LIMSI-COT at SemEval-2016 Task 12: Temporal relation identification using a pipeline of classifiers

Julien Tourille
LIMSI, CNRS,
Univ. Paris-Sud,
Université Paris-Saclay
julien.tourille@limsi.fr

Olivier Ferret
CEA, LIST,
Gif-sur-Yvette, F-91191 France.
olivier.ferret@cea.fr

Aurélie Névéol
LIMSI, CNRS,
Université Paris-Saclay
aurelie.neveol@limsi.fr

Xavier Tannier
LIMSI, CNRS,
Univ. Paris-Sud,
Université Paris-Saclay
xtannier@limsi.fr

Abstract

SemEval 2016 Task 12 addresses temporal reasoning in the clinical domain. In this paper, we present our participation for relation extraction based on gold standard entities (subtasks DR and CR). We used a supervised approach comparing plain lexical features to word embeddings for temporal relation identification, and obtained above-median scores.

1 Introduction

SemEval 2016 Task 12 offers 6 subtasks addressing temporal reasoning in the clinical domain using the THYME corpus (Styler IV et al., 2014). This corpus provides annotated clinical and pathological notes from colon cancer patients. The first group of subtasks concerns the identification of time and event expressions within raw text. The second group of subtasks deals with the identification of temporal relations. The latter consists of two subtasks. In the *Document Creation Time Relation* subtask (DR), participants are challenged to identify relations between the events and the document creation time. For the *Container Relation* subtask (CR), participants have to identity container relations between entities. Participants may submit either a complete system extracting entities and relations or focus on either the entity extraction or relation extraction (using the gold standard entities provided by the organizers). More details about the task and the definition of each subtask can be found in Bethard et al. (2016).

In this paper, we present our submission for the CR and DR subtasks based on gold-standard entities (phase 2). Our global approach, which is illustrated in Figure 1, tackles the identification of temporal relations as a set of supervised classification tasks. We submitted two runs, one using plain lexical features and one using word embeddings computed on a large clinical corpus. We obtained scores well above the median scores in both subtasks.

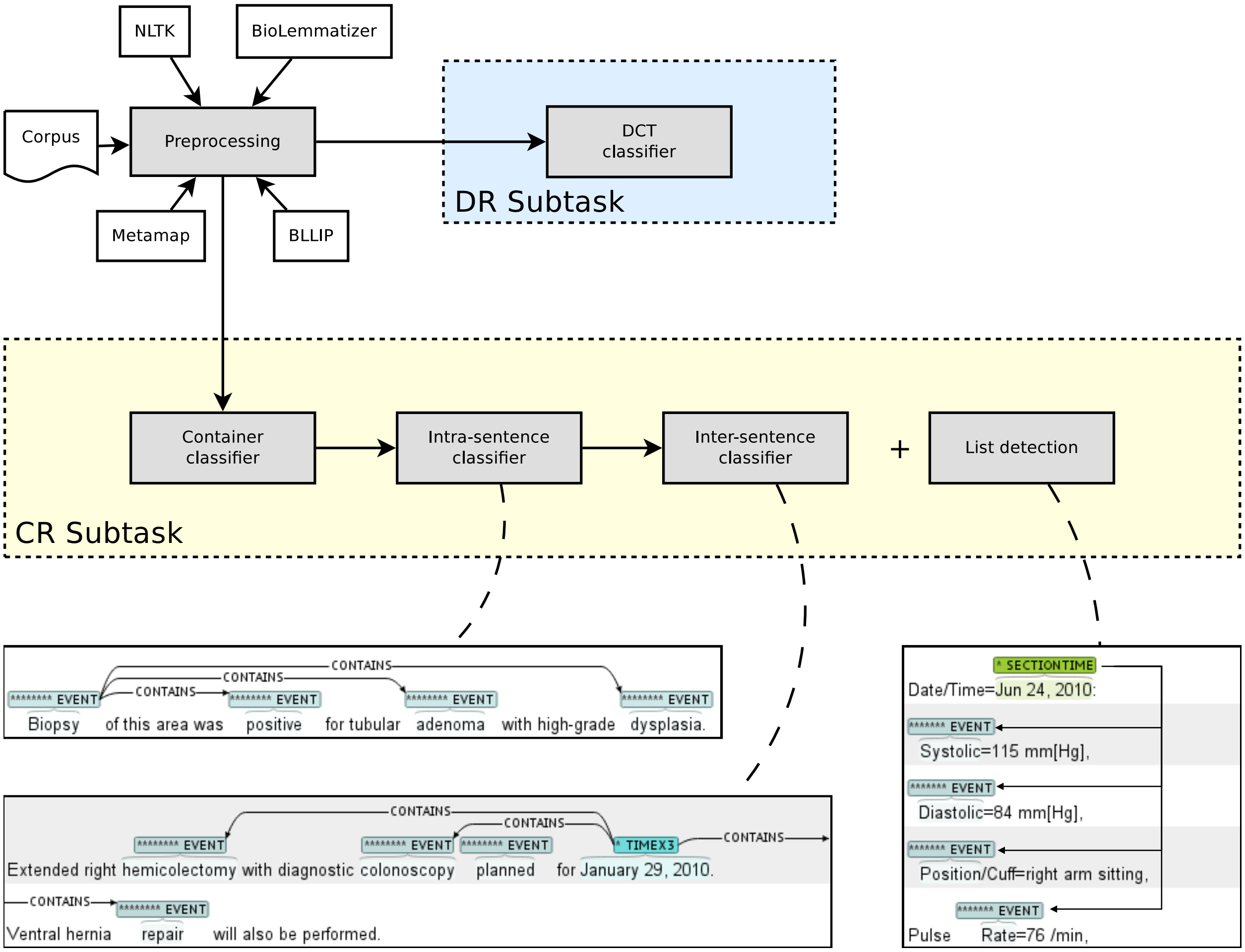
The remainder of this paper is organized as follows. Section 2 presents our system for the DR subtask while Section 3 describes our system for the CR subtask. Section 4 gives an overview of the system implementation. Finally, Section 5 presents our results.

2 Document Creation Time Relation (DR) Subtask

We treated the subtask as a supervised classification problem where each `EVENT` entity was classified into four categories (*Before*, *Before-Overlap*, *Overlap*, *After*).

We extracted lexical, contextual and structural features from the texts. Regarding the lexical features of `EVENT` entities, we took their surface forms, their gold standard attributes (*type*, *modality*, *degree* and *polarity*), their lemma(s)[1], as well as their Part-Of-Speech (POS) and Coarse Part-Of-Speech (CPOS) tags. We also extracted the semantic types

[1] The span of an `EVENT` entity can overlap with several tokens.



Biopsy of this area was positive for tubular adenoma with high-grade dysplasia.

Extended right hemicolectomy with diagnostic colonoscopy planned for January 29, 2010.

Ventral hernia repair will also be performed.

Date/Time=Jun 24, 2010:

Systolic=115 mm[Hg],

Diastolic=84 mm[Hg],

Position/Cuff=right arm sitting,

Pulse Rate=76 /min,

Figure 1: Overview of the processing chain

and semantic groups of the medical entities that have been detected by Metamap (Aronson and Lang, 2010) and that share a span overlap with the EVENT entities.

Concerning the contextual features, we extracted the gold standard entities that were present in the right and left contexts[2] within the sentence. We added the corresponding lemmas, surface forms, types, POS and CPOS tags, as well as the corresponding semantic groups and semantic types. We also added the lemmas of the tokens occurring in both left and right contexts. At the section level, we added entities occurring before and after the EVENT entity within the section. We used the same set of features as the one we used for the intra-sentence context: surface forms, lemmas, types, POS and CPOS tags, semantic types and semantic groups. We also added the gold standard attributes of these entities.

Regarding the structural features, we used the position of the sentence within the section and the position of the section within the document. We added the number of tokens and entities occurring before and after the EVENT entity within the section, and the number of entities figuring before and after the EVENT entity at the document level.

3 Container Relation (CR) Subtask

3.1 Principles

Similarly to the DR subtask, we treated the CR subtask as a supervised classification problem and more particularly, as a binary classification task applied to pairs of EVENT and/or TIME entities in documents. However, considering all possible pairs of such entities for building the training set without any scope

[2] There is no size restriction on the contexts

restriction would lead to unbalanced training examples where the negative examples largely outnumber the positive examples. Hence, some choices have been made to reduce the number of training examples.

The analysis of the training corpus shows that a large majority – around 76% – of the CONTAINS relations are intra-sentence relations, which means that the problem of their scope actually occur for one quarter only. The remaining relations, called inter-sentence relations, spread over at least two sentences. Given this context, we have built two separate classifiers, the first one for the intra-sentence relations, the second one for the inter-sentence relations. This distinction has two main advantages: first, it reduces drastically the number of negative examples, which produces better results as we observed on our development set; second, the intra-sentence classifier can benefit from a larger and richer set of features coming from sentence-level linguistic analyzers.

Concerning the inter-sentence relations, considering all pairs of EVENT and/or TIME entities would still give us a very large amount of negative examples. We first observed that all of them were contained within sections (no relation overlaps section boundaries). Within the scope of a section, we further noticed that inter-sentence relations within a 3-sentence window covered approximately 89% over all existing relations. A wider window would bring too much noise while giving us a very small bump on coverage. Table 1 shows the number of covered relations according to the size of the window, expressed in the number of sentences. The first line corresponds to the intra-sentence level (window=1).

To further reduce the number of candidates for both inter- and intra-sentence classifiers, we transformed the 2-category problem (*contains vs. no-relation*) into a 3-category classification problem (*contains*, *is-contained* or *no-relation*). Instead of considering all permutations of events within a sentence or a sentence-window, we considered all pairs of events from left to right, changing when necessary the *contains* relations into *is-contained* relations. This strategy allowed us to divide by a factor of two the number of candidates. We obtained 111,349 pairs for the intra-sentence classifier and 311,284 pairs for the inter-sentence classifier.

win.[a]	nb. of rel.[b]	total[c]
1	13,304	13,304 (76.30%)
2	1,463	14,767 (84.69%)
3	752	15,519 (89.00%)
4	497	16,016 (91.85%)
5	364	16,380 (93.94%)
6	151	16,531 (94.80%)

[a] Sentence window
[b] Number of CONTAINS relations
[c] Cumulative count of CONTAINS relations

Table 1: CONTAINS relations according to sentence window size. Window of size 1 corresponds to the intra-sentence level.

Some entities are more likely to be containers. By example, SECTIONTIME and TIMEX entities are, by nature, potential containers. This is also the case for some medical events. For instance, a *surgical operation* may contain other events such as *bleeding* or *suturing*. It will not be the same with the two latter in most cases. Following this observation, we have built a model to classify entities as being a potential *container* or *not*. As we will show in Section 5, this classifier obtain a high accuracy. We used its output as feature for our intra- and inter-sentence classifiers.

Finally, we developed a rule-based module to capture specific CONTAINS relations. There are some strong regularities in the handling of laboratory results where the first SECTIONTIME contains all the results, which are expressed with EVENT entities (see an example in Figure 1). The module we have built aims at capturing these inter-sentential regularities with the use of rules.

To summarize, our system is composed of four modules:

1. **Container detection module**: entities are classified according to whether or not they are the source of one or more CONTAINS relations;

2. **Intra-sentence relation module**: combinations of entities within sentences are considered (relations *contains*, *is-contained* or *no-relation*);

3. **Inter-sentence relation module**: combinations of entities within a 3-sentence window are

considered. We use the same relation classes as those used for intra-sentence relations;

4. **List detection module**: specific laboratory results written as lists are handled via manual rules.

3.2 Feature Extraction

For the container classifier, we used the form and the type of the considered entity, as well as its gold standard attributes. We extracted the semantic types and semantic groups of the entities that have been detected by Metamap and that share an overlap with the considered entity. We also extracted its lemma(s), POS and CPOS tags. Concerning the contextual features, we extracted the entities that are present in both left and right contexts within the sentence boundaries. Similarly to the DR subtask, we used entity forms, types, semantic groups and semantic types, POS and CPOS tags and lemmas. We also added the tokens from both left and right contexts. We used the corresponding lemmas, POS and CPOS tags. Finally, we added the number of entities within the sentence and the number of entities before and after the considered entity within the sentence.

For the intra-sentence classifier, we extracted the forms, semantic types and semantic groups, and the gold standard attributes of the two considered entities. We added their lemmas and their POS and CPOS tags. We added the number of tokens occurring between the two entities. We also extracted the entities occurring between the two considered entities. We used their entity types, gold standard attributes, semantic groups and semantic types. We added the number of entities that have been classified as containers by our container classification model and the number of entities that appear between the pair of entities. Finally, we added the syntactic paths[3] between the two entities (from left to right). We also added the results of our contain classification model for the two considered entities.

For the inter-sentence classifier, we used similar features. We extracted the forms, types, gold standard attributes and semantic groups and semantic types of the two considered entities. We also extracted features from the results of the intra-sentence classifier. We specified whether the considered entities are intra-sentence containers or are contained by other entities at the intra-sentence level. We also extracted entities that are positioned between the two considered entities. We used entity types, gold-standard attributes, semantic groups and semantic types. We also added a feature specifying if these entities are containers at the intra-sentence level and the number of entities between the considered pair of entities. Finally we added the positions, at the section level, of the sentences in which the considered events are embedded.

4 System Implementation

4.1 Strategies

We implemented two strategies to represent the lexical features in both the DR and CR subtasks. In the first one, we used the plain forms of the different lexical attributes we mentioned (Strategy 1). In the second strategy, we substituted the lemmas and forms with word embeddings (Strategy 2). These embeddings have been computed on the Mimic 2 corpus (Saeed et al., 2011) using the word2vec tool with a CBOW model[4] (Mikolov et al., 2013). We used the mean of the vectors for multi-word units. Lexical contexts are thus represented by 200-dimensional vectors. When several contexts are considered e.g. right and left, several vectors are used.

4.2 Algorithm Selection

A grid search strategy was applied to select the most appropriate machine learning algorithm and its parameters. For Strategy 1, three algorithms were considered in our search: Random Forests, Linear Support Vector Machine (liblinear) and Support Vector Machine with a RBF kernel (libsvm). For Strategy 2, we only considered the Linear Support Vector Machine for the CR task and Random Forests for the DR task.

In both cases, 5-fold cross-validation was used to choose the algorithm and its parameters. We also implemented statistical feature selection as part of the grid search for Strategy 1 reducing progressively the number of attributes, using ANOVA F-test.

[3] Several paths are considered when the entities spread over more that one token.

[4] Parameters used during computation: min-count: 5; vector size: 200; window: 20; number of word classes: 1000; frequency threshold: 1e-3.

Run	Classifier	Algorithm	Parameters	% feat.[a]
1[c]	CONTAINER	SVM (RBF)	C=10, gamma=0.01	60
	INTRA	SVM (RBF)	C=10, gamma=0.01	60
	INTER	SVM (RBF)	C=1000, gamma=0.01	100
	DCT	SVM (Linear)	C=1, tol[b]=0.0001, normalization=l2, loss function=hinge	100
2[d]	CONTAINER	LinearSVM	C=1, tol=0.01, normalization=l2, loss function=hinge	100
	INTRA	SVM (Linear)	C=1, tol=0.01, normalization=l2, loss function=squared hinge	100
	INTER	SVM (Linear)	C=1000, tol=0.01, normalization=l2, loss function=hinge	100
	DCT	Random Forests	max features=auto, criterion=entropy, estimators=100	100

[a] Percentage of feature space kept for final submission (using ANOVA F-test)
[b] Tolerance for stopping criteria
[c] Using plain text features
[d] Using word embeddings

Table 2: Machine learning algorithms and parameters used for the final submission

The machine learning algorithms used for the final submission are presented in Table 2 together with their parameters and the percentage of the feature space kept after statistical feature selection. We used the Scikit-learn machine learning library (Pedregosa et al., 2011) for both implementing our classification models and performing statistical feature selection.

4.3 Corpus Preprocessing

We applied a four-step preprocessing on the 440 texts that were provided for the subtasks. First, we used NLTK (Loper and Bird, 2002) to segment the texts into sentences with the *Punkt Sentence Tokenizer* pre-trained model for English provided within the framework.

The second step consisted of parsing the resulting sentences. For this task, we used the BLLIP Reranking Parser (Charniak and Johnson, 2005) and a pre-trained biomedical parsing model (McClosky, 2010).

In the third step, we lemmatized the corpus using BioLemmatizer (Liu et al., 2012), a tool built for processing the biomedical literature. We used the Part-Of-Speech tags from the previous step as parameters for the lemmatization.

The last step consisted in using Metamap (Aronson and Lang, 2010) to detect biomedical events and linking them, after disambiguation, to their related UMLS® (Unified Medical Language System) concept. We chose to keep biomedical entities that had a span overlapping with at least one entity of the gold standard.

5 Results and Discussion

In Table 3, we present the cross-validation accuracies of our DCT and Container models over the development corpus. For DCT, we obtain high performance with Strategy 1, which is based on plain lexical features. Strategy 2, which exploits words embeddings, gives lower performance. Concerning the Container model, we obtain high performance with both strategies.

Model	plain text	word embeddings
DCT	0.873	0.778
CONTAINER	0.917	0.924

Table 3: DCT and CONTAINER model accuracies

We submitted two runs with our system, one for each strategy. The results for both subtasks are presented in Tables 4 and 5.

Concerning the DR subtask, we obtained above-median scores (median score: 0.724) for both runs. The second run, which relies on word embeddings to represent the lexical features of the EVENT entities, achieves better performance. These results are consistent with what was expected during the cross-validation process using the development set. The fact that the second strategy achieves the best performance is however in contradiction with the scores obtained during cross-validation, where Strategy 1 performed best.

In the CR subtask, we obtained above-median F1 for the first run and median scores for the second run (median score: 0.449). Using plain lexical features gives us a more balanced system than using word embeddings. With a F1 of 0.538, our system achieves performance close to the best system (0.573), thus validating our modeling choices. These results are consistent with those we obtained when testing against the development part of the corpus. The reasons for the decrease in recall when using the second strategy are however unclear and need further investigation.

Run	ref[a]	pred[b]	corr[c]	P	R	F1
1[d]	18,990	18,989	14,603	0.769	0.769	0.769
2[e]	18,990	18,989	15,317	0.807	0.807	**0.807**

[a] Number of gold standard relations
[b] Number of predicted relations
[c] Number of correct predictions
[d] Using plain text features
[e] Using word embeddings

Table 4: DR subtask - Evaluation script output

Run	ref[a]	pred[b]	corr[c]	P	R	F1
1[d]	5,894	3,755	2,642 2,570	0.704	**0.436**	**0.538**
2[e]	5,894	2,544	1,911 1,889	**0.751**	0.320	0.449

[a] Number of gold standard relations
[b] Number of predicted relations
[c] Number of correct relations (without and with temporal closure)
[d] Using plain text features
[e] Using word embeddings

Table 5: CR subtask - Evaluation script output

Acknowledgments

We would like to thank the Mayo Clinic for granting us the permission to use the THYME corpus. This work has been funded by Labex Digicosme, operated by the Foundation for Scientific Cooperation (FSC) Paris-Saclay. This work was supported in part by the French National Agency for Research under grant CABeRneT, ANR-13-JS02-0009-01.

References

Alan R Aronson and François-Michel Lang. 2010. An overview of MetaMap: historical perspective and recent advances. *Journal of the American Medical Informatics Association*, 17(3):229–236.

Steven Bethard, Guergana Savova, Wei-Te Chen, Leon Derczynski, James Pustejovsky, and Marc Verhagen. 2016. SemEval-2016 Task 12: Clinical TempEval. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval 2016)*, San Diego, California, June. Association for Computational Linguistics.

Eugene Charniak and Mark Johnson. 2005. Coarse-to-Fine n-Best Parsing and MaxEnt Discriminative Reranking. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 173–180, Ann Arbor, Michigan, June. Association for Computational Linguistics.

Haibin Liu, Tom Christiansen, William A Baumgartner Jr, and Karin Verspoor. 2012. BioLemmatizer: a lemmatization tool for morphological processing of biomedical text. *Journal of Biomedical Semantics*, 3(3).

Edward Loper and Steven Bird. 2002. NLTK: The Natural Language Toolkit. In *Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics*, pages 63–70, Philadelphia, Pennsylvania. Association for Computational Linguistics.

David McClosky. 2010. *Any Domain Parsing: Automatic Domain Adaptation for Natural Language Parsing*. Ph.D. thesis, Department of Computer Science, Brown University.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Mohammed Saeed, Mauricio Villarroel, Andrew T. Reisner, Gari Clifford, Li-Wei Lehman, George Moody, Thomas Heldt, Tin H. Kyaw, Benjamin Moody, and Roger G. Mark. 2011. Multiparameter Intelligent Monitoring in Intensive Care II (MIMIC-II): A publicaccess intensive care unit database. *Critical Care Medicine*, 39:952–960, May.

William Styler IV, Steven Bethard, Sean Finan, Martha Palmer, Sameer Pradhan, Piet de Groen, Brad Erickson, Timothy Miller, Chen Lin, Guergana Savova, and James Pustejovsky. 2014. Temporal Annotation in the Clinical Domain. *Transactions of the Association for Computational Linguistics*, 2:143–154.