Comparing Two Approaches for the Recognition of Temporal Expressions*

Oleksandr Kolomiyets and Marie-Francine Moens

Department of Computer Science, K.U. Leuven, Celestijnenlaan 200A,
Heverlee, B-3001, Belgium

{Oleksandr.Kolomiyets, Sien.Moens}@cs.kuleuven.be

Abstract. Temporal expressions are important structures in natural language. In order to understand text, temporal expressions have to be extracted and normalized. In this paper we present and compare two approaches for the automatic recognition of temporal expressions, based on a supervised machine learning approach and trained on TimeBank. The first approach performs a token-by-token classification and the second one does a binary constituent-based classification of chunk phrases. Our experiments demonstrate that on the TimeBank corpus constituent-based classification performs better than the token-based one. It achieves F1-measure values of 0.852 for the detection task and 0.828 when an exact match is required, which is better than the state-of-the-art results for temporal expression recognition on TimeBank.

1 Introduction

Temporal information extraction in free text has been a research focus since 1995, when temporal expressions, sometimes also referred to as time expressions or TIMEXes, were processed as single capitalized tokens within the scope of the Message Understanding Conference (MUC) and the Named Entity Recognition task. As the demand for deeper semantic analysis tools increased, rule-based systems were proposed to solve this problem. The rule-based approach is characterized by providing decent results with a high precision level, and yields rules, which can be easily interpreted by humans. With the advent of new, annotated linguistic corpora, supervised machine-learning approaches have become the enabling technique for many problems in natural language processing. In 2004 the Automated Content Extraction (ACE) launched a competition campaign for Temporal Expression Recognition Normalization, TERN. The tasks were set to identify temporal expressions in free text and normalize them providing ISO-based date-time values.

While the ACE TERN initiative along with the provided corpus was aimed at the recognition and normalization problems, more advanced temporal processing on the same dataset was not possible. The most recent annotation language for temporal expressions, TimeML [1], and the underlying annotated corpus TimeBank [2], opens up new horizons for automated temporal information extraction and reasoning.

* This work has been partly funded by the Flemish government (through IWT) and by Space Applications Services NV as part of the ITEA2 project LINDO (ITEA2-06011).

A large number of rule-based and machine learning approaches were proposed for identification of temporal expressions. Comparative studies became possible with standardized annotated corpora, such as the ACE TERN and TimeBank. While the ACE TERN corpus is very often used for performance reporting, it restricts the temporal analysis to identification and normalization. By contrast, TimeBank provides a basis for all-around temporal processing, but lacks experimental results. In this paper we describe and compare two supervised machine learning approaches for identifying temporal information in free text. Both are trained on TimeBank, but follow two different classification techniques: token-by-token following B-I-O encoding and constituent-based classifications.

The remainder of the paper is organized as follows. In Section 2 we provide the details of relevant work done in this field along with corpora and annotations schemes used. Section 3 describes the approaches. Experimental setup, results and error analysis are provided in Section 4. Finally, Section 5 gives an outlook for further improvements and research.

2 Background and Related Work

2.1 Evaluation Metrics

With the start of the ACE TERN competition in 2004, two major evaluation conditions were proposed: Recognition+Normalization (full task) and Recognition only [3]. For the recognition task, temporal expressions have to be found. The system performance is scored with respect to two major metrics: detection and exact match. For detection the scoring is very generous and implies a minimal overlap of one character between the reference extent and the system output. For an exact match, in turn, the correct extent boundaries are sought.

As the exact match evaluation appeared to be too strict for TimeBank, an alternative evaluation metric was used in [4] and called “sloppy span”. In this case the system scores as long as the detected right-side boundary is the same as in the corresponding TimeBank’s extent.

2.2 Datasets

To date, there are two annotated corpora used for performance evaluations of temporal taggers, the ACE TERN corpus and TimeBank [2]. Most of the implementations referred to as the state-of-the-art were developed in the scope of the ACE TERN 2004. For evaluations, a training corpus of 862 documents with about 306 thousand words was provided. Each document represents a news article formatted in XML, in which TIMEX2 XML tags denote temporal expressions. The total number of temporal expressions for training is 8047 TIMEX2 tags with an average of 10.5 per document. The test set comprises 192 documents with 1828 TIMEX2 tags [5].

The annotation of temporal expressions in the ACE corpus was done with respect to the TIDES annotation guidelines [6]. The TIDES standard specifies so-called markable expressions, whose syntactic head must be an appropriate lexical trigger, e.g. “*minute*”, “*afternoon*”, “*Monday*”, “*8:00*”, “*future*” etc. When tagged, the full extent of the tag must correspond to one of the grammatical categories: nouns, noun

phrases, adjectives, adjective phrases, adverbs and adverb phrases. According to this, all pre- and postmodifiers as well as dependent clauses are also included to the TIMEX2 extent, e.g. “five days after he came back”, “nearly four decades of experience”.

The most recent annotation language for temporal expressions, TimeML [1], with the underlying TimeBank corpus [2], opens up new avenues for temporal information extraction. Besides a new specification for temporally relevant information, such as TIMEX3, EVENT, SIGNAL, TLINK etc, TimeML provides a means to capture temporal semantics by annotations with suitably defined attributes for fine-grained specification of analytical detail [7]. The annotation schema establishes new entity and relation marking tags along with numerous attributes for them. The TimeBank corpus includes 186 documents with 68.5 thousand words and 1423 TIMEX3 tags.

2.3 Approaches for Temporal Tagging

As for any recognition problem, there are two major ways to solve it – *rule-based* and *machine-learning methods*. As the temporal expression recognition is not only about to detect them but also to provide an exact match, machine learning approaches can be divided into *token-by-token classification* following B(egin)-I(nside)-O(utside) encoding and *binary constituent-based classification*, in which an entire chunk-phrase is considered for classification as a temporal expression.

2.3.1 Rule-Based Systems

One of the first well-known implementations of temporal taggers was presented in [8]. The approach relies on a set of hand-crafted and machine-discovered rules, based upon shallow lexical features. On average the system achieved a value of 0.832 for F1-measure against hand-annotated data for the exact match evaluation. The dataset used comprised a set of 22 New York Times articles and 199 transcripts of Voice of America. Another example of rule-based temporal taggers is Chronos, described in [9], which achieved the highest F1-scores in the ACE TERN 2004 of 0.926 and 0.878 for detection and exact match respectively.

Recognition of temporal expressions using TimeBank as an annotated corpus, is reported in [4] and based on a cascaded finite-state grammar (500 stages and 16000 transitions). A complex approach achieved an F1-measure value of 0.817 for exact match and 0.896 for detecting “sloppy” spans. Another known implementation for TimeBank is GUTime² – an adaptation of [8] from TIMEX2 to TIMEX3 with no reported performance level.

2.3.2 Machine Learning Recognition Systems

Successful machine learning TIMEX2 recognition systems are described in [10; 11; 12]. Proposed approaches made use of a token-by-token classification for temporal expressions represented by B-I-O encoding with a set of lexical and syntactic features, e.g., the token itself, part-of-speech tag, weekday names, numeric year etc. The performance levels are presented in Table 1. All the results were obtained on the ACE TERN dataset.

² <http://www.timeml.org/site/tarsqi/modules/gutime/index.html>

Table 1. Performance of machine learning approaches with B-I-O encoding

Approach	<i>F1</i> (detection)	<i>F1</i> (exact match)
Ahn et al. [10]	0.914	0.798
Hacioglu et al. [11]	0.935	0.878
Poveda et al. [12]	0.986	0.757

A constituent-based, also known as chunk-based, classification approach for temporal expression recognition was presented in [13]. By comparing to the previous work of the same authors [10] and on the same ACE TERN dataset, the method demonstrates a slight decrease in detection with *F1*-measure of 0.844 and a nearly equivalent *F1*-measure value for exact match of 0.787.

3 Our Approaches

The approaches presented in this section employ a supervised machine learning algorithm following a similar feature design but different classification strategies. Both classifiers implement a Maximum Entropy Model³.

3.1 Token-Based Classification Approach

Multi-class classifications, such as the one with B-I-O encoding, are a traditional way for recognition tasks in natural language processing, for example in Named Entity Recognition and chunking. For this approach we employ the OpenNLP toolkit⁴ when pre-processing the data. The toolkit makes use of the same Maximum Entropy model for detecting sentence boundaries, part-of-speech (POS) tagging and parsing tasks [14; 15]. The tokenized output along with detected POS tags is used for generating feature vectors with one of the labels from the B-I-O encoding. The feature-vector design comprises the initial token in lowercase, POS tag, character type and character type pattern. Character type and character type pattern features are implemented following Ahn et al. [10]. The patterns are defined by using the symbols X, x and 9. X and x are used to represent capital and lower-case letters respectively, 9 is used for numeric tokens. Once the character types are computed, the corresponding character patterns are produced by removing redundant occurrences of the same symbol. For example, the token “January” has character type “Xxxxxxx” and pattern “X(x)”. The same feature design is applied to each token in the context window of three tokens to the left and to the right in the sequence limited by sentence boundaries.

3.2 Constituent-Based Classification Approach

For constituent-based classification the entire phrase is under consideration for labeling as a TIMEX or not. We restrict the classification for the following phrase types and grammatical categories: nouns, proper nouns, cardinals, noun phrases,

³ <http://maxent.sourceforge.net/>

⁴ <http://opennlp.sourceforge.net/>

adjectives, adjective phrases, adverbs, adverbial phrases and prepositional phrases. For each sentence we parse the initial input line with a Maximum Entropy parser [15] and extract all phrase candidates with respect to the types defined above. Each phrase candidate is examined against the manual annotations for temporal expressions found in the sentence. Those phrases, which correspond to the temporal expressions in the sentence are taken as positive examples, while the rest are considered as a negative set. After that, for each candidate we produce a feature vector, which includes the following features: head phrase, head word, part-of-speech for head word, character type and character type pattern for head word as well as for the entire phrase.

4 Experiments, Results and Error Analysis

All experiments were conducted following 10-fold cross validation and evaluated with respect to the TERN 2004 evaluation plan [3].

4.1 Token-Based Classification Experiments

After pre-processing the textual part of TimeBank, we had a set of 26 509 tokens with 1222 correctly aligned TIMEX3 tags. The experimental results demonstrated the performance in detection of temporal expressions with precision, recall and F1-measure of 0.928, 0.628 and 0.747 respectively. When an exact match is required, the classifier performs at the level of 0.888, 0.382 and 0.532 for precision, recall and F1-measure respectively.

4.2 Constituent-Based Classification Experiments

After pre-processing the TimeBank corpus of 182 documents we had 2612 parsed sentences with 1224 temporal expressions in them. 2612 sentences resulted in 49 656 phrase candidates. After running experiments the classifier demonstrated the performance in detection of TIMEX3 tags with precision, recall and F1-measure of 0.872, 0.836 and 0.852 respectively. The experiments on exact match demonstrated a small decline and received scores of 0.866, 0.796 and 0.828 for precision, recall and F1-measure respectively.

4.3 Comparison and Improvements

Comparing the performance levels of the tested temporal taggers, we discovered the differences in classification results of chunk-based and token-based approaches with corresponding F1-measure values of 0.852 vs. 0.747 for detection, and 0.828 vs. 0.532 for exact match. Previous experiments on the ACE TERN corpus, especially those in [10; 13], reported the same phenomenon and a drop in F1-measure between detection and exact match, but the token-based approach delivers generally better results. For our experimental results we assume that the problem lies in the local token classification approach based on pure lexico-syntactic features alone. To prove this hypothesis, the next series of experiments is performed with an additional feature set, which contains the classification results obtained for preceding tokens, so called Maximum Entropy Markov Model, MEMM. The experimental setup varies the

Table 2. Performance levels for token-by-token classifications with MEMM

N	Detection			Exact match		
	<i>P</i>	<i>R</i>	<i>F1</i>	<i>P</i>	<i>R</i>	<i>F1</i>
0	0.928	0.628	0.747	0.888	0.382	0.532
1	0.946	0.686	0.793	0.921	0.446	0.599
2	0.94	0.652	0.768	0.911	0.426	0.578
3	0.936	0.645	0.762	0.905	0.414	0.566

number of previously consecutive obtained labels between 1 and 3 with the same context window size. The context is considered within the sentence only. The results of these experiments are presented in Table 2. The number of the previously obtained labels used as features is denoted by *N* – the order of the Markov Model, with *N*=0 as a baseline (see Section 3.1).

It is worth to mention that by taking into account labels obtained for preceding tokens the performance level rises and reaches the maximum at *N*=1 for both, the detection and exact match tasks, and decreases from *N*=2 onwards.

Putting these results into context, we can conclude that the chunk-based machine learning approach for temporal expression recognition performed at a comparable operational level to the state-of-the-art rule-based approach of Boguraev and Ando [4] and outperformed it in exact match. A comparative performance summary is presented in Table 3.

4.4 Error Analysis

Analyzing the classification errors, we see several causes for them. We realized that the current version of TimeBank is still noisy with respect to annotated data. An ambiguous use of temporal triggers in different context, like “*today*”, “*now*”, “*future*”, makes correct identification of relatively simple temporal expressions difficult. Apart from obviously incorrect parses, the inexact alignment between temporal expressions and candidate phrases was caused by annotations that occurred at the middle of a phrase, for example “*eight-years-long*”, “*overnight*”, “*yesterday’s*”. In total there are 99 TIMEX3 tags (or 8.1%) misaligned with the parser output, which resulted in 53

Table 3. Performance summary for the constituent-based classification (CBC) approach

	<i>P</i>	<i>R</i>	<i>F1</i>
Detection			
CBC approach	0.872	0.836	0.852
Sloppy Span			
Boguraev and Ando [4]	0.852	0.952	0.896
Exact Match			
CBC approach	0.866	0.796	0.828
Boguraev and Ando [4]	0.776	0.861	0.817

(or 4.3%) undetected TIMEX3s. Definite and indefinite articles are unsystematically left out or included into TIMEX3 extent, which introduces an additional bias in classification.

5 Conclusion and Future Work

In this paper we presented two machine learning approaches for the recognition of temporal expressions using a recent annotated corpus for temporal information, TimeBank. Two approaches were implemented: a token-by-token and a binary constituent-based classifiers. The feature design for both methods is very similar and takes into account contentual and contextual features. As the evaluation showed, both approaches provide a good performance level for the detection of temporal expressions. The constituent-based classification outperforms the token-based one with F1-measure values of 0.852 vs. 0.747. If an exact match is required, only the constituent-based classification can provide a reliable recognition with an F1-measure value of 0.828. For the same task token-based classification reaches only 0.532 in terms of F1-measure. By employing a Maximum Entropy Markov Model, the method increases in performance and reaches its maximum, when only the classification result for the previous token is used (with F1-measures of 0.793 and 0.599 for detection and exact match respectively).

Our best results were obtained by the binary constituent-based classification approach with shallow syntactic and lexical features. The method achieved a performance level of a rule-based approach presented in [4] and for the exact match task our approach even outperforms the latter. Although a direct comparison with other state-of-the-art systems is not possible, our experiments disclose a very important characteristic. While the recognition systems in the ACE TERN 2004 reported a substantial drop of F1-measure between detection and exact match results (6.5 - 11.6%), our phrase-based detector demonstrates a light decrease in F1-measure (2.4%), whereas the precision declines only by 0.6%. This important finding leads us to the conclusion that most of TIMEX3s in TimeBank can be detected at a phrase-based level with a reasonably high performance.

References

1. Pustejovsky, J., Castaño, J., Ingria, R., Saurí, R., Gaizauskas, R., Setzer, A., Katz, G.: TimeML: Robust Specification of Event and Temporal Expressions in Text. In: Fifth International Workshop on Computational Semantics (2003)
2. Pustejovsky, J., Hanks, P., Saurí, R., See, A., Day, D., Ferro, L., Gaizauskas, R., Lazo, M., Setzer, A., Sundheim, B.: The TimeBank Corpus. In: Corpus Linguistics 2003, pp. 647–656 (2003)
3. TERN, Evaluation Plan (2004), http://fofoca.mitre.org/tern_2004/tern_evalplan-2004.29apr04.pdf
4. Boguraev, B., Ando, R.K.: TimeBank-Driven TimeML Analysis. In: Annotating, Extracting and Reasoning about Time and Events. Dagstuhl Seminar Proceedings. Dagstuhl, Germany (2005)

5. Ferro, L.: TERN Evaluation Task Overview and Corpus,
http://fofoca.mitre.org/tern_2004/ferrol_TERN2004_task_corpus.pdf
6. Ferro, L., Gerber, L., Mani, I., Sundheim, B., Wilson, G.: TIDES 2003, Standard for the Annotation of Temporal Expressions (2003), <http://timex2.mitre.org>
7. Boguraev, B., Pustejovsky, J., Ando, R., Verhagen, M.: TimeBank Evolution as a Community Resource for TimeML Parsing. *Language Resource and Evaluation* 41(1), 91–115 (2007)
8. Mani, I., Wilson, G.: Robust Temporal Processing of News. In: 38th Annual Meeting on Association for Computational Linguistics, pp. 69–76 (2000)
9. Negri, M., Marseglia, L.: Recognition and Normalization of Time Expressions: ITC-irst at TERN 2004. Technical Report, ITC-irst, Trento (2004)
10. Ahn, D., Adafre, S.F., de Rijke, M.: Extracting Temporal Information from Open Domain Text: A Comparative Exploration. *Digital Information Management* 3(1), 14–20 (2005)
11. Hacioglu, K., Chen, Y., Douglas, B.: Automatic Time Expression Labeling for English and Chinese Text. In: Gelbukh, A. (ed.) *CICLing 2005*. LNCS, vol. 3406, pp. 548–559. Springer, Heidelberg (2005)
12. Poveda, J., Surdeanu, M., Turmo, J.: A Comparison of Statistical and Rule-Induction Learners for Automatic Tagging of Time Expressions in English. In: *International Symposium on Temporal Representation and Reasoning*, pp. 141–149 (2007)
13. Ahn, D., van Rantwijk, J., de Rijke, M.: A Cascaded Machine Learning Approach to Interpreting Temporal Expressions. In: *NAACL-HLT 2007* (2007)
14. Ratnaparkhi, A.: A Maximum Entropy Model for Part-of-Speech Tagging. In: *Conference on Empirical Methods in Natural Language Processing*, pp. 133–142 (1996)
15. Ratnaparkhi, A.: Learning to Parse Natural Language with Maximum Entropy Models. *Machine Learning* 34(1), 151–175 (1999)