators familiar with TIMEX3.
- It provides a relatively easy means of conversion from local semantics to global semantics.

If the temporal expression is explicit, there is no difference between LVAL and VAL representation. The specification is fully described in work (Kocoń and Marcińczuk, 2017). Table 2 shows the example values of the LVAL attribute and the semantic meaning.

LVAL	Meaning (EN)	Meaning (PL)
xxxx-01-03	January 3rd	3 stycznia
xxxx-xx-19	nineteenth	dziewiętnasty
xxxx-SU	summer	lato
-0000-00-01	yesterday	wczoraj
-0000-02	two months ago	dwa miesiące temu
<M06	last June	ostatni czerwiec
2D4	second Thursday	drugi wtorek
$1M03	last February	ostatni luty

Table 2: Examples of LVAL attributes and the semantic meaning of *timexes*. LVAL values can be assigned manually during annotation or by automatic system.

3 Normalisation Improvement

The previous rule-based normalisation system is presented in work (Kocoń and Marcińczuk, 2017) and contains 224 rules for local normalisation, for 3 classes of temporal expressions: *Date, Time, Duration*. Global normalisation is made as a Java code and contains 16 rules for two classes: *date, time* (for *duration* local and global interpretation is the same). Rules were both created and tuned using *train* data set.

We evaluated rules on *test* set. The evaluation was fully described in the article (UzZaman et al., 2013). We performed the evaluation on a full pipeline for the best model to recognise boundaries of entities and their types (trained on *train+tune* data set). We focus on the end-to-end comparison with other systems presented in the article (UzZaman et al., 2013).

3.1 Metrics

To evaluate temporal expressions we need to check *how many entites* are correctly identified, if *the extents for the entities* are correctly identified and *how many attributes* are correctly identified. We use classical precision, recall and F-measure for the recognition (UzZaman et al., 2013).

3.2 Previous Approach

We prepared module for Liner2 tool (Kocoń and Marcińczuk, 2017) to perform the normalisation process in order to get the local and global meaning of temporal expressions. We created the normalisation process as a two-step rule-based approach. The first step is to get the local meaning of the temporal expression, and the second (created on the basis of the local interpretation) is to get the global meaning. Rules were created by domain experts. For each class of temporal expression we defined several files with rules, patterns and normalisation dictionaries. The whole number of manually prepared resources is presented in Table 3.

Table 3: Previous normalisation resources for determining local value (LVAL) of temporal expressions, prepared using *train* data set. Table presents number of **Rules**, **Patt**erns and **Norm**alisations for each timex **Type**.

Type	Rules	Patt.	Norm.
date	110	75	21
time	93	52	13
duration	21	20	11
SUM	224	147	45

This approach is very similar to method presented in HeidelTime, except the extraction part, which in Liner2 is performed using Conditional Random Fields. In case of HeidelTime, it is necessary to manually specify the extraction rules and corresponding normalisation rules. Because in our solution we have already recognised chunks, we proposed a rule-based method for partial normalisation of timex chunk constituents.

3.3 Cascade of Partial Rules

We observed that in previous approach the further increase of quality (above 75% of LVAL F1) became very expensive, because only very specific timex examples not covered by rules left and there was no option to cover more than 2-3 of these examples with a single rule. Furthermore we saw that many rules for *time* are simple extensions of rules for *date* timexes. Because in Liner2 there is machine learning solution to chunk and classify timexes, we decided to prepare partial rules, which can match specific parts of timexes, which, *e.g.,* always denote *year* part in each timex containing that part, no matter if its type is *date* or *time*.

Because *durations* strongly differ from timexes describing points in a timeline, we prepared 2 sets of rules, the first for *dates* and *times*, the second for *durations*. Each set contains 3 subsets: *keys*, *maps* and *rules*. Examples:

```
{"keys": {
  "digitM_written": ["zero", "jeden",
   "dwa", "trzy", "cztery", ...],
  //zero, one, two, three, four
 "timeM_written": [
  "wieczór", "wieczor", "po północ",
  //evening, evening, after midnight
  "o północ", "północ", "rano", ...],
  //midnight, midnight, morning
 "tomorrowM_written": ["jutro",
  //                    tomorrow
  "następny dzień", "nazajutrz",
  //next day,        morrow
  "dzień następny", "jutrzejszy"]},
  //day after        tomorrow
"maps": {
 "time_written": {
   "wieczór": "EV", "wieczor": "EV",
  //evening          evening
   "po północ": "NI", "rano": "MO",
  //after midnight     morning
   "północ": "24:00", ...]},
  //midnight
 "digit_written": {
   "zero": "0", "jeden": "1",
  //zero         one
   "dwa": "2", "trzy": "3",
   "cztery": "4", "pięć": "5",...}},
"rules": [
{"desc": "[1] jutro",
 "keys": ["tomorrowM_written"],
 "groups": [],
 "match": "($tomorrowM_written)",
 "map": {},
 "value": {"year": "+0000",
   "month": "00", "day": "01"}},
{"desc": "[2] rano",
  "keys": ["timeM_written"],
  "groups": ["hour"],
  "match": "(?<hour>$timeM_written)",
  "map": {"hour": "time_written"},
  "value": {"hour": "$hour",
            "separator": "$hour"}},
{"desc": "[3] godzina 16 . 00",
  //      hour
  "keys": [],
  "groups": ["hour","minute"],
  "match": "godzina (?<hour>%d%d?)"+
           "%. (?<minute>%d%d?)",
  "map": {}, "value": {
    "hour": "$hour",
    "minute": "$minute"}}]}
```

Keys are `key:value` pairs, where `value` is a list of words. *Maps* are also `key:value` pairs, but each value is a `key:value` pair of `word:word`. *Rules* is a list of items. Each item consists of the following elements:

- **desc**ription – name of the rule,
- **keys** – a list of keys used in the rule,
- **groups** – a list of group names used in *match* regular expression,
- **match** – regular expression for partial matching, may contain `groups` in format: `(?<group_name>group_regex)`.
  May also contain `$key_name` elements which are replaced by `keys[key_name]` joined with regexp alternative character, *e.g.,* `$digitM_written` would be `zero|jeden|dwa|trzy|....`
- **map** – a dictionary, where key is a group name used in `match`, and value is a `map` key,
- **value** – a dictionary, where key is a name of the final partial normalisation element and value is the final form of the timex part.

Consider the following example of *time*:

Pol.:	*jutro*	*rano*	*o godzinie*	*9:30*
Eng.:	tomorrow	morning	at	9:30

This example is covered by the whole cascade of the given example rules. Rule `[1]` is simple: if there is any of words from `keys["tomorrowM_written"]` (in our example it is *jutro*) then set the final partial normalisations: `year="+0000", month="00", "day"="01"`. The order of rules is important, as the final values set by previous rules may be overwritten by next rules.

Rule `[2]` covers *rano* part and the match is: `(?<hour>$timeM_written)`. All textual words from temporal expressions are lemmatised and *match* looks at the chain of lemmas. There is one named group in this match: `hour`. In this case `groups["hour"]="rano"`.

There is also `map` defined as: `"map": {"hour": "time_written"}`. It means that the group from text has to be transformed like: `groups["hour"]= =maps["time_written"]["rano"]` and after the transformation: `groups["hour"]= "MO"`. Final values to set are: `{"hour": "$hour", "separator": "$hour"}`, so it looks like: `hour="MO", separator="MO"`.

Rule `[3]` covers *godzinie 9:30* part and the match is: `godzina (?<hour>%d%d?) %. (?<minute>%d%d?)`. There are two named groups: `hour` and `minute`, but there is no `map` defined in this rule, so values captured by groups remain unchanged: `hour="9", minute="30"`. Note that rule `[2]` sets the final value of `hour="MO"`, but rule `[3]` overwrites this as `hour="9"`.

After the whole cascade of rules, the final values are: `year="+0000", month="00", "day"="01", hour="9", minute="30", separator="MO"`. These values are combined together to obtain the proper LVAL value, which is: `+0000-00-01T9:30`. Note, that `separator` is important in this case, because without the information about the exact time of a day *rano* (Eng. *morning*), it could be 9AM or 9PM and the final LVAL value would be: `+0000-00-00t9:30` (small letter `t` instead of `T`, saying that the *hour* part of the local normalisation value is still ambiguous).

There are some other elements, which are not presented in examples, *e.g.,* it is possible to specify the *limit* parameter to apply rule only to a defined subset of temporal types, for example this rule runs only on **Time** expressions:

```
{
  "desc": "3 .",
  "keys": [],
  "groups": ["minute"],
  "match": "^(?<minute>%d+)( %.| %’)$",
  "map": {},
  "limit": ["t3_time"],
  "value": {
    "year": "+0000", "month": "00",
    "day": "00", "hour": "+00",
    "minute": "$minute"
  }
}
```

The number of items prepared in this approach is presented in Table 4.

Table 4: Current normalisation resources for determining local value (LVAL) of temporal expressions, prepared using *train* data set. Table presents number of **Rules**, **Patt**erns and **Norm**alisations for each timex **Type**.

Type	**Rules**	**Keys**	**Maps**
date & time	122	24	13
duration	45	4	3
SUM	167	28	16

4 Experiments and Results

Results in Table 5 show, that our system achieves similar quality as the best systems in case of extraction of temporal expressions. It is not easy to compare the quality for different languages, but the result shows, that the complexity of that task for Polish is more close to English in case of normalisation. A strict F-measure of Liner2 tool for the extraction task is very high (88.76%) and outperforms all other systems. In case of global normalisation there is probably still a room for improvement, mainly from the perspective of validating the training data. Still guidelines are not precise enough in case of determining the global value and it requires further work to prepare the exact procedure of determining *VAL* value for temporal expressions. Obtained results for VAL F1 are very close to these achived by best system for English (HeidelTime) and the proposed method (Liner2-new) is almost 11 p.p. better than the previous one (Liner2-old).

5 Conclusions

The comparison of the recognition results (see: Section 3.1) for previous (Liner2-old) and current approach (Liner2-new) is presented in Table 5. We analysed the statistical significance of differences, using paired-differences Student's t-test with a significance level $\alpha = 0.05$ (Dietterich, 1998). The improvement of the recognition quality is statistically significant.

The proposed Cascade of Partial Rules method, applied for determining the LVAL attribute for temporal expressions, outperformed the previous solution by almost 11 p.p. with using only 167 rules instead of 224. We believe, that the further improvement is still possible at the level of de-

System	Extr.	Lang.	Rel.F1	Rel.P	Rel.R	Str.F1	VAL F1	LVAL F1
HeidelTime	RB	SP	90.10	96.00	84.90	85.30	87.50	
TIPSem	DD	SP	87.40	93.70	81.90	82.60	82.00	
HeidelTime	RB	EN	90.30	93.08	87.68	81.34	77.61	
NavyTime	RB	EN	90.32	89.36	91.30	79.57	70.97	
ManTime	DD	EN	89.66	95.12	84.78	74.33	68.97	
SUTime	RB	EN	90.32	89.36	91.30	79.57	67.38	
ATT	DD	EN	85.25	98.11	75.36	78.69	65.57	
TIPSem	DD	EN	84.90	97.20	75.36	81.63	65.31	
ClearTK	DD	EN	90.23	93.75	86.96	82.71	64.66	
JU-CSE	DD	EN	86.38	93.28	80.43	75.49	63.81	
KUL	H	EN	83.67	92.92	76.09	69.32	62.95	
FSS-TimEx	RB	SP	65.20	86.60	52.30	49.50	62.70	
FSS-TimEx	RB	EN	85.06	90.24	80.43	49.04	58.24	
Liner2-old	DD	PL	88.50	90.25	86.81	85.06	66.71	75.14
Liner2-new	DD	PL	92.83	94.56	91.15	88.76	77.23	89.23

Table 5: Evaluation of the end-to-end **System**s for recognition and normalisation of temporal expressions – a comparison of *old* and *new* version of *Liner2* tool with other systems presented during SemEval 2013 (only the best results regarding VAL F1 measure for each tool). **Extr**action methods of temporal entities in these systems are: **RB** – rule-based, **DD** – data-driven, **H** – hybrid. **Lang**uages: **SP**anish, **EN**glish, **PL** – Polish. We used **Str**ict and **Rel**axed variants of evaluation measures presented in Section 3.1, also with measures for **VAL** and **LVAL**.

termining the VAL attribute for temporal expressions.

Acknowledgments

Work financed as part of the investment in the CLARIN-PL research infrastructure funded by the Polish Ministry of Science and Higher Education.

References

Peggy M. Andersen, Philip J. Hayes, Alison K. Huettner, Linda M. Schmandt, Irene B. Nirenburg, and Steven P. Weinstein. 1992. Automatic extraction of facts from press releases to generate news stories. In *In: Processing of the Third Conference on Applied Natural Language Processing*. pages 170–177.

Naomi Daniel, Dragomir Radev, and Timothy Allison. 2003. Sub-event based multi-document summarization. In *Proceedings of the HLT-NAACL 03 on Text Summarization Workshop - Volume 5*. Association for Computational Linguistics, Stroudsburg, PA, USA, HLT-NAACL-DUC '03, pages 9–16. https://doi.org/10.3115/1119467.1119469.

Thomas G. Dietterich. 1998. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation* 10:1895–1923.

Lisa Ferro. 2001. Instruction manual for the annotation of temporal expressions.

Jan Kocoń and Michał Marcińczuk. 2015. Recognition of Polish temporal expressions. *Proceedings of the Recent Advances in Natural Language Processing* pages 282–290. Recent Advances in Natural Language Processing (RANLP 2015).

Jan Kocoń and Michał Marcińczuk. 2017. Supervised approach to recognise Polish temporal expressions and rule-based interpretation of timexes. *Natural Language Engineering* 23(3):385–418. https://doi.org/10.1017/S1351324916000255.

Jan Kocoń, Michał Marcińczuk, Marcin Oleksy, Tomasz Bernaś, and Michał Wolski. 2015. Temporal Expressions in Polish Corpus KPWr. *Cognitive Studies — Etudes Cognitives* 15.

Hector Llorens, Estela Saquete, and Borja Navarro-Colorado. 2010. TimeML events recognition and classification: Learning CRF models with semantic roles. In *Proceedings of the 23rd International Conference on Computational Linguistics*. Association for Computational Linguistics, Stroudsburg, PA, USA, COLING '10, pages 725–733.

Paweł Mazur. 2012. *Broad-Coverage Rule-Based Processing of Temporal Expressions*. Phd thesis, Wrocław University of Science and Technology.

James Pustejovsky, Bob Ingria, Roser Sauri, Jose Castano, Jessica Littman, Rob Gaizauskas, Andrea Setzer, Graham Katz, and Inderjeet Mani. 2005a. The specification language timeml. *The language of time: A reader* pages 545–557.

James Pustejovsky, Robert Knippen, Jessica Littman, and Roser Saurí. 2005b. Temporal and event information in natural language text. *Language Resources and Evaluation* 39(2-3):123–164. https://doi.org/10.1007/s10579-005-7882-7.

Estela Saquete, Rafael Muñoz, and Patricio Martínez-Barco. 2003. Terseo: Temporal expression resolution system applied to event ordering. In Václav Matoušek and Pavel Mautner, editors, *Text, Speech and Dialogue*, Springer Berlin Heidelberg, volume 2807 of *Lecture Notes in Computer Science*, pages 220–228. https://doi.org/10.1007/978-3-540-39398-6_31.

Roser Saurí, Jessica Littman, Robert Gaizauskas, Andrea Setzer, and James Pustejovsky. 2006. TimeML annotation guidelines, version 1.2.1.

Jannik Strötgen and Michael Gertz. 2013. Multilingual and cross-domain temporal tagging. *Language Resources and Evaluation* 47(2):269–298. https://doi.org/10.1007/s10579-012-9179-y.

Jannik Strötgen, Julian Zell, and Michael Gertz. 2013. Heideltime: Tuning english and developing spanish resources for tempeval-3. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*. Association for Computational Linguistics, Atlanta, Georgia, USA, pages 15–19. http://www.aclweb.org/anthology/S13-2003.

Naushad UzZaman, Hector Llorens, Leon Derczynski, Marc Verhagen, James Allen, and James Pustejovsky. 2013. Semeval-2013 Task 1: Tempeval-3: Evaluating time expressions, events, and temporal relations. *Atlanta, Georgia, USA* page 1.