Recognition of Polish Temporal Expressions

Jan Kocoń†
G4.19 Research Group
Wrocław University of Technology
jan.kocon@pwr.edu.pl

Michał Marcińczuk
G4.19 Research Group
Wrocław University of Technology
michal.marcinczuk@pwr.edu.pl

Abstract

In this article we present the result of the recent research in the recognition 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 prepared a broad description of Polish temporal expressions, called PLIMEX. It is based on the state-of-the-art solutions for English, mostly TimeML specification. This solution can be used for the extraction of events and their attributes, in order to anchor events in time and to reason about the persistence of events. We prepared the annotation guidelines and we annotated all documents in Polish Corpus of Wrocław University of Technology (KPWr) using our specification. Here we describe results achieved by Liner2 machine learning system, adapted to recognise Polish temporal expressions.

1 Introduction

Recognition of temporal expressions and events became an active area of the research and plays a significant role in many natural language engineering systems. It is one of the major tasks in information extraction, which aim is to extract specific elements from unstructured data. In this research we focus on tracking changes over time in text written in natural language. Further reasoning about changes requires the information about temporally grounded events.

Textual references to time tell us how long something lasts, when something happens or how often occurs. People are usually conscious of their location in time — in most cases we know what is the current year, month and date and we use this information to capture the meaning of expressions like "yesterday", "tomorrow", "five days ago", "16th of November". Even in texts written in formal language (like newspaper articles), the global meaning of the given temporal expressions can be deduced by the analysis of the whole context of the document (often with metadata, such as document creation time). We can treat the global meaning of a temporal expression as a point in a timeline (e.g. "5th of December 2005"), a range (not always anchored to a specific point in a timeline, e.g. "two weeks") or even as a set of points in a timeline (e.g. "each Tuesday"). To determine the exact date, human (or machine learning system) often must know the full temporal context. These examples do not cover the complexity of the temporal expressions understanding. Sometimes a temporal expression is not the reference to the real world, but describes a fictional event. Sometimes a part of the text describes past or future, but it is not explicitly stated, but in other part of the document there are some clues to find out what tense is given. Also determining the temporal function of an expression can be a serious problem, even for a human, e.g. "four weeks" can be used to describe duration (how long something lasts) or point in time (e.g. "in four weeks"). An automatic system should distinguish between different categories of temporal expressions to capture its local and global semantic meaning properly. The extraction of temporal expressions identifies when something occurred by the recognition and normalization of expressions which refer to time. Often it is part of other reasoning systems, like in automatic question answering (Pustejovsky et al., 2005b) or event recognition (Andersen et al., 1992; Llorens et al., 2010b).

Mazur (2012) compared many state-of-the-art approaches to describe temporal expressions and divided these expressions into two main categories: instants and intervals. These are atoms

Proceedings of Recent Advances in Natural Language Processing, pages 282–290,
Hissar, Bulgaria, Sep 7–9 2015.

of time, which can be used to represent and reason about time. In the literature we can find many terms to describe instants, e.g. *a time point, a point, a point in time, a moment*. Also interval sometimes is called *period* (Benthem, 1983). Benthem (1983) uses interval as something that is between boundaries. On the other hand Allen (1995) finds interval temporal expressions in Benthem's meaning denoted by the term *duration*. The main difference between instants and intervals is that instants have no duration (treated as a feature of a period).

One of the most widely used specification for English to describe temporal information in natural language corpora is TimeML (Saurí et al., 2006). It was developed in the context of a workshop TERQAS[1], as a part of the ARDA-funded program AQUAINT[2] in a multi-project effort to improve the performance of question answering systems over documents written in natural language (Pustejovsky et al., 2005a). The aim of this research was to improve the access to information in the text through content rather than keywords. The main problem was the recognition of events and their temporal anchoring.

PLIMEX is a temporal annotation language suitable 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.

TimeML was successfully adapted to many languages and 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 11 languages: English, German, Dutch, Vietnamese, Arabic, Spanish, Italian, French, Chinese, Russian, and Croatian. Our research gives the opportunity to create a cross-domain temporal tagger which supports Polish.

[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/

2 Types of Temporal Expression in PLIMEX

In this section we define the TimeML types of temporal expressions adapted to Polish. All English translations of Polish examples are given in parentheses. The extent of the annotation in text (if needed) is marked with square brackets.

2.1 DATE

DATE is a type of temporal expressions which denotes a point on a timeline, i.e. a unit of time greater than or equal to a day. The key question is *when*.

Examples of DATE:

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

(2) *to wydarzyło się [drugiego listopada]*
(it happened on [the second of November])

(3) *w [październiku 1963 roku]*
(in [October 1963])

(4) *to będzie we [wtorek osiemnastego]*
(it will be on [Tuesday, the eighteenth])

(5) *byłem nad jeziorem [latem tamtego roku]*
(I was at the lake in [the summer of that year])

2.2 TIME

It is a type of a point expression that describes temporal expressions which refer to the time of a day, even if it is not clearly defined. The key question is also *when*. For example *Smith wrócił (Smith returned)*:

(6) *[za dziesięć trzecia]*
(at [ten to three])

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

(8) *o [ósmej rano]*
(at [eight in the morning])

(9) *o [9.00 w piątek 1 października 1999 roku]*
(at [9 am on Friday, October 1st, 1999])

(10) *[wczoraj późno w nocy]*
([yesterday late at night])

(11) *[wczoraj w nocy]*
([last night])

2.3 DURATION

DURATION, in contrast to DATE, has two points on a timeline associated with it — a start and an end point. An another name for it used in the literature is period (Saquete et al., 2003). The key question is *how long*.

Sometimes the range expressions are also included to this group (Mizobuchi et al., 1998), but these expressions can be treated as separate points in time (Mani and Wilson, 2000). For example *Smith był tutaj (Smith stayed there)*:

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

(13) *[48 godzin] (for [48 hours])*

(14) *[trzy tygodnie] (for [three weeks])*

(15) *[całą ostatnią noc] ([all last night])*

(16) *[20 dni] w lipcu ([20 days] in July)*

(17) *przez [trzy godziny] w zeszły poniedziałek*
(for [three hours] last Monday)

If a specific piece of information, which relates to the calendar, occurs in the temporal expression, then DATE is the right type of annotation. This is true even if the context suggests that this type of temporal expression indicates the duration of an event, e.g. *[Cały 1985] przebywał na emigracji ([The entire 1985] he lived in exile)*.

2.4 SET

The SET expression is a type of temporal expressions which is related 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)*:

(18) *[dwa razy w tygodniu] ([twice a week])*

(19) *[co dwa dni] ([every two days])*

(20) *[każdej niedzieli] ([every Sunday])*

3 Inter-annotator Agreement

The inter-annotator agreement was measured on randomly selected 100 documents from the Corpus of Wrocław University of Technology called KPWr. We used the positive specific agreement (Hripcsak and Rothschild, 2005) as it was measured for T3Platinum corpus (UzZaman et al., 2012) and two domain experts to annotate the subset of 100 documents from KPWr. We calculate the value of positive specific agreement (PSA) for each category. The results are presented in Table 1.

Type	1 and 2	only 1	only 2	PSA [%]
date	182	12	22	91.46
time	28	13	8	72.73
duration	13	3	4	78.79
set	6	2	9	52.17
$\sum$	229	30	43	86.25

Table 1: The value of positive specific agreement (PSA) calculated on the subset of 100 documents from KPWr, annotated independently by two domain experts using PLIMEX 1.0 guidelines. *1 and 2* means all annotations in which annotators 1 and 2 agreed. *Only 1* is the number of annotations made only by annotator 1 and *only 2* – the number of annotations made only by annotator 2.

According to (UzZaman et al., 2012) the best quality of data was achieved for TempEval-3 platinum corpus (T3Platinum) and it was annotated and reviewed by the organizers. Every file was annotated independently by at least two expert annotators. The result of overall T3Platinum inter-annotator positive specific agreement (PSA) at the level of annotating of temporal expressions with types was 0.88. In our case for 100 randomly selected documents the PSA value achieved was 86.25 (annotating using PLIMEX 1.0 specification).

4 Recognition

Many state of the art systems which recognize time expressions use supervised sequence labelling methods, mostly Conditional Random Fields (CRFs) (Lafferty et al., 2001). Recent studies in comparison of temporal expressions recognition systems for English like TempEval-2 and TempEval-3 (UzZaman et al., 2013) show a shift in the state-of-the-art. While normalisation is done best by rule-engineered systems, recognition is done well by a variety of methods. The conclusion is that rule-engineering and machine learning are equally good at timex recognition (UzZaman et al., 2013). Two best machine learning systems (comparing results of recognition, not nor-

malization) reported by UzZaman et al. (2013) — ClearTK (Steven, 2013) and TIPSem (Llorens et al., 2010a) — utilize CRFs in recognition of temporal expressions.

Our approach is based on *Liner2* tool[4] (Marcińczuk et al., 2013), which uses CRF++ toolkit[5]. This tool was successfully used in other natural language engineering tasks, mainly in named entities recognition (NER) (Marcińczuk and Kocoń, 2013; Marcińczuk et al., 2013).

5 Features

In recognition, the values of features are obtained at the token level. As a baseline we used a default set of features available in the Liner2 tool which was used to train models for named entity recognition (Marcińczuk and Kocoń, 2013; Marcińczuk et al., 2013). The set includes the following types of features:

Morphosyntactic — lemma, grammatical class, case, number, gender, complete morphological tag;

Orthographic — word, word shape (pattern), prefix, suffix, starts with upper case, starts with lower case, starts with symbol, starts with digit, has upper case, has symbol, has digit;

Semantic — word synonym, hypernym;

Dictionary — person first name, person last name, country name, city name, road name, person prefix, country prefix, person noun, person suffix, road prefix, specific triggers (country, district, geographic name, organization name, person name, region, settlement).

We decided to implement special features, which better characterize timexes' constituents:

Orthographic

- is_number — is word a number;
- structure — each character composing a word is converted to: **x** (if character is a letter), **d** (if character is a digit), **-** (in other case);
- structure_packed — each sequence of the same characters in *structure* is converted to a single character, e.g. **ddd** → **d**;
- other features describing word shape: is number, all upper, all letters, all digits, all alphanumeric, no letters, no alphanumeric, regex, word length

Semantic — *tophyper*: this feature uses plWordNet (Piasecki et al., 2014; Maziarz et al., 2013) to find the possible root of the given word in a graph built from the hyponymy relations joining lexical units in plWordNet. This process is currently not preceded by word sense disambiguation (Kedzia et al., 2014).

Dictionary — *timex*: a lexicon prepared by a domain expert, which contains words referring to time, e.g. **godzina** (Eng. *hour*), **minuta** (Eng. *minute*), etc.

6 Evaluation

We performed evaluation of temporal expressions recognition as it was proposed by UzZaman et al. (2013). The evaluation process is based on Task A of TempEval 2013, described in UzZaman et al. (2013), which aim is to determine the extent of temporal expressions in text as defined by the TimeML TIMEX3 tag and determine the class of expression (date, time, duration or set). To evaluate if the extents of entities and the classes are correctly identified (*exact match* evaluation) we used *precision, recall* and F_1*-score*. We also performed a relaxed match if there is an overlap between the *system entity* and *gold entity*, e.g. "sunday" vs "sunday morning". A detailed instruction for the relaxed match test score can be found in (Chinchor, 1998). Metrics used for *relaxed match*: **COR** – number correct, **ACT** – number actual, **POS** – number possible. Metrics used for *strict match*: **TP** – true positive, **FP** – false positive, **FN** – false negative. Measures used for both *strict* and *relaxed match*: **P** – precision, **R** – recall, $\mathbf{F}_1$ – F_1-score.

KPWr corpus consists of 1635 documents. A *train* set is 50% of all documents (819) and both *test* and *tune* evaluation data sets are 25% of all documents (408 on each set).

[4] http://nlp.pwr.wroc.pl/en/tools-and-resources/liner2

[5] http://crfpp.sourceforge.net/

6.1 Baseline

The baseline models utilize a set of features used for named entity recognition for Polish (Marcińczuk and Kocoń, 2013; Marcińczuk et al., 2013).

Annotation	TP	FP	FN	P [%]	R [%]	F_1 [%]
timex	2272	338	677	87.05	77.04	81.74
date	1760	201	353	89.75	83.29	86.40
time	111	56	177	66.47	38.54	48.79
duration	280	75	200	78.87	58.33	67.07
set	17	2	51	89.47	25.00	39.08
TOTAL	2168	334	781	86.65	73.52	79.55

Table 2: *Exact match* evaluation of TIMEX3 recognition (10-fold cross-validation on train set) — baseline features.

Annotation	COR	ACT	POS	P [%]	R [%]	F_1 [%]
timex	4694	526	1199	89.92	79.65	84.48
date	3594	328	628	91.64	85.13	88.26
time	243	91	330	72.75	42.41	53.58
duration	583	127	376	82.11	60.79	69.86
set	34	4	102	89.47	25.00	39.08
TOTAL	4454	550	1436	89.01	75.62	81.77

Table 3: *Relaxed match* evaluation of TIMEX recognition (10-fold cross-validation on train set) — baseline features.

Table 2 shows the results of the *exact match* evaluation of Polish temporal expressions recognition, performed as 10-fold cross-validation on the train set (see Table **??**). Table 3 shows the same result using *relaxed match* evaluation. Each table contains the result of two models: *4-class* (boundaries recognition and classification of temporal expressions; available classes: DATE, TIME, DURATION and SET) and *1-class* (boundaries recognition only, all classes casted to a single class named *timex*). Each model utilizes the baseline set of features.

6.2 Baseline with New Features

We added new features (described in Section 5) to the baseline set. The evaluation procedure is the same as described in Section 6.1.

Table 4 shows the results of the *exact match* evaluation of models which utilize both baseline and new features. Table 5 shows the same result using *relaxed match* evaluation. Each table contains the result of two models: *4-class* (boundaries recognition and classification of temporal expressions; available classes: DATE, TIME, DURATION and SET) and *1-class* (boundaries recognition only, all classes cast to a single class called *timex*).

Annotation	TP	FP	FN	P [%]	R [%]	F_1 [%]
timex	2389	367	560	86.68	81.01	83.75
date	1830	231	283	88.79	86.61	87.69
time	114	62	174	64.77	39.58	49.14
duration	299	104	181	74.19	62.29	67.72
set	18	3	50	85.71	26.47	40.45
TOTAL	2261	400	688	84.97	76.67	80.61

Table 4: *Exact match* evaluation of TIMEX recognition (10-fold cross-validation on train set) – baseline + new features.

Annotation	COR	ACT	POS	P [%]	R [%]	F_1 [%]
timex	4944	568	949	89.70	83.90	86.70
date	3733	389	491	90.56	88.38	89.46
time	259	93	316	73.58	45.04	55.88
duration	625	181	334	77.54	65.17	70.82
set	36	6	100	85.71	26.47	40.45
TOTAL	4653	669	1241	87.43	78.94	82.97

Table 5: *Relaxed match* evaluation of TIMEX recognition (10-fold cross-validation on train set) – baseline + new features.

We can see that adding new features improved F_1 for each model and for each match evaluation. Detailed analysis of these results is presented in Section 7.

6.3 Feature Selection

Feature selection methods can be divided into three categories: wrapper, filter and embedded methods (Blum and Langley, 1997; Hou and Jiao, 2010; Kohavi and John, 1997). We managed to find most suitable method, which can be applied to the CRFs probabilistic framework in order to avoid overfitting and reduce the storage and computational problem without the significant loss of F_1-score.

In this work we used the wrapper approach, where the feature subset selection is performed using the induction algorithm as a black box. The same algorithm is used to estimate the accuracy of the classifier trained on a selected subset of features. Each selection step depends on the result of

the classifier evaluation. We utilized the method described by Zhu (2010), which contains the following steps:

1. Let $M = \varnothing$ be the initial set of features.
2. Let C be the candidate feature set as atomic features. These are usually predicates on simple combination of words and tags, e.g.(x = John, z = PERSON), (x = John, z = LOCATION), (x = John, z = ORGANIZATION), etc. We used a context window size of 5.
3. Build an individual CRF model with features $M \cup \{f\}$ for each candidate feature $f \in C$. Select the candidate feature f^* which improve the CRF model the most (e.g., by the result of model evaluation). Let $M = M \cup \{f^*\}$, and $C = C - \{f^*\}$.
4. Go to step 3 until enough features have been added to the CRF model or there is no F_1-score gain after the current iteration.

Table 6 shows the result of the feature selection for TIMEX recognition. The procedure was performed for both *1-class* and *4-class* model. The initial set of features was the baseline with new features. We used average *exact match* F_1-score of 10-fold cross-validation on train set to evaluate the result after each step of the selection.

Model	Iter.	Selected feature	F_1 [%]	Gain [pps]
1-class	1	prefix-3	71.33	71.333
	2	hypernym1	77.59	6.260
	3	pattern	80.35	2.756
	4	dict_timex_base	81.46	1.114
	5	top4hyper1	81.46	0.947
	6	case	82.77	0.363
	7	structP	83.00	0.226
	8	dict_trigger_int_district	83.09	0.094
	9	starts_with_upper_case	83.15	0.055
	10	prefix-1	83.17	0.018
	11	hypernym2	83.40	0.231
4-class	1	prefix-3	70.03	70.031
	2	hypernym1	75.39	5.361
	3	struct	78.40	3.014
	4	dict_timex_base	79.10	0.695
	5	top4hyper4	79.89	0.789

Table 6: Result of the feature selection for TIMEX recognition (2 models: boundaries recognition and 4-class model). Used measure: average *exact match* F_1-score of 10-fold cross-validation on train set. Initial set of features: baseline + new features.

We can see that most of the proposed new features were selected (*dict_timex_base, top4hyper1, structP, starts_with_upper_case* for *1-class* model and *struct, dict_timex_base, top4hyper4* for *4-class* model). None of the proposed features were selected in the first or the second iteration. The most discriminative feature for both models is orthographic *prefix-3* and the second is semantic *hypernym1*.

Table 7 and Table 8 show the results of match evaluation of models which utilize features after the selection (*B+new*).

Annotation	TP	FP	FN	P [%]	R [%]	F_1 [%]
timex	225	42	47	84.27	82.72	83.49
date	1801	240	312	88.24	85.23	86.71
time	108	60	180	64.29	37.50	47.37
duration	296	106	184	73.63	61.67	67.12
set	17	2	51	89.47	25.00	39.08
TOTAL	2222	408	727	84.49	75.35	79.66

Table 7: *Exact match* evaluation of TIMEX recognition (10-fold cross-validation on train set) – after feature selection (see Table 6).

Annotation	COR	ACT	POS	P [%]	R [%]	F_1 [%]
timex	465	69	78	87.08	85.64	86.35
date	3673	409	547	89.98	87.04	88.48
time	247	89	316	73.51	43.87	54.95
duration	622	182	337	77.36	64.86	70.56
set	34	4	102	89.47	25.00	39.08
TOTAL	4576	684	1302	87.00	77.85	82.17

Table 8: *Relaxed match* evaluation of TIMEX recognition (10-fold cross-validation on train set) – after the features selection (see Table 6).

Detailed analysis of these results is presented in Section 7.

6.4 Processing Time

Table 6.4 shows the processing time of TIMEX recognition for the given feature sets: baseline, baseline with added new features (B+new) and features selected after the feature selection process (the initial set was B+new).

We see that *1-class* model after selection is about 3.6 times faster in recognition processing time than *baseline* and about 5 times faster than *B+new*. *4-class* model after selection is about 4.2 times faster than *baseline* and about 5.2 times faster than *B+new*. The selection process signifi-

Model	Fold	Baseline [s]	B+new [s]	Selection [s]
1-class	1	127.41	168.81	39.19
	2	114.98	148.92	29.13
	3	115.53	163.08	31.44
	4	112.65	158.13	31.15
	5	111.61	168.60	31.24
	6	113.70	151.39	30.93
	7	106.88	162.24	30.05
	8	111.81	158.49	30.94
	9	114.17	152.16	30.39
	10	111.95	159.34	31.14
	$\sum$	1140.68	1591.15	315.59
4-class	1	296.99	376.95	67.13
	2	263.21	335.42	69.44
	3	291.32	330.95	62.04
	4	276.87	334.62	73.17
	5	291.37	358.58	66.23
	6	273.14	354.18	69.02
	7	296.39	359.58	67.36
	8	282.13	340.85	66.72
	9	297.54	352.18	66.87
	10	276.97	369.55	70.42
	$\sum$	2845.93	3512.86	678.39

Table 9: Comparison of TIMEX recognition processing time (in seconds) for different feature sets on train set (10-fold cross-validation).

cantly improved the overall speed of the recognition.

7 Conclusions

Table 10 shows the comparison of results (F_1-score) achieved on different sets. We performed 10-fold cross-validation on the train set. Then each model was trained using the train set and evaluated on the tune set, divided into 10 parts. All the given results are averaged. We analyzed the statistical significance of differences between the baseline and the other models. To check the statistical significance of F_1-score difference we used paired-differences Student's t-test based on 10-fold cross-validation with a significance level $\alpha = 0.05$ (Dietterich, 1998). The statistically significant improvement with respect to the baseline is marked in bold.

We made the following observations:

Set	Model	Match	Baseline [%]	B+new [%]	Selection [%]
train	1-class	exact	81.74	**83.75**	**83.29**
		relaxed	84.48	**86.70**	**86.30**
	4-class	exact	79.55	**80.61**	79.66
		relaxed	81.77	**82.97**	82.17
tune	1-class	exact	79.37	80.91	80.06
		relaxed	82.81	**84.87**	**84.16**
	4-class	exact	77.75	**79.49**	77.96
		relaxed	80.30	**82.19**	80.89

Table 10: Comparison of results (F_1-score) achieved on different sets (*train* – 10-fold cross-validation on *train* set; tune – model is trained on *train* set and evaluated on *tune* set). Variants with *1 class* are boundaries recognition only. The difference between baseline and results in bold are statistically significant.

- Adding special features (see Section 5) to the baseline (*B+new* column) significantly improved the result for each evaluation variant except *exact match* for boundaries recognition (1 class) performed on tune set (the improvement is not statistically significant in that case).
- Performing the feature selection (see Section 6.3) statistically improved the results for 3 evaluation variants, only in boundaries detection. In each case we can see small improvement according to the baseline, but most of them (all 4-class recognition variants) are not statistically significant.
- Selection of features reduced the quality of the recognition (comparing to B+new) but the difference is not statistically significant.
- Each proposed model evaluation result is not worse comparing to the baseline result, most of them (10 of 16) are significantly better.
- The selection process significantly improved the overall speed of the recognition.

Acknowledgements

Work financed by the Polish Ministry of Science and Higher Education, a program in support of scientific units involved in the development of a European research infrastructure for the humanities

and social sciences in the scope of the consortia CLARIN ERIC and ESS-ERIC, 2015-2016.

References

James Allen. 1995. *Natural Language Understanding (2Nd Ed.)*. Benjamin-Cummings Publishing Co., Inc., Redwood City, CA, USA.

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.

Johan van Benthem. 1983. *The logic of time : a model-theoretic investigation into the varieties of temporal ontology and temporal discourse*. Synthese library. D. Reidel, Dordrecht, London, Boston.

Avrim L. Blum and Pat Langley. 1997. Selection of relevant features and examples in machine learning. *Artificial Intelligence*, 97(1–2):245 – 271. Relevance.

Nancy Chinchor. 1998. MUC-7 test scores introduction (appendix b). In *Proceedings of the 7th Message Understanding Conference*.

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.

Cuiqin Hou and Licheng Jiao. 2010. Selecting features of linear-chain conditional random fields via greedy stage-wise algorithms. *Pattern Recognition Letters*, 31(2):151 – 162.

George Hripcsak and Adam S Rothschild. 2005. Agreement, the f-measure, and reliability in information retrieval. *Journal of the American Medical Informatics Association*, 12(3):296–298.

Paweł Kedzia, Maciej Piasecki, Jan Kocoń, and Agnieszka Indyka-Piasecka. 2014. Distributionally extended network-based word sense disambiguation in semantic clustering of polish texts. *IERI Procedia*, 10(0):38 – 44. International Conference on Future Information Engineering (FIE 2014).

Ron Kohavi and George H. John. 1997. Wrappers for feature subset selection. *Artificial Intelligence*, 97(1–2):273 – 324. Relevance.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 282–289, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Hector Llorens, Estela Saquete, and Borja Navarro. 2010a. Tipsem (english and spanish): Evaluating crfs and semantic roles in tempeval-2. In *Association for Computational Linguistics*, pages 284–291.

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

Inderjeet Mani and George Wilson. 2000. Robust temporal processing of news. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, ACL '00, pages 69–76, Stroudsburg, PA, USA. Association for Computational Linguistics.

Michał Marcińczuk and Jan Kocoń. 2013. Recognition of Named Entities Boundaries in Polish Texts. In *ACL Workshop Proceedings (BSNLP 2013)*.

Michał Marcińczuk, Jan Kocoń, and Maciej Janicki. 2013. Liner2 – a customizable framework for proper names recognition for Poli. In Robert Bembenik, Lukasz Skonieczny, Henryk Rybinski, Marzena Kryszkiewicz, and Marek Niezgodka, editors, *Intelligent Tools for Building a Scientific Information Platform*, pages 231–253.

Marek Maziarz, Maciej Piasecki, Ewa Rudnicka, and Stanisław Szpakowicz. 2013. Beyond the transfer-and-merge wordnet construction: plwordnet and a comparison with wordnet. In *Proc. RANLP*, pages 443–452.

Paweł Mazur. 2012. *Broad-Coverage Rule-Based Processing of Temporal Expressions*. Ph.D. thesis, Politechnika Wrocławska.

Shoji Mizobuchi, Toru Sumitomo, Masao Fuketa, and Jun-ichi Aoe. 1998. A method for understanding time expressions. In *Systems, Man, and Cybernetics, 1998. 1998 IEEE International Conference on*, volume 2, pages 1151–1155 vol.2, Oct.

Maciej Piasecki, Marek Maziarz, Stanisław Szpakowicz, and Ewa Rudnicka. 2014. Plwordnet as the cornerstone of a toolkit of lexico-semantic resources. In *Proc. 7th International Global Wordnet Conference*, pages 304–312.

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.

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*, volume 2807 of *Lecture Notes in Computer Science*, pages 220–228. Springer Berlin Heidelberg.

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

Bethard Steven. 2013. Cleartk-timeml: A minimalist approach to tempeval 2013. pages 10–14.

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)*, pages 15–19, Atlanta, Georgia, USA, June. Association for Computational Linguistics.

Jannik Strötgen and Michael Gertz. 2013. Multilingual and cross-domain temporal tagging. *Language Resources and Evaluation*, 47(2):269–298.

Naushad UzZaman, Hector Llorens, James F. Allen, Leon Derczynski, Marc Verhagen, and James Pustejovsky. 2012. Tempeval-3: Evaluating events, time expressions, and temporal relations. *CoRR*, abs/1206.5333.

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.

Xiaojin Zhu. 2010. Conditional random fields. CS769 Advanced Natural Language Processing.