FSS-TimEx for TempEval-3: Extracting Temporal Information from Text

Vanni Zavarella
Joint Research Centre
European Commission
21027 Ispra, Italy
vanni.zavarella
@jrc.ec.europa.eu

Hristo Tanev
Joint Research Centre
European Commission
21027 Ispra, Italy
hristo.tanev
@jrc.ec.europa.eu

Abstract

We describe FSS-TimEx, a module for the recognition and normalization of temporal expressions we submitted to Task A and B of the TempEval-3 challenge. FSS-TimEx was developed as part of a multilingual event extraction system, Nexus, which runs on top of the EMM news processing engine. It consists of finite-state rule cascades, using minimalistic text processing stages and simple heuristics to model the relations between events and temporal expressions. Although FSS-TimEx is already deployed within an IE application in the medical domain, we found it useful to customize its output to the TimeML standard in order to have an independent performance measure and guide further developments.

1 Introduction

The FSS-TimEx (Finite State-based Shallow Time Extractor) system participating in TempEval-3 is integrated in the event extraction engine Nexus (Tanev et al., 2008), developed at the EC's Joint Research Center for extracting event information from on-line news articles gathered by the Europe Media Monitor (EMM) news aggregation and analysis family of applications(Steinberger et al., 2009). Nexus is highly multilingual[1] and easily portable across domains through semi-automatic learning of lexical resources. In the domain of epidemiological surveillance, the event extraction task required a particularly deep temporal information analysis, in order to detect temporal relations among event reports and mitigate the classical event duplication problem. As an example, from a report like:

> The overall death toll has risen to 160 **since the beginning of the year**, after 2 patients in Gulu and 2 in Masindi died **on Tue 5 Dec 2000**.

a system might be prevented to wrongly sum up the two victim counts (160+4) only if it is made aware of the inclusion relation between the first time interval and the date, which in turn implies normalizing the two temporal expressions.

Currently, FSS-TimEx is deployed for French, English and Italian and extensions are foreseen for further languages. Given such requirements for multilinguality, we developed FSS-TimEx using a linguistically light-weight approach, applying shallow processing modules only. On the other hand, as we need to extract highly structured information out of the detected temporal expressions, to be used in the subsequent normalization phase, we mostly opted for a rule-based approach, using finite-state grammar cascades, rather than machine learning methods. Nonetheless, some of the required lexicons were semi-automatically learned.

In our participation in Tasks A and B of the TempEval-3, we experimented with adapting an existing timex recognition module for the English language, to Spanish.

We first describe our system in 2,3 and 4, then in 5 we show and shortly discuss the results for Task A and Task B, and conclude with some thoughts on prospective developments.

[1] Currently, it covers English, French, Italian, Spanish, Portuguese, Turkish, Russian, Arabic.

*Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 58–63, Atlanta, Georgia, June 14-15, 2013. ©2013 Association for Computational Linguistics

2 System Modules

The system makes use of cascades of finite-state grammar rules applied to the output of a set of shallow text processing modules.

Text Processing Modules. These include tokenization, sentence splitting, domain-specific dictionary look-up and morphological analysis, which are all part of the CORLEONE (Core Linguistic Entity Online Extraction) engine (Piskorski, 2008). Morphological analysis purely consists of matching text tokens over full-form entries of a dictionary from the MULTEXT project (Erjavec, 2004), which encodes rich morphological features in a cross-lingual standard. Consequently, no PoS-tagging or parsing is performed upstream of the extraction grammars.

Finite-State Grammar Engine. We use the ExPRESS finite-state grammar engine (Piskorski, 2007). Grammars in the ExPRESS formalism consist of cascades of pattern-action rules, whose left-hand side (LHS) are regular expressions over flat feature structures (FFS) and the right-hand side (RHS) consists of a list of FFS (see Figure 1 below for an example). Variable binding from LHS to RHS, as well as string processing and Boolean operators on the RHS, allow to impose relatively complex constraints in the form of Boolean-valued predicates.

Weakly-supervised Learning of Lexical Resources. In order to determine the Class feature for the event extraction task, we experimented with using a language-independent method for weakly-supervised lexical acquisition. The algorithm takes as input a small set of seed terms, an unannotated text corpus and a parameter for the number of bootstrapping iterations: it then learns a ranked list of further terms, which are likely to belong to the same class, based on distributional n-gram features and term clustering (Tanev et al., in press). Although manual post-filtering is required, output term accuracy is reasonably high, and very high for top ranked terms.

3 Event and Event Feature Detection (Task B)

Although Nexus is a high precision event extraction system, we have not deployed it to model the event detection task. The reason is that Nexus is customized to recognize a number of highly domain-specific event types (e.g. `Armed_Conflict`, `Earthquake`,`Terrorist_Attack`) and will necessarily perform low in recall given the general, domain-independent definition of events in Task B. Instead, we tentatively used a small set of language-dependent finite-state rules to model verb phrase structure. Rules take as input MULTEXT morphological tokens and detect verb phrases along with a number of VP features, including Tense, which is used by the temporal normalizer to ground event modifying temporal expressions (see 4.2).

Class attribute was encoded in the morphological dictionary by using the output of the machine learning method sketched above: for each TimeML Event Class (Pustejovsky et al., 2003), we provided seed verb forms for all of its sub-classes, performed multi-class learning, and used the main Class label to annotate the union of output forms in the lexicon, after some manual cleaning.

The `OCCURRENCE` class was used as the default Class value for event verb forms, and it was overridden whenever a more specific event Class value was present[2].

We do not cover event nominal forms, as after some tests event referring and non-event referring noun classes appeared too difficult to tell apart by machine learning methods. Consequently, we expect system recall in Task B to be heavily limited.

4 Temporal Expressions (Task A)

FSS-TimEx's temporal expression processing consists of two stages.

In the Recognition phase, temporal expressions are detected and segmented in text and a more abstract representation of them is filled for further processing. Local parsing of timexes is performed by a cascade of hand-coded, partially language-dependent finite-state grammar rules using the ExPRESS engine, resulting in an intermediate fea-

[2] Otherwise, we chose randomly among alternative values of Class-ambiguous event expressions.

```
rule :> ( (lex & [TYPE:"temp_signal", SURFACE:#signal, NORMALIZED:"INCLUDED"]
              | lex & [TYPE:"temp_signal", NORMALIZED:"DURING"])
  lex & [TYPE:"quantifier", NORMALIZED:#mod]?  determiner?
  lex & [TYPE:"temp_mod", OP:#op, REF_TYPE:#ref_type]
  ( (lex & [TYPE: "numeral", NORMALIZED:#amount1]
 lex & [TYPE: "numeral", NORMALIZED:#amount2]?)
      | token & [TYPE: "any_natural_number", SURFACE:#amount1]
 lex & [TYPE:"time_unit", NUM:"p", GRAN:#gran]):x
-> x: period & [DIR:#op,REF_TYPE:#ref_type,MOD:#mod,GRAN:#gran,QUANT:#amount,SIGNAL:#signal]
& #amount := ConcForSum(#amount1,#amount2).
```

Figure 1: Sample recognition rule

ture structure-like representation, which is subsequently used by a language-independent Normalization stage to compute exact values of the time expressions, according to the TimeML standard.

We judge that such a strict coupling of recognition and normalization is better achieved through feature extraction rules than by deploying two separate processes[3].

4.1 Recognizing Temporal Expressions

A cascade of around 90 rules is deployed for the English language. These comprise lower-level rules, in charge of modelling language constructions in the target language, and typization rules that check the attribute configuration of lower-level rule output and return a corresponding structure, typed according to an intermediate annotation type set, exporting all attribute values relevant for normalization.

As an example, the rule shown in Figure1 detects single-boundary period expressions (e.g. *in the previous four weeks* or *during the next five days*).

Notice that the rule output type is the non TimeML-compliant `period` (i.e. an anchored time duration). This is an intermediate annotation type which is subsequently converted into a TimeML type (`Duration`) during the Normalization phase.

The temporal lexicon referenced by the grammar contains around 300 entries for the English language, classified into as many as 24 types, each described by a small attribute list. Sample entries from the English lexicon are listed in Figure 2.

This lexicon structure (types and attributes) was applied as such to the Spanish language; lexicon population was manually done in one day of work, by first translating lexical triggers (e.g. day, month names, numerals) and then gathering more functional entries (temporal adverbs, modifiers, etc.) by running test rules on large corpora. It turned out that, by using a parallel lexicon structure, we could reduce the cross-lingual re-arrangement of extraction rules for the Spanish grammar, minimizing the work cost to only 2 days, excluding fine tuning.

```
monday | TYPE:day_name | NORMALIZED:Monday
weeks | TYPE:time_unit | GRAN:week | NUM:p
night | TYPE:day_period_name | NORMALIZED:NI
ago | TYPE:temp_adv | OP:- | REF_TYPE:speaker
last | TYPE:temp_mod | OP:- | REF_TYPE:speaker
since | TYPE:temp_signal | NORMALIZED:BEGIN
early | TYPE:mod | NORMALIZED:START
```

Figure 2: Sample lexicon entries

4.2 Normalization

Normalization is a fully language-independent process, working with calendar representations of temporal expressions[4] built out of the output feature structures from the Recognition phase. It comprises two sub-processes:

Anchor selection. First, anchor selection determines and maintains a reference time for relative timex resolution, starting by using the Article Creation Date and updating it along the resolution process according to a simple search heuristic: select the closest preceding resolved timex with a compatible level of granularity. We experimented with two alternative settings for this, one restricting the search to timexes within the same sentence, the other spanning over the whole article text: we noticed a systematic gain in normalization accuracy with the former setting and we used it for Task A.

[3] This architecture is very close to the one proposed by the ITA-Chronos system (Negri, 2007).

[4] The normalization is entirely implemented in Java code.

Timex-Event mapping. For certain timex classes[5] we need to resort to Tense information from event-referring verb phrases in order to disambiguate between future and past interpretation. For this purpose, a simple, syntax-free heuristic is implemented to compute a mapping from each time expression onto the event it modifies, which just uses a weighted token distance metric, promoting events preceding the timex over those following it.

Finally, calendar arithmetic is used to resolve and normalize the value of relative timexes.

5 Results[6]

5.1 Temporal Expression Extraction

For English, our system scored in the middle range over all participant systems on relaxed match F1 measure. Strict match figures are not indicative: indeed, temporal signals (like *on* in *on Friday*) were systematically included in the extracted extent, contrary to the TIMEX3 tag specification, because this is required by finite-state parsing of the IE system with which FSS-TimEx was integrated.

Compared to the best performing system (BestEN in Table1), our approach mainly suffered from relatively low recall. Although such a rate of false negatives can be expected from a rule-based approach, in our case it was mostly due to two main "bugs" in the normalization code: first, in the process of tuning system output types to TimeML, we erroneously discarded date expressions introduced by temporal signals, like in *from now*; secondly, we do not normalize single adverbial expressions (*currently*), although they are detected by grammar rules.

We outperformed in Precision the best F1 system. Many false positives were all coming from a single article, where the word season in *flu season* was systematically annotated as an event in the gold standard. This kind of context-based inference seems to be out of reach for our rule-based, local parsing approach.

The major flaw in porting the system to Spanish language was a 28% Recall drop. Main types of false negatives included fuzzy expressions (e.g. *hace tiempo*), and compositional expressions.

Performance in timex classification and normalization still falls behind top scoring systems. Finite-state techniques can only parse local constructions, greedily consuming as long text spans as possible: therefore we systematically miss clausal relations like in: ***The day*** *before Raymond Roth was* ***pulled*** where we wrongly parsed a fully specified, relative timex *The day before*. Similar cases resulted at the same time in incorrect `Type` assignment, like in ***Two years*** *after his brain-cancer* ***diagnosis*** where we wrongly detect a `Date` type expression (*Two years after*).

Inaccurate event `Tense` attribute extraction sometimes caused wrong timex `Value` normalization. One noticeable source of such an error is reported speech, which temporarily changes the discourse utterance time and that we do not attempt to model in our anchor selection procedure. Interestingly, we noticed that even in cases when both timex-event mapping, and event `Tense` were correct,`Value` normalization was not. For example, in: *Northern Ireland's World Cup qualifier with Russia has been postponed until 15:00 GMT Saturday*, one can see that a shallow approach like ours, with no access to lexico-semantic knowledge, cannot pick up the implicit future tense interpretation of the event verb.

5.2 Event and Event Attribute Extraction

Results for Spanish (Table 2) show that a small set of rules were sufficient to detect event verbal expressions with high precision. The task was much harder for English, where morphological derivation is less often marked and given that we were not performing any PoS disambiguation.

Our main aim for Task B exercise was evaluating the performance of semi-automatic methods for verb classification, and to see how much verb tense information could help normalizing time expressions. `Class` attribute performance is rather poor, even considering that 7% of false hits in English were due to a bug in the MULTEXT lexicon causing the frequent form *said* not to be annotated as `REPORTING` event. A high rate of overlapping occurs among verb classes, causing our attempt to "lexicalize" the `Class` attribute, rather than trying to compute it

[5] E.g. what we refer to as `relativeTime` or `relativeOffset`, like *on Thursday* and *this weekend*, respectively.

[6] Results were obtained in 1.89 and 1.97 seconds of computation time respectively for English and Spanish data, on an Intel Core i3 M380 2.53GHz processor.

	Recognition						*Normalization*			
	Relaxed			*Strict*			*Value*		*Type*	
System	F1	P	R	F1	P	R	F1	A	F1	A
EN	0.85	0.90	0.80	0.49	0.52	0.46	0.58	0.68	0.69	0.81
BestEN	0.90	0.89	0.91	0.79	0.78	0.80	0.78	0.86	0.80	0.88
ES	0.65	0.86	0.52	0.49	0.65	0.39	0.50	0.77	0.62	0.95
BestES	0.90	0.96	0.84	0.85	0.90	0.80	0.85	0.94	0.87	0.97

Table 1: Performance of Temporal Expression Extraction and Normalization.

	Recognition			*Class*		*Tense*	
System	F1	P	R	F1	A	F1	A
EN	0.65	0.63	0.67	0.43	0.66	0.39	0.60
BestEN	0.81	0.81	0.81	0.72	0.89	0.60	0.73
ES	0.58	0.90	0.42	0.26	0.45	0.49	0.84
BestES	0.89	0.92	0.86	0.85	0.96	0.87	0.98

Table 2: Performance of Event and Event Attribute Extraction.

from context features of verb instances, to be unfeasible. `Tense` attribute performance[7] was too low to draw any conclusion on its impact on the Normalization task. However, for Spanish its accuracy (A in Figure 2) was higher and yet this did not result in increased timex `Value` scores[8].

6 Conclusion

The main positive outcome of our participation in TempEval-3 was that we were able to build a system with acceptable performance on Task A for Spanish, after a relatively quick adaptation from an existing English system. Recall was the bottleneck of such an experiment, while precision figures did not drop significantly, and Normalization accuracy even increased for Spanish[9], suggesting that a developer may be able to iteratively add language-specific rules so as to reduce false negatives, without endangering overall system precision.

A major flaw of our finite-state, local parsing approach is in recognizing event-anchored time expressions. In order to address this, our timex recognition rules must be further tuned to the TimeML standard in order to fully isolate temporal signals, and event detection recall must be significantly increased so as to cover event nominalizations. The detection of event referring expressions according to the general, context-independent definition in TimeML is not our main research target, however we plan to use statistical classification methods to increase the performance on this task as this is a prerequisite to achieve a reliable evaluation of our event-timex mapping heuristic. Event *Tense* extraction should be increased with the same purpose.

Acknowledgments

Many thanks to Maud Ehrmann for several useful discussions on the ontology of temporal entities and the TimeML standard.

References

Tomaz Erjavec. 2004. MULTEXT - East Morphosyntactic Specifications. *URL:http://nl.ijs.si/ME/V3/msd/html/*.

Silja Huttunen, Roman Yangarber, and Ralph Grishman. 2002. Diversity of Scenarios in Information Extraction. *Proceedings of the Third International Conference On Language Resources And Evaluation*, Las Palmas.

Matteo Negri. 2007. Dealing with Italian Temporal Expressions: The ITA-Chronos System. *Proceedings of EVALITA 2007*, Workshop held in conjunction with AI*IA 2007.

[7] `Tense` figures are unofficial, as we did not manage to export this attribute value because of a bug in the submitted system. However, we were able to reproduce the evaluation on a fixed system.

[8] We do not have independent performance figures of the timex-event mapping, although this mechanism was invariable across the two languages.

[9] Due to low F1 for timex entity extraction.

Piskorski, Jakub. 2007. ExPRESS Extraction Pattern Recognition Engine and Specification Suite. In *In Proceedings of the International Workshop Finite-State Methods and Natural language Processing 2007 (FSMNLP2007)*, Postdam, Germany.

Piskorski, Jakub. 2008. CORLEONE Core Linguistic Entity Online Extraction. Technical Report, EN 23393, Joint Research Center of the European Commission, Ispra, Italy.

James Pustejovsky, Jos M. Castao, Robert Ingria, Roser Sauri, Robert J. Gaizauskas, Andrea Setzer, Graham Katz, and Dragomir R. Radev. 2003. TimeML: Robust Specification of Event and Temporal Expressions in Text. In Mark T. Maybury, editor *New Directions in Question Answering*, pages 2834. AAAI Press, 2003.

Pustejovsky, J., P. Hanks, R. Sauri, A. See, R. Gaizauskas, A. Setzer, D. Radev, B. Sundheim, D. Day, L. Ferro, et al. 2003. The TimeBank corpus. In *Corpus Linguistics*volume 2003, 40.

Steinberger Ralf, Bruno Pouliquen & Erik van der Goot. 2009. An introduction to the Europe Media Monitor Family of Applications. In Fredric Gey, Noriko Kando & Jussi Karlgren (eds.): *Information Access in a Multilingual World - Proceedings of the SIGIR 2009 Workshop (SIGIR-CLIR'2009)*, pp.1-8.

Tanev Hristo, Piskorski Jakub, Atkinson Martin. 2008. Real-Time News Event Extraction for Global Crisis Monitoring. In *Proceedings of NLDB 2008*,2008:207218.

Hristo Tanev and Vanni Zavarella. in press. Multilingual Learning and Population of Event Ontologies. A Case Study for Social Media. In Paul Buitelaar and Philipp Cimiano editors *Towards the Multilingual Semantic Web*, Springer.

Naushad UzZaman, Hector Llorens, James F. Allen, Leon Derczynski, Marc Verhagen, James Pustejovsky. 2012. TempEval-3: Evaluating Events, Time Expressions, and Temporal Relations. *arXiv:1206.5333v1*.

Verhagen, M., R. Sauri, T. Caselli, and J. Pustejovsky 2010. SemEval-2010 task 13: TempEval-2. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, 5762, Association for Computational Linguistics.