10.12871/clicit201428

FBK-HLT-time: a complete Italian Temporal Processing system for EVENTI-Evalita 2014

Paramita Mirza
FBK, Trento, Italy
University of Trento
paramita@fbk.eu

Anne-Lyse Minard
FBK, Trento, Italy
minard@fbk.eu

Abstract

English. In this paper we present an end-to-end system for temporal processing of Italian texts based on a machine learning approach, specifically supervised classification. The system participated in all subtasks of the EVENTI task at Evalita 2014 (identification of time expressions, events, and temporal relations), including the pilot task on historical texts.

Italiano. *In questo articolo presentiamo un sistema end-to-end per l'analisi temporale su testi in italiano basato su algoritmi di apprendimento automatico (classificazione supervisionata). Il sistema ha partecipato a tutti i sottotask di EVENTI a Evalita 2014 (individuazione di espressioni di tempo, eventi e relazioni temporali), incluso il task pilota relativo a testi storici.*

1 Introduction

Research on temporal processing has been gaining a lot of attention from the NLP community in the recent years. The goal is to automatically extract events and temporal information from texts in natural language. The most recent shared task, TempEval-3 (UzZaman et al., 2013), focused on these goals. However, even though TempEval-3 organizers also released annotated data in Spanish, English is still given the most attention.

EVENTI[1], one of the new tasks of Evalita 2014[2], is established to promote research in temporal processing for Italian texts. Currently, even though there exist some independent modules for temporal expression extraction (e.g. HeidelTime (Strötgen et al., 2014)) and event extraction (e.g. Caselli et al. (2011)), there is no complete system for temporal processing for Italian. The main EVENTI task is composed of 4 subtasks for time expression recognition and normalization, event detection and classification and temporal relation extraction from newspaper articles. A pilot task on temporal processing of historical texts was also proposed. Our system participated in both tasks.

In this paper, we summarize our attempts and approaches in building a complete extraction system for temporal expressions, events, and temporal relations, which participates in the EVENTI challenge.

2 End-to-end system

We developed an end-to-end system to participate in the EVENTI challenge. It combines three subsystems: *(i)* time expression (timex) recognizer and normalizer, *(ii)* event extraction and *(iii)* temporal relation identification and classification. The subsystems used have been first developed for English as part of the NewsReader project[3] and then adapted to Italian. In order to adapt and test them for Italian, we used the training data released by the task organizers and split them into development and test data (in 80%/20% proportion).

The timex normalizer includes an adaptation of TimeNorm developed by Bethard (2013) for English, based on synchronous context free grammars. The other subsystems are based on machine learning and use Support Vector Machines algorithm. All subtasks, except the timex normalization subtask, are treated as classification problems. The feature sets used for building the classification models share a common ground, including morphological, syntactical and contextual features. The best combination of features and pre- and post-processing steps have been selected on the basis of experiments performed on the development data. The

[1] https://sites.google.com/site/eventievalita2014/

[2] http://www.evalita.it/2014

[3] http://www.newsreader-project.eu/

models used in the final system runs for the challenge have been trained on the whole training data.

3 Data and Tools

3.1 Data

The training data, the EVENTI corpus, is a simplified annotated version of the Ita-TimeBank released by the task organizers for developing purpose, containing 274 documents and around 112,385 tokens.

3.2 Tools

- **TextPro**[4] (Pianta et al., 2008), a suite of NLP tools for processing English and Italian texts. Among the modules we use: lemmatizer, morphological analyzer, part-of-speech tagger, chunker, named entity tagger and dependency parser.
- **YamCha**[5], a text chunker which uses SVMs algorithm. YamCha supports the dynamic features that are decided dynamically during the classification. It also supports multi-class classification using either *one-vs-rest* or *one-vs-one* strategies.
- **Snowball Italian stemmer**[6], a library for getting the stem form of a word.

3.3 Resources

- **MultiWordNet**[7], a multilingual lexical database containing WordNet aligned with the Italian WordNet. We extracted a list of words and their domains (e.g. *ricerca* [research] is associated to the domain *factotum*).
- **derIvaTario lexicon**[8], an annotated lexicon of about 11,000 Italian derivatives.
- **Lists of temporal signals** extracted from the training corpus. Mirza and Tonelli (2014) shows that the system performance benefits from distinguishing event-related signals (e.g. *mentre [while]*) from timex-related signals (e.g. *tra [within]*), therefore we split the list of signals into two separate lists.

4 Timex Extraction System

4.1 Timex Extent and Type Identification

The task of recognizing the extent of a timex, as well as determining the timex type (i.e. DATE, TIME, DURATION and SET), can be taken as a text chunking task. Since the extent of timex can be expressed by multi-token expressions, we employ the IOB2 tagging[9] to annotate the data. In the end, the classifier has to classify a token into 9 classes: B-DATE, I-DATE, B-TIME, I-TIME, B-DURATION, I-DURATION, B-SET, I-SET and O (for other).

The classifier is built using YamCha. One-vs-rest strategy for multi-class classification is used. The following features are defined to characterize a token:

- Token's text, lemma, part-of-speech (PoS) tags, flat constituent (noun phrase or verbal phrase), and the entity's type if the token is part of a named entity;
- Whether a token matches regular expression patterns for unit (e.g. *secondo [second]*), part of a day, name of days, name of months, name of seasons, ordinal and cardinal numbers, year (e.g. *'80*, *2014*), time, duration (e.g. *1h3'*, *50"*), temporal adverbs, names (e.g. *natale [Christmas]*), set (e.g. *mensile [monthly]*), or temporal signal as defined in TimeML;
- All of the above features for the preceding two and following two tokens, except the token's text;
- The preceding two labels tagged by the classifier.

4.2 Timex Value Normalization

For timex normalization, we decided to extend TimeNorm[10] (Bethard, 2013) to cover Italian time expressions. For English, it is shown to be the best performing system for most evaluation corpora compared with other systems such as HeidelTime (Strötgen et al., 2013) and TIMEN (Llorens et al., 2012).

We translated and modified some of the existing English grammar into Italian. Apart from the grammar, we modified the TimeNorm code in order to support Italian language specificity: normalization of accented letters, unification of articles and articulated prepositions, and handling the token splitting for Italian numbers that are concatenated (e.g. *duemilaquattordici [two thousand fourteen]*).

TimeNorm parses time expressions, and given an anchor time returns all possible normalizations following TimeML specifications. The anchor time

[4] `http://textpro.fbk.eu/`

[5] `http://chasen.org/~taku/software/yamcha/`

[6] `http://snowball.tartarus.org/algorithms/italian/stemmer.html`

[7] `http://multiwordnet.fbk.eu`

[8] `http://derivatario.sns.it/`

[9] IOB2 tagging format is a common tagging format for text chunking. The B- prefix is used to tag the beginning of a chunk, and the I- prefix indicates the tags inside a chunk. The label O indicates that a token belongs to no chunk.

[10] `http://github.com/bethard/timenorm`

passed to TimeNorm is always assumed to be the document creation time.

We have added post-process rules in order to select one of the returned values. The system chooses the value format that is most consistent with the timex type. For example if the timex is of type DURATION, the system selects the value starting with `P` (for Period of time).

After evaluating TimeNorm on the training data, we have added some pre-processing and post-processing steps in order to improve the performance of the system. The pre-processing rules treat time expressions composed by only one or two digits, and append either a unit or a name of month, which is inferred from a nearby timex or from the document creation time (e.g. *Siamo partiti il* $\underline{7_{timex}}$ *[We left (on) the 7]* (`DCT=2014-09-23 tid="t0"`) → $\underline{\text{7 settembre}_{timex}}$ [September 7]). We noticed that the TimeNorm grammar does not support the normalization of the *semester* or *half-year* unit (e.g. *il primo semestre [the first semester]*). In order to cope with this issue, we have developed some post-processing rules. Despite that, some expressions cannot be normalized because they are too complex, e.g. *'ultimo trimestre dell'anno precedente' [last quarter of the previous year]*.

4.3 Empty Timex Identification

The EVENTI annotation guidelines specifies the creation of empty TIMEX3 tags whenever a temporal expression can be inferred from a text-consuming one. For example, for the expression "*un mese fa [one month ago]*" two TIMEX3 tags are annotated: *(i)* one of type DURATION that strictly corresponds to the duration of one month (`P1M`) and *(ii)* one of type DATE that is not text consuming, referring to the date of one month ago.

As these timex are not text consuming they cannot be discovered by the text chunking approach. We performed the recognition of the empty timex using some simple post-processing rules and the timex normalization module.

5 Event Extraction System

Event detection is taken as a text chunking task, in which tokens have to be classified in two classes: EVENT (i.e. the token is included in an event extent) or O (for other). Then events are classified into one of the 7 TimeML classes: OCCURRENCE, STATE, I_STATE, REPORTING, I_ACTION, PERCEPTION and ASPECTUAL.

In the case of multi-token events, we considered only the head of events in building the classification models. Once the events have been extracted and classified, we post-process the text to detect the full extent of multi-token events. The post-processing is done by using the list of multi-token expressions in Italian provided by the task organizers.

The classification models are built using Yamcha. The following features are taken into consideration both for event extent and class identification:

- Token's lemma, stem, PoS tags, flat constituent (noun phrase or verbal phrase), and the entity's type if the token is part of a named entity;
- Whether the token is part of a time expression (labels from the Timex Extraction system);
- Token's simplified PoS (e.g. `n` for nouns, `v` for verbs, etc.), tense for verbs;
- Token's suffix if it is one of the following: -zione, -mento, -tura and -aggio;
- The frequency of the token's appearance in an event extent within the training corpus. We have defined three values to represent the frequency: *never* (the token never appears in an event extent), *sometimes* (it appears more often outside of an event extent than inside), *often* (it appears more often in an event extent than outside);
- Token's WordNet domain;
- Token's derivative if applicable (e.g. *chiudere [close]* for *chiusura [closure]*);
- The preceding 3 labels tagged by the classifier.

The features related to token's suffix, derived word, WordNet domain and frequency are used mainly to improve the recognition of nominal events. The eventive meaning of a noun is indeed difficult to detect with only simple features.

We have submitted three runs that differ from the number of classifiers and the multi-class classification strategy used.

Run 1 / Run2 In both runs two classifiers are used: *(i)* one to identify event extents and *(ii)* one to classify the identified events. For Run 1, the method used for multi-class classification is the one-vs-one strategy, while the one-vs-rest strategy is used for Run 2. All the features described above are used. In addition, some features of the two preceding and the two following tokens are included (e.g. token's PoS, lemma). For event class classification, we have added in the feature set the label predicted by the first classifier (EVENT or O).

Run 3 One single classifier is trained to both detect and classify events. Each token is classified into one of the seven event classes or O for other (i.e. the token is not part of an event extent). The one-vs-rest multi-class classification method is used.

6 Temporal Relation Extraction System

6.1 Temporal Link Identification

In the EVENTI challenge, the task of temporal link identification is restricted to event/event and event/timex pairs within the same sentence. We consider all combinations of event/event and event/timex pairs within the same sentence (in a forward manner) as candidate temporal links. For example, if we have a sentence with an entity order such as "...ev_1...tmx_1...ev_2...", the candidate pairs are (ev_1, tmx_1), (ev_2, tmx_1) and (ev_1, ev_2).

Next, in order to filter the candidate links, we classify a given event/event or event/timex pair into two classes: REL (i.e. the pair is considered as having a temporal link) or O (for other).

A classification model is trained for each type of entity pair (event/event and event/timex), as suggested in previous works (Mani et al., 2006). Again, YamCha is used to build the classifiers. However, this time, a feature vector is built for each pair of entities (e_1, e_2) and not for each token as in the previous classification tasks. The same set of features used for the temporal relation classification task, which are explained in the following section, is applied.

6.2 Temporal Relation Type Classification

Given an ordered pair of entities (e_1, e_2) that could be either event/event or event/timex pair, the classifier has to assign a certain label, namely one of the 13 TimeML temporal relation types: BEFORE, AFTER, IBEFORE, IAFTER, INCLUDES, IS_INCLUDED, MEASURE, SIMULTANEOUS, BEGINS, BEGUN_BY, ENDS, ENDED_BY and IDENTITY.

The classification models are built in the same way as in identifying temporal links. The overall approach is largely inspired by an existing framework for the classification of temporal relations in English documents (Mirza and Tonelli, 2014). The implemented features are as follows:

String and grammatical features. Tokens, lemmas, PoS tags and flat constituent (noun phrase or verbal phrase) of e_1 and e_2, along with a binary feature indicating whether e_1 and e_2 have the same PoS tags (only for event/event pairs).

Textual context. Pair order (only for event/timex pairs, i.e. event/timex or timex/event), textual order (i.e. the appearance order of e_1 and e_2 in the text) and entity distance (i.e. the number of entities occurring between e_1 and e_2).

Entity attributes. Event attributes (*class*, *tense*, *aspect* and *polarity*) [11], and timex *type* attribute [12] of e_1 and e_2 as specified in TimeML annotation. Four binary features are used to represent whether e_1 and e_2 have the same event attributes or not (only for event/event pairs).

Dependency information. Dependency relation type existing between e_1 and e_2, dependency order (i.e. *governor-dependent* or *dependent-governor*), and binary features indicating whether e_1/e_2 is the *root* of the sentence.

Temporal signals. We take into account the list of temporal signals as explained in Section 3.3. Tokens of temporal signals occurring around e_1 and e_2 and and their positions with respect to e_1 and e_2 (i.e. *between* e_1 and e_2, *before* e_1, or at the beginning of the sentence) are used as features.

In order to provide the classifier with more data to learn from, we bootstrap the training data with inverse relations (e.g. BEFORE/AFTER). By switching the order of the entities in a given pair and labelling the pair with the inverse relation type, we roughly double the size of the training corpus.

There are two variations of system submitted.

Run 1 We only consider the frequent relation types, i.e. BEFORE, AFTER, INCLUDES, IS_INCLUDED, MEASURE, SIMULTANEOUS and IDENTITY, in building the classifier for event/event pairs. Using only the frequent relation types results in better performance than using the full set of relation types, because the dataset becomes more balanced.

Run 2 Similar as Run 1, however, we incorporate the TLINK rules for event/timex pairs which conforms to specific signal patterns as explained in the task guidelines[13]. For example, EVENT + dal + DATE$_{type}$ → relType=BEGUN_BY. The event/timex

[11] The event attributes *tense*, *aspect* and *polarity* have been annotated using rules based on the EVENTI guidelines and using the morphological analyses of each token.

[12] The *value* attribute tends to decrease the classifier performance as shown in Mirza and Tonelli (2014), and therefore, it is excluded from the feature set.

[13] http://sites.google.com/site/eventievalita2014/file-cabinet/specificheEvalita_v2.pdf

pairs matching the patterns are automatically assigned with relation types according to the rules, and do not need to be classified.

7 Results

Table 1 shows the results of our system on the two tasks of the EVENTI challenge, i.e. the main task (MT) and the pilot task (PT), and on the 4 subtasks (Task A, B, C and D). For the pilot task we give only the results obtained with the best system.

7.1 Timex Extraction - Task A

For the main task, in recognizing the extent of timex, the system achieves 0.827 F-score using strict-match scheme. The accuracy in determining the timex type is 0.8, while the accuracy in determining the timex value is 0.665.

For the pilot task, in recognizing the extent of timex, the system achieves comparable scores with the main task. However, in determining the timex type and value, the accuracies drop considerably.

7.2 Event Extraction - Task B

On task B the best results are achieved with Run 1, with a strict F-score of 0.867 for event detection and an F-score of 0.671 for event classification. In this run we trained two classifiers using the one-vs-one multi-class classification strategy. On the pilot task data the results are a little bit lower, with a strict F-score of 0.834 for event detection and an F-score of 0.604 for event classification.

Note that for Run 3 due to a problem while training the model on all the training data, we have re-trained the model on only 80% of the data.

7.3 Determining Temporal Relation Types - Task D

For the main task, note that there is a slight error in the format conversion for Run 2. Hence, we recomputed the scores of *Run 2** independently, which results in a slightly better performance compared with Run 1. The system (*Run 2**) yields 0.738 F-score using TempEval-3 evaluation scheme.

For the pilot task (post-submission evaluation), both Run 1 and Run 2 have exactly the same scores, which are 0.588 F-score using TempEval-3 evaluation scheme. This suggests that in the pilot data there is no event/timex pair matching the EVENT-signal-TIMEX3 pattern rules listed in the task guidelines.

7.4 Temporal Awareness - Task C

For this task, we combine the timex extraction system, the 3 system runs for event extraction (Ev), the system for identifying temporal links, and the 2 system runs for classifying temporal relation types (Tr). We found that for both main task and pilot task, the best performing system is the combination of the best run of task B (Ev Run 1) and the best run of task D (Tr Run 1), with 0.341 F-score and 0.232 F-score respectively (strict-match evaluation).

8 Discussion

We have developed an end-to-end system for temporal processing of Italian text. In the EVENTI challenge, we have tested our system on recent newspaper articles, taken from the same sources as the training data, as well as on newspaper articles published in 1914. Without any specific adaptation to historical text, our system yields comparable results.

For the timex extraction task, in identifying the extent and the type of timex, the system achieves good results. In normalizing the timex value, however, the performance is still considerably lower than the state-of-the-art system for English (TimeNorm). This suggests that the TimeNorm adaptation for Italian can still be improved.

For determining timex types and values (as well as temporal relation types), the system performs better on the main task than on the pilot task. With the assumption that the articles written with a gap of one century differ more at the lexical level than at the syntactic level, our take on this phenomena is that in determining timex types, timex values and temporal relation types, the system relies more on the lexical/semantic features. Hence, the performances of the system decrease when it is applied on historical texts.

In the event extraction task, we observed that the event classification performed better with the one-vs-one multi-class strategy than with the one-vs-rest one. Looking at the number of predicted events with both classifiers, the second classifier did not classify all the events found (1036 events were not classified). For this reason the precision is slightly better but the recall is much lower. We have also observed some problems in the detection of multi-token events.

For the relation classification task, as the dataset is heavily skewed, we have decided to reduce the set of temporal relation types. It would be inter-

Subtask	Task	Run	F1	R	P	Strict F1	Strict R	Strict P	type F1	value F1
Task A	**MT**	R1	**0.886**	0.841	0.936	**0.827**	0.785	0.873	0.800	0.665
	PT	R1	**0.870**	0.794	0.963	**0.746**	0.680	0.825	0.678	0.475
Task B	**MT**	R1	**0.884**	0.868	0.902	**0.867**	0.850	0.884	0.671	
		R2	0.749	0.632	0.917	0.732	0.618	0.897	0.632	
		R3	0.875	0.838	0.915	0.858	0.822	0.898	0.670	
	PT	R1	**0.843**	0.793	0.900	**0.834**	0.784	0.890	0.604	
Task D	**MT**	R1	0.736	0.731	0.740	0.731	0.727	0.735		
		R2	0.419	0.541	0.342	0.309	0.307	0.311		
		*R2**	***0.738***	*0.733*	*0.742*	***0.733***	*0.729*	*0.737*		
	PT	*R1 & R2*	***0.588***	*0.588*	*0.588*	***0.570***	*0.570*	*0.570*		
Task C	**MT**	Ev R1 / Tr R1	**0.264**	0.238	0.296	**0.341**	0.308	0.381		
		Ev R1 / Tr R2	0.253	0.241	0.265	0.325	0.313	0.339		
		Ev R2 / Tr R1	0.209	0.167	0.282	0.267	0.209	0.368		
		Ev R2 / Tr R2	0.203	0.168	0.255	0.258	0.212	0.329		
		Ev R3 / Tr R1	0.247	0.211	0.297	0.327	0.279	0.395		
		Ev R3 / Tr R2	0.247	0.211	0.297	0.327	0.279	0.395		
	PT	Ev R1 / Tr R1	**0.185**	0.139	0.277	**0.232**	0.173	0.349		

Table 1: FBK-HLT-time results (MT: Main Task; PT: Pilot Task; Ev Rn: run n of Task B; Tr Rn: run n of Task D)

esting to see if using patterns or trigger lists as a post-processing step can improve the system on the detection of the under-represented relations. For example, the relation type IAFTER (as a special case of the relation AFTER) can be recognized through the adjective *immediato [immediate]*.

In a close future, our system will be included in the TextPro tools suite, both for Italian and English.

Acknowledgments

The research leading to this paper was partially supported by the European Union's 7th Framework Programme via the NewsReader Project (ICT-316404).

References

Steven Bethard. 2013. A synchronous context free grammar for time normalization. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 821–826, Seattle, Washington, USA.

Tommaso Caselli, Hector Llorens, Borja Navarro-Colorado, and Estela Saquete. 2011. Data-driven approach using semantics for recognizing and classifying timeml events in italian. In *Proceedings of the International Conference Recent Advances in Natural Language Processing 2011*, pages 533–538, Hissar, Bulgaria.

Hector Llorens, Leon Derczynski, Robert Gaizauskas, and Estela Saquete. 2012. Timen: An open temporal expression normalisation resource. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey.

Inderjeet Mani, Marc Verhagen, Ben Wellner, Chong Min Lee, and James Pustejovsky. 2006. Machine learning of temporal relations. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*, ACL-44, pages 753–760, Stroudsburg, PA, USA.

Paramita Mirza and Sara Tonelli. 2014. Classifying Temporal Relations with Simple Features. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 308–317, Gothenburg, Sweden.

Emanuele Pianta, Christian Girardi, and Roberto Zanoli. 2008. The TextPro Tool Suite. In *Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC 2008)*, Marrakech, Morocco.

Jannik Strötgen, Julian Zell, and Michael Gertz. 2013. Heideltime: Tuning english and developing spanish resources for tempeval-3. In *Proceedings of the Seventh International Workshop on Semantic Evaluation*, SemEval '13, pages 15–19, Atlanta, Georgia, USA.

Jannik Strötgen, Ayser Armiti, Tran Van Canh, Julian Zell, and Michael Gertz. 2014. Time for more languages: Temporal tagging of arabic, italian, spanish, and vietnamese. *ACM Transactions on Asian Language Information Processing (TALIP)*, 13(1):1–21.

Naushad UzZaman, Hector Llorens, Leon Derczynski, James Allen, Marc Verhagen, and James Pustejovsky. 2013. Semeval-2013 task 1: Tempeval-3: Evaluating time expressions, events, and temporal relations. In *Proceedings of the Seventh International Workshop on Semantic Evaluation*, SemEval '13, pages 1–9, Atlanta, Georgia, USA.