# Structured Learning for Temporal Relation Extraction from Clinical Records

Artuur Leeuwenberg and Marie-Francine Moens
Department of Computer Science
KU Leuven, Belgium
f tuur.leeuwenberg, sien.moens g@cs.kuleuven.be



## Abstract

We propose a scalable structured learning model that jointly predicts temporal relations between events and temporal expressions (TLINKS), and the relation between these events and the document creation time (DCTR). We employ a structured perceptron, together with integer linear programming constraints for document-level inference during training and prediction to exploit relational properties of temporality, together with global learning of the relations at the document level. Moreover, this study gives insights in the results of integrating constraints for temporal relation extraction when using structured learning and prediction. Our best system outperforms the state-of-the art on both the CONTAINS TLINK task, and the DCTR task.

## 1 Introduction

Temporal information is critical in many clinical areas (Combi and Shahar, 1997). A big part of this temporal information is captured in the free text of patient records. The current work aims to improve temporal information extraction from such clinical texts. Extraction of temporal information from clinical text records can be used to construct a time-line of the patient's condition (such as in Figure 1). The extracted time-line can help clinical researchers to better select and recruit patients with a certain history for clinical trials. Moreover, the time-line is crucial for making a good patient prognosis and clinical decision support (Onisko et al., 2015; Stacey and McGregor, 2007).

Temporal information extraction can be divided into three sub-problems: (1) the detection of events ($E_e$); (2) the detection of temporal expressions ($E_t$); and (3) the detection of temporal relations between them. In the clinical domain, events include medical procedures, treatments, or symptoms (e.g. *colonoscopy*, *smoking*, *CT-scan*). Temporal expressions include dates, days of the week, months, or relative expressions like *yesterday*, *last week*, or *post-operative*. In this work, we focus on the last sub-problem, extraction of temporal relations (assuming events and temporal expressions are given). As a small example of the task we aim to solve, given the following sentence:

> In 1990 the patient was diagnosed and received surgery directly afterwards.

in which we assume that the events *diagnosed* and *adenocarcinoma*, and the temporal expression *1990* are given, we wish to extract the following relations:

CONTAINS(1990, diagnosed)

CONTAINS(1990, surgery)

BEFORE(diagnosed, surgery)

BEFORE(diagnosed, $d$)

BEFORE(surgery, $d$)

where $d$ stands for the document creation time.

Our work leads to the following contributions: First, we propose a scalable structured learning model that jointly predicts temporal relations between events and temporal expressions (TLINKS), and the relation between these events and the document creation time (DCTR). In contrast to existing approaches which detect relation instances separately, our approach employs a structured perceptron (Collins, 2002) for global learning with joint inference of the temporal relations on a document level. Second, we ensure scalability through using integer linear programming (ILP)

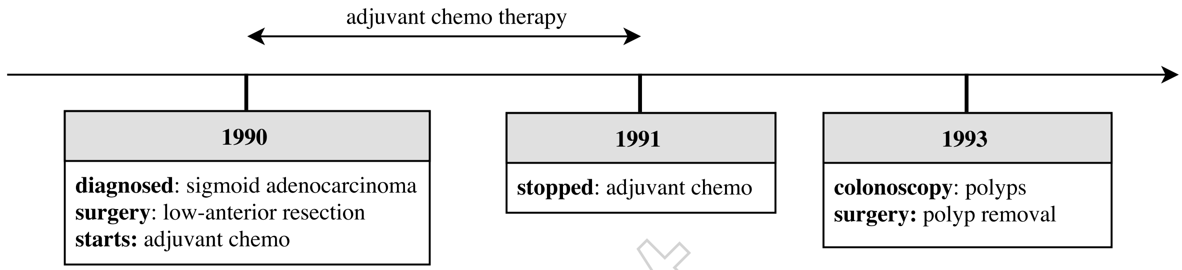


Figure 1: Fragment of a (partial) patient time-line.



constraints with fast solvers, loss augmented subsampling, and good initialization. Third, this study leads to valuable insights on when and how to make inferences over the found candidate relations both during training and prediction and gives an in-depth assessment of the use of additional constraints and global features during inference. Finally, our best system outperforms the state-of-the-art of both the CONTAINS TLINK task, and the DCTR task.

## 2 Related Work

There have been two shared tasks on the topic of temporal relation extraction in the clinical domain: the I2B2 Temporal Challenge (Sun et al., 2013), and more recently the Clinical TempEval Shared Task with two iterations, one in 2015 and one in 2016 (Bethard et al., 2014; Bethard et al., 2015; Bethard et al., 2016). In the I2B2 Temporal Challenge eight types of relations were initially annotated. However, due to low inter-annotator agreement these were merged to three types of temporal relations, OVERLAP, BEFORE, and AFTER. Good annotation of temporal relations is dif cult, as annotators frequently miss relation mentions. In the Clinical TempEval Shared tasks the THYME corpus is used (Styler IV et al., 2014), with a different annotation scheme that aims at annotating those relations that are most informative w.r.t. the time-line, and gives less priority to relations that can be inferred from the others. This results in two categories of temporal relations: The relation between each event and the document creation time (DCTR), dividing all events in four temporal buckets (BEFORE, BEFORE/OVERLAP, OVERLAP, AFTER). These buckets are called narrative containers (Pustejovsky and Stubbs, 2011). And second, relations between temporal entities that both occur in the text (TLINKS). TLINKS may occur between events ($E_e$ $E_e$), and between events and temporal expressions ($E_e$ $E_t$ and $E_t$ $E_e$). The TLINK types (and their relative frequency in the THYME corpus) are CONTAINS (64,42%), OVERLAP (15,19%), BEFORE (12,65%), BEGINS-ON (6.15%), and ENDS-ON (1.59%). The relations AFTER, and DURING are expressed in terms of their inverse, BEFORE, and CONTAINS respectively. In our experiments, we use the THYME corpus for its relatively high inter-annotator agreement (particularly for CONTAINS).

To our knowledge, in all submissions (4 in 2015, and 10 in 2016) of Clinical TempEval the task is approached as a classical entity-relation extraction problem, and the predictions for both categories of relations are made independently from each other, or in a one way dependency, where the containment classi er uses information about the predicted document-time relation. Narrative containment, temporal order, and document-time relation have very strong dependencies. Not modeling these may result in inconsistent output labels, that do not result in a consistent time-line.

An example of inconsistent labeling is given in Figure 2. The example is inconsistent when assigning the AFTER label for the relation between lesion and the document-time. It is inconsistent because we can also infer that *lesion* occurs BEFORE the document-time, as the *colonoscopy* event occurs before the document-time, and the *lesion* is contained by the *colonoscopy*.

Temporal inference, in particular *temporal closure*, is frequently used to expand the training data (Mani et al., 2006; Chambers and Jurafsky, 2008; Lee et al., 2016; Lin et al., 2016b), most of the times resulting in an increase in performance, and is also taken into account when evaluating the predicted labels (Bethard et al., 2014; UzZaman and Allen, 2011). Only very limited research regards the modeling of temporal dependencies into the machine learning model. (Chambers and Juraf-

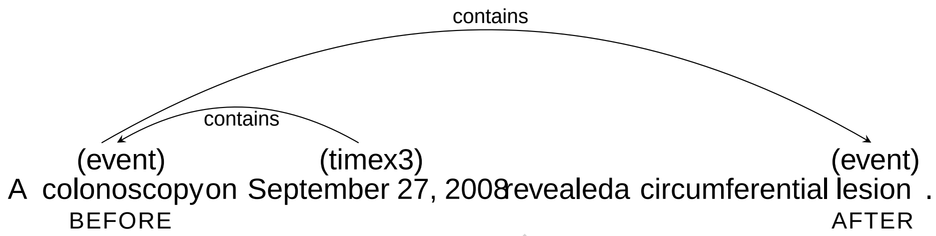


Figure 2: Example of inconsistent output labeling. Containment is indicated by directed edges, and the relation to the document-time by small caps below the events.



sky, 2008) and (Do et al., 2012) modeled label dependencies when predicting TimeBank TLINKS (Pustejovsky et al., 2003). They trained local classifiers and used a set of global temporal label constraints. Integer linear programming was employed to maximize the score from the local classifiers, while satisfying the global label constraints at prediction time. For both, this gave a significant increase in performance, and resulted in consistent output labels.

(Yoshikawa et al., 2009) modeled the label dependencies between TLINKS and DCTR with Markov Logic Networks (MLN), allowing for soft label constraints during training and prediction. However, MLN can sometimes be sub-optimal for text mining tasks w.r.t. time efficiency (Mojica and Ng, 2016). Quite recently, for a similar problem, spatial relation extraction, (Kordjamshidi et al., 2015) used an efficient combination of a structured perceptron or structured support vector machine with integer linear programming. In their experiments, they compare a local learning model (LO), a local learning model with global inference at prediction time (L+I), and a structured learning model with and without inference during training (IBT+I, and IBT-I respectively). In their experiments L+I gave better results than LO, but a more significant improvement was made when using structured learning in contrast to local learning.

In this work, we aim to jointly predict TLINKS and DCTR in a structured learning model with inference during training and prediction, to assess inference with temporal constraints of (Chambers and Jurafsky, 2008; Do et al., 2012) for the THYME relations, and to experiment with both local, and document-level inference for temporal information extraction in the clinical domain.

## 3 The Model

For jointly learning both tasks on a document level, we employ a structured perceptron learning paradigm (Collins, 2002). The structured perceptron model uses a joint feature function $\Phi(X, Y)$ to represent a full input document $X$ with a label assignment $Y$. During training the model learns a weight vector $\lambda$ to score how good the label assignment is. Predicting label assignment $Y$ for a document $X$ corresponds to finding the $Y$ with the maximal score. In the following sub-sections we define the joint feature function $\Phi$, describe the prediction procedure of the model, and provide how we train the model (i.e. learn a good $\lambda$).

### 3.1 Joint Features

To compose the joint feature function, we first define two local feature functions: $\phi_{tl} : (x, y) \rightarrow \mathbb{R}^p$ assigns features for the local classifications regarding TLINKS (with possible labels $L_{tl}$ = {CONTAINS, BEFORE, OVERLAP, BEGINS-ON, ENDS-ON, NO_LABEL}), and a second local feature function $\phi_{dr} : (x, y) \rightarrow \mathbb{R}^q$, for local features regarding document-time relation classification (with labels $L_{dr}$ = {BEFORE, BEFORE_OVERLAP, OVERLAP, AFTER}). The features used by these local feature functions are given in Table 1.

From these, we define a joint feature function $\Phi_{joint} : (X, Y) \rightarrow \mathbb{R}^{p+q}$, that concatenates $(\oplus)$ the summed local feature vectors, creating the feature vector for the global prediction task (predicting all labels in the document for both sub-tasks at once). $\Phi_{joint}$ is defined in Equation 1, where $C_{tl}(X)$ and $C_{dr}(X)$ are candidate generation functions for the TLINK sub-task, and DCTR sub-task respectively (further explained in Section 3.2).

| Features | dr | tl |
|---|---|---|
| String features for tokens and POS of each entity | X | X |
| String features for tokens and POS in a window of size 3, 5, left and right of each entity | X | X |
| Boolean features for entity attributes (event polarity, event modality, event degree, and type) | X | X |
| String feature for the token and POS of the closest verb | X | |
| String feature for the token and POS of the closest left and right entity | X | |
| String features for the token 1, 2, 3-grams and POS 1, 2, 3-grams in-between the two entities | | X |
| Dependency path between entities (consisting of POS and edge labels) | | X |
| Boolean feature on if the rst argument occurs before the second (w.r.t. word order) | | X |

Table 1: Features of the local feature functions of each sub-task, $_{tl}$ for TLINKS, and $_{dr}$ for DCTR.

$$_{joint}(X;Y) = \sum_{x \in C_{dr}(X)} \sum_{l \in L_{dr}} {}_{dr}(x;l) \quad \sum_{x \in C_{tl}(X)} \sum_{l \in L_{tl}} {}_{tl}(x;l) \qquad (1)$$

| Feature | Description |
|---|---|
| $_{sdr}$ | Bigram and trigram counts of subsequent DCTR-labels in the document |
| $_{drtl}$ | Counts of DCTR-label pairs of the entities of each TLINK |

Table 2: Global (document-level) features.

### 3.2 Local Candidate Generation

For each document $X$, we create local candidates for both sub-tasks. In this work, we assume that event ($E_e$) and temporal expression ($E_t$) annotations are provided in the input. The DCTR-candidates in document $X$ are then given by $C_{dr}(X)$, which returns all events in the document, i.e. $E_e(X)$. $C_{tl}(X)$ returns all TLINK candidates, i.e. $E_e(X) \cup E_t(X) \quad E_e(X)$. In our experiments we usually restrict the number of candidates generated by $C_{tl}$ to gain training and prediction speed (without signi cant loss in performance). This is explained further in Section 4.3.

### 3.3 Global Features

We also experiment with a set of global features, by which we mean features that are expressed in terms of multiple local labels. The global features are speci ed in Table 2. Global features are de ned by a feature function $_{global}(X;Y) \rightarrow \mathbb{R}^r$ and have their corresponding weights in weight vector . When using global features $_{global}$ is concatenated with the joint feature function $_{joint}$ to form the nal feature function , as show in in Equation 2.

$$(X;Y) = {}_{joint}(X;Y) \quad {}_{global}(X;Y) \qquad (2)$$

When not using global features, we use only the joint features, as shown in Equation 3.

$$(X;Y) = {}_{joint}(X;Y) \qquad (3)$$

### 3.4 Prediction

The model assigns a score to each input document $X$ together with output labeling $Y$. The score for $(X;Y)$ is de ned as the dot product between the learned weight vector and the outcome of the joint feature function $(X;Y)$, as shown in Equation 4.

$$S(X;Y) = \quad (X;Y) \qquad (4)$$

The prediction problem for an input document $X$ is nding the label assignment $Y$ that maximizes the score $S$ based on the weight vector, shown in Equation 5.

$$\hat{Y}_k = \arg\max_{Y} S(X;Y) \qquad (5)$$

We use integer linear programming (ILP) to solve the prediction problem in Equation 5. Each possible local decision is modeled with a binary decision variable. For each local relation candidate input $x_{i;j}$ (for the relation between $i$ and $j$) a binary decision variable $w^l_{i;j}$ is used for each potential label $l$ that could be assigned to $x_{i;j}$, depending on the sub-task. The objective of the integer linear program, given in Equation 6, is to maximize the sum of the scores of local decisions. In all equations the constant $dct$ refers to the document-creation time. The objective is maximized under two sets of constraints, given in Equations 7 and 8, that express that each candidate is assigned exactly one label, for each sub-task.

$$O = \arg\max_{W} \sum_{x_{i;d} \in C_{dr}(X)} \sum_{l \in L_{dr}} w^{l}_{i;d}\, S(x_{i;d}; y^{l}_{i;d}) + \sum_{x_{i;j} \in C_{tl}(X)} \sum_{l \in L_{tl}} w^{l}_{i;j}\, S(x_{i;j}; y^{l}_{i;j}) \quad (6)$$

$$\forall_{i} : \sum_{l \in L_{dr}} w^{l}_{i;d} = 1 \quad (7)$$

$$\forall_{i;j} : \sum_{l \in L_{tl}} w^{l}_{i;j} = 1 \quad (8)$$

For solving the integer linear program we use Gurobi (Gurobi Optimization, 2015).

#### 3.4.1 Temporal Label Constraints

Because temporal relations are interdependent, we experimented with using additional constraints on the output labeling. The additional temporal constraints we experiment with are shown in Table 3. Constraints are expressed in terms of the binary decision variables used in the integer linear program.

In Table 3, constraints $C_{Ctrans}$, and $C_{Btrans}$ model transitivity of CONTAINS, and BEFORE respectively. Constraints $C_{CBB}$, and $C_{CAA}$ model the consistency between TLINK relation CONTAINS and DCTR relations BEFORE, and AFTER respectively (resolving the inconsistent example of $C_{CBB}$ in section 1, and Figure 2). Similarly, $C_{BBB}$, and $C_{BAA}$ model the consistency between TLINK relation BEFORE and DCTR relations BEFORE, and AFTER.

Constraints can be applied during training and prediction, as Equation 5 is to be solved for both. If not mentioned otherwise, we use constraints both during training and prediction.

### 3.5 Training

The training procedure for the averaged structured perceptron is given by Algorithm 1, for $I$ iterations, on a set of training documents $T$. Notice that the prediction problem is also present during training, in line 6 of the algorithm. Weight vector is usually initialized with ones, and updated when the predicted label assignment $\hat{Y}_k$ for input document $X_k$ is not completely correct. The structured perceptron training may suffer from overfitting. Averaging the weights over the training examples of each iteration is a commonly used way to counteract this handicap (Collins, 2002; Freund and Schapire, 1999). In Algorithm 1, $c$ is used to count the number of training updates, and $_a$ as a cache for averaging the weights. We also employ local loss-augmented negative sub-sampling, and local pre-learning to address class-imbalance and training time.

**Algorithm 1** Averaged Structured Perceptron

Require: ; $_a$; $c$; $I$; $T$
1: $c \;\; 0$
2: $h\ 1; :::; 1i$
3: $_a\ h\ 1; :::; 1i$
4: for $i$ in $I$ do
5: for $k$ in $T$ do
6: $\hat{Y}_k \;\; \arg\max_{Y} \; (X_k; Y)$
7: if $\hat{Y}_k \neq Y_k$ then
8: $+ \; (X_k; Y_k) \;\; (X_k; \hat{Y}_k)$
9: $_a \;\; _a + c \;\; (X_k; Y_k) \;\; c \;\; (X_k; \hat{Y}_k)$
10: $c \;\; c + 1$
return $_a = c$

#### 3.5.1 Loss-augmented Negative Sub-sampling

For the TLINK sub-task, we have a very large negative class (NO_LABEL) and a relatively small positive class (the other TLINK labels) of training examples. To speed up training convergence and address class imbalance at the same time, we subsample negative examples during training. Within a document $X$, for each positive local training example $(x_{positive}; y_{positive})$ we take 10 random negative examples and add the negative example $(x_{negative}; y_{no\_label})$ with the highest score for relation $y_{positive}$, i.e. $S(x_{negative}; y_{positive})$. This cutting plane optimization gives preference to negative training examples that are more likely to be classified wrongly, and thus can be learned from (in an online manner), and it provides only one negative training example for each positive training example, balancing the TLINK classes.

#### 3.5.2 Local Initialization

To reduce training time, we don't initialize with ones, but we train a perceptron for both local subtasks, based on the same local features mentioned in Table 1, and use the trained weights to initialize for those features. A similar approach was used by (Weiss et al., 2015) for dependency parsing. Details on the training parameters of the perceptron are given in Section 4.3.

## 4 Experiments

We use our experiments to look at the effects of four modeling settings.

| Abbrev. | Label Dependencies | Constraints |
|---|---|---|
| $C_{Ctrans}$ | $\text{CONTAINS}_{i,j} \wedge \text{CONTAINS}_{j,k} \rightarrow \text{CONTAINS}_{i,k}$ | $\forall_{i,j,k}: w^{contains}_{i,k} \quad w^{contains}_{i,j} \quad w^{contains}_{j,k} \quad 1$ |
| $C_{Btrans}$ | $\text{BEFORE}_{i,j} \wedge \text{BEFORE}_{j,k} \rightarrow \text{BEFORE}_{i,k}$ | $\forall_{i,j,k}: w^{before}_{i,k} \quad w^{before}_{i,j} \quad w^{before}_{j,k} \quad 1$ |
| $C_{CBB}$ | $\text{CONTAINS}_{i,j} \wedge \text{BEFORE}_{i,d} \rightarrow \text{BEFORE}_{j,d}$ | $\forall_{i,j}: w^{before}_{j,d} \quad w^{contains}_{i,j} \quad w^{before}_{i,d} \quad 1$ |
| $C_{CAA}$ | $\text{CONTAINS}_{i,j} \wedge \text{AFTER}_{i,d} \rightarrow \text{AFTER}_{j,d}$ | $\forall_{i,j}: w^{after}_{j,d} \quad w^{contains}_{i,j} \quad w^{after}_{i,d} \quad 1$ |
| $C_{BBB}$ | $\text{BEFORE}_{i,j} \wedge \text{BEFORE}_{j,d} \rightarrow \text{BEFORE}_{i,d}$ | $\forall_{i,j}: w^{before}_{i,d} \quad w^{before}_{i,j} \quad w^{before}_{j,d} \quad 1$ |
| $C_{BAA}$ | $\text{BEFORE}_{i,j} \wedge \text{AFTER}_{i,d} \rightarrow \text{AFTER}_{j,d}$ | $\forall_{i,j}: w^{after}_{j,d} \quad w^{before}_{i,j} \quad w^{after}_{i,d} \quad 1$ |

Table 3: Temporal label dependencies expressed as integer linear programming constraints. The variables $i, j$ and $k$ range over the corresponding TLINK arguments, and constant $d$ refers to the document-creation-time. $\text{CONTAINS}_{i,j}$ indicates that entity $i$ contains entity $j$.



1. Document-level learning in contrast to pair-wise entity-relation learning.
2. Joint learning of DCTR and TLINKS.
3. Integrating temporal label constraints.
4. Using global structured features.

We will discuss our results in Section 4.4. But first, we describe how we evaluate our system, and provide information on our baselines, and the preprocessing and hyper-parameter settings used in the experiments.

## 4.1 Evaluation

We evaluate our method on the clinical notes test set of the THYME corpus (Styler IV et al., 2014), also used in the Clinical TempEval 2016 Shared Task (Bethard et al., 2016). Some statistics about the dataset can be found in Table 4. F-measure is used as evaluation metric. For this we use the evaluation script from the Clinical TempEval 2016 Shared Task. TLINKS are evaluated under the temporal closure (UzZaman and Allen, 2011).

| Section | Documents | TLINKS | EVENTS |
|---|---|---|---|
| Train | 440 | 17.109 | 38.872 |
| Test | 151 | 8.903 | 18.989 |

Table 4: Dataset statistics for the THYME sections we used in our experiments.

## 4.2 Baselines

Our first baseline is a perceptron algorithm, trained for each local task using the same local features as used to compose the joint feature function $\phi_{joint}$ of our structured perceptron. We have two competitive state-of-the-art baselines, one for the DCTR sub-task, and one for the TLINK sub-task. The first baseline is the best performing system of the Clinical TempEval 2016 on the DCTR task (Khalifa et al., 2016). They experiment with a feature rich SVM and a sequential conditional random field (CRF) for the prediction of DCTR and report the – to our knowledge – highest performance on the DCTR task. The competitive TLINK baseline is the latest version of the cTAKES Temporal system (Lin et al., 2016b; Lin et al., 2016a). They employ two SVMS to predict TLINKS, one for TLINKS between events, and one for TLINKS between events and temporal expressions and recently improved their system by generating extra training data using extracted UMLS concepts. They report the – to our knowledge – highest performance on CONTAINS TLINKS in the THYME corpus.

## 4.3 Hyper-parameters and Preprocessing

In all experiments, we preprocess the text by using a very simple tokenization procedure considering punctuation[1] or newline tokens as individual tokens, and splitting on spaces. For our part-of-speech (POS) features, and dependency parse path features, we rely on the cTAKES POS tagger and cTAKES dependency parser respectively (Savova et al., 2010). After POS tagging and parsing we lowercase the tokens. As mentioned in Section 3.2, we restrict our TLINK candidate generation in two ways. First, both entities should occur in a token window of 30, selected from $\{20, 25, 30, 35, 40\}$ based on development set performance. And second, both entities should occur in the same paragraph (paragraphs are separated

[1] ; :=n'"=+-;:()!?<>%&$ * |[] fg

by two consecutive newlines). Our motivation for not using sentence based candidate generation is that the clinical records contain many ungrammatical phrases, bullet point enumerations, and tables that may result in missing cross-sentence relation instances (Leeuwenberg and Moens, 2016). In all experiments, we train the normal perceptron for 8 iterations, and the structured perceptron for 32 iterations, both selected from $\{1, 2, 4, 8, 16, 32, 64\}$ based on best performance on the development set. The baseline perceptron is also used for the initialization of the structured perceptron. Moreover, we apply the transitive closure of CONTAINS, and BEFORE on the training data.

### 4.4 Results

Our experimental results on the THYME test set are reported in Table 5. In the table, the abbreviation SP refers to the structured perceptron model described in Section 3 but without temporal label constraints or global features, i.e. the joint document-level unconstrained structured perceptron, using local initialization, and loss-augmented negative sub-sampling. We compare this model with a number of modi ed versions to explore the effect of the modi cations.

#### 4.4.1 Document-Level Learning

When we compare the local perceptron baseline with any of the document-level models (any SP variation), we can clearly see that learning the relations at a document-level improves our model signi cantly[2] ($P < 0.0001$ for both DCTR and TLINKS). Furthermore, when comparing loss-augmented sub-sampling (SP) with random sub-sampling of negative TLINKS ($SP_{random\ sub\text{-}sampling}$) it can be seen that a good selection of negative training instances is very important for learning a good model (again $P < 0.0001$), and resulted in our model to improve the state-of-the-art by 1.4 on the CONTAINS TLINK task[3].

#### 4.4.2 Jointly Learning DCTR and TLINKS

When comparing the disjoint model ($SP_{disjoint}$) with our joint model (SP) it can be noticed that joint prediction gives only a very small improvement ($P = 0.0768$ for TLINKS, and $P = 0.0451$ for DCTR). However, joint learning on a document level provides the exibility to formulate constraints connecting the labels of both tasks, such as the last four constraints in Table 3, resulting in a more consistent labeling over both tasks. Similarly, in the joint learning setting, we can de ne global features that connect both tasks (like $_{drtl}$).

#### 4.4.3 Integrating Temporal Constraints

We experimented with integrating label constraints in two ways (1) both during training and prediction ($SP^{pc}$ + C ), or (2) only during prediction ($SP^{uc}$ + C ). In general it can be noticed that in our experiments using the temporal label constraints from Table 3 slightly increases DCTR performance, but slightly decreases TLINK performance. A reason for this can be that the model generally gives better predictions for DCTR, that might result in providing a better alternative to a constraint violating solution. A difference in consistency of the annotation between both tasks could also be a reason. Furthermore, we can see that integrating the constraints both during training and prediction gives slightly lower performance compared only integrating them during prediction.

#### 4.4.4 Using Global Structured Features

We have two types of features, $_{sdr}$, which is only based on DCTR labels, and $_{drtl}$, which is based on a combination of DCTR and TLINK labels. When we add $_{sdr}$ to our model, the overall F-measure on the DCTR task improves with 1.3 points ($P < 0.0001$), improving the state-of-the-art by 0.3 points. A reason for this can be the sequential dependency of DCTR labels, also exploited by (Khalifa et al., 2016), using the sequential CRF. The second global feature, $_{drtl}$, in fact models the same type of dependencies as the last four constraints in Table 3, relating the TLINK relations with the DCTR labels of each TLINK argument, however as a soft dependency and not as a hard constraint. In our experiments, this feature did not improve either of the two sub-tasks. It appears that training with cross-task constraints, or global cross-task features is not trivial, and further research is needed on how to exploit these cross-task dependencies also during training. We assume that the lower-than-expected scores when modeling cross-task dependencies may be related to sub-sampling the negative TLINK training instances.

[2] Signi cance is based on a document-level paired t-test.

[3] Only CONTAINS is generally reported for the THYME corpus, as the other TLINKS are less frequent, and the inter-annotator agreement for them is very low. We included them just for completeness in our experiments.

| System | $F^{DCTR}_{BEFORE}$ | $F^{DCTR}_{AFTER}$ | $F^{DCTR}_{OVERLAP}$ | $F^{DCTR}_{BEFORE/OVERLAP}$ | $F^{DCTR}_{ALL}$ | $F^{TLINK}_{CONTAINS}$ | $F^{TLINK}_{BEFORE}$ | $F^{TLINK}_{OVERLAP}$ | $F^{TLINK}_{BEGINS-ON}$ | $F^{TLINK}_{ENDS-ON}$ | $F^{TLINK}_{ALL}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Baseline: perceptron | 0.776 | 0.744 | 0.769 | 0.528 | 0.759 | 0.456 | 0.147 | 0.073 | 0.060 | 0.024 | 0.364 |
| (Khalifa et al., 2016) | - | - | - | - | 0.843 | - | - | - | - | - | |
| (Lin et al., 2016b) | - | - | - | - | - | 0.594 | - | - | - | - | - |
| SP | 0.837 | 0.805 | 0.860 | 0.575 | 0.833 | 0.608 | 0.294 | 0.185 | 0.158 | 0.231 | 0.518 |
| $SP_{random\ sub\text{-}sampling}$ | 0.837 | 0.803 | 0.859 | 0.575 | 0.833 | 0.564 | 0.275 | 0.204 | 0.154 | 0.218 | 0.490 |
| $SP_{disjoint}$ | 0.835 | 0.801 | 0.859 | 0.576 | 0.832 | 0.607 | 0.290 | 0.183 | 0.146 | 0.232 | 0.516 |
| $SP^{cc}$ + C | 0.843 | 0.810 | 0.861 | 0.573 | 0.836 | 0.603 | 0.292 | 0.186 | 0.148 | 0.222 | 0.514 |
| $SP^{uc}$ + C | 0.843 | 0.814 | 0.861 | 0.574 | 0.837 | 0.606 | 0.291 | 0.184 | 0.157 | 0.236 | 0.516 |
| SP + $_{sdr}$ | 0.856 | 0.830 | 0.867 | 0.569 | 0.846 | 0.608 | 0.291 | 0.182 | 0.159 | 0.222 | 0.518 |
| SP + $_{drtl}$ | 0.838 | 0.811 | 0.855 | 0.564 | 0.831 | 0.605 | 0.286 | 0.176 | 0.147 | 0.217 | 0.514 |

Table 5: Results on the THYME test set. SP refers to our structured perceptron model, without constraints or global features, using local initialization and loss-augmented negative sub-sampling. C refers to using all constraints. Superscript CC and UC refer to using constraints at training and prediction time, or only at prediction time respectively.



## 5 Conclusions

In this work, we proposed a structured perceptron model for learning temporal relations between events and the document-creation time (DCTR), and between temporal entities in the text (TLINKS) in clinical records. Our model efficiently learns and predicts at a document level, exploiting loss-augmented negative sub-sampling, and uses global features allowing it to exploit relations between local output labels. For construction of a consistent output labeling, needed for time-line construction, we formulated a number of constraints, including those from (Chambers et al., 2007; Do et al., 2012), and assessed them during inference. Our best system outperforms the state-of-the-art of both the CONTAINS TLINK task, and the DCTR task. Our code for this work is available at https://github.com/tuur/SPTempRels.

## Acknowledgment

The authors would like to thank the reviewers for their constructive comments which helped us to improve the paper. Also, we would like to thank the Mayo Clinic for permission to use the THYME corpus. This work was funded by the KU Leuven C22/15/16 project "MAchine Reading of patient recordS (MARS)", and by the IWT-SBO 150056 project "ACquiring CrUcial Medical information Using LAnguage TEchnology" (ACCUMULATE).

## References

[Bethard et al.2014] Steven Bethard, Leon Derczynski, James Pustejovsky, and Marc Verhagen. 2014. Clinical tempeval. arXiv preprint arXiv:1403.4928.

[Bethard et al.2015] Steven Bethard, Leon Derczynski, Guergana Savova, Guergana Savova, James Pustejovsky, and Marc Verhagen. 2015. Semeval-2015 task 6: Clinical tempeval. In Proceedings of SemEval, pages 806–814. Association for Computational Linguistics.

[Bethard et al.2016] Steven Bethard, Guergana Savova, Wei-Te Chen, Leon Derczynski, James Pustejovsky, and Marc Verhagen. 2016. Semeval-2016 task 12: Clinical tempeval. pages 1052–1062. Association for Computational Linguistics.

[Chambers and Jurafsky2008] Nathanael Chambers and Dan Jurafsky. 2008. Jointly combining implicit constraints improves temporal ordering. In Proceedings of EMNLP, pages 698–706. Association for Computational Linguistics.

[Chambers et al.2007] Nathanael Chambers, Shan Wang, and Dan Jurafsky. 2007. Classifying temporal relations between events. In Proceedings of ACL, pages 173–176. Association for Computational Linguistics.

[Collins2002] Michael Collins. 2002. Ranking algorithms for named-entity extraction: Boosting and the voted perceptron. In Proceedings of ACL, pages 489–496. Association for Computational Linguistics.

[Combi and Shahar1997] Carlo Combi and Yuval Shahar. 1997. Temporal reasoning and temporal data maintenance in medicine: issues and challenges. Computers in Biology and Medicine, 27(5):353–368.

[Daume2006] Harold Charles Daume. 2006. Practical Structured Learning Techniques for Natural Language Processing. ProQuest.

[Do et al.2012] Quang Xuan Do, Wei Lu, and Dan Roth. 2012. Joint inference for event timeline construction. In Proceedings of EMNLP, pages 677–687. Association for Computational Linguistics.

[Freund and Schapire1999] Yoav Freund and Robert E Schapire. 1999. Large margin classification using the perceptron algorithm. *Machine Learning*, 37(3):277–296.

[Gurobi Optimization2015] Inc. Gurobi Optimization. 2015. Gurobi optimizer reference manual.

[Khalifa et al.2016] Abdulrahman Khalifa, Sumithra Velupillai, and Stephane Meystre. 2016. Utahbmi at SemEval-2016 task 12: Extracting temporal information from clinical text. *Proceedings of SemEval*, pages 1256–1262.

[Kordjamshidi et al.2015] Parisa Kordjamshidi, Dan Roth, and Marie-Francine Moens. 2015. Structured learning for spatial information extraction from biomedical text: bacteria biotopes. *BMC Bioinformatics*, 16(1):1.

[Lee et al.2016] Hee-Jin Lee, Yaoyun Zhang, Jun Xu, Sungrim Moon, Jingqi Wang, Yonghui Wu, and Hua Xu. 2016. UTHealth at SemEval-2016 task 12: an end-to-end system for temporal information extraction from clinical notes. *Proceedings of SemEval*, pages 1292–1297.

[Leeuwenberg and Moens2016] Artuur Leeuwenberg and Marie-Francine Moens. 2016. KULeuven-LIIR at SemEval 2016 task 12: Detecting narrative containment in clinical records. *Proceedings of SemEval*, pages 1280–1285.

[Lin et al.2016a] Chen Lin, Dmitriy Dligach, Timothy A Miller, Steven Bethard, and Guergana K Savova. 2016a. Multilayered temporal modeling for the clinical domain. *Journal of the American Medical Informatics Association*, 23(2):387–395.

[Lin et al.2016b] Chen Lin, Timothy Miller, Dmitriy Dligach, Steven Bethard, and Guergana Savova. 2016b. Improving temporal relation extraction with training instance augmentation. *Proceedings of ACL*, page 108.

[Mani et al.2006] Inderjeet Mani, Marc Verhagen, Ben Wellner, Chong Min Lee, and James Pustejovsky. 2006. Machine learning of temporal relations. In *Proceedings of COLING–ACL*, pages 753–760. Association for Computational Linguistics.

[Mojica and Ng2016] Luis Gerardo Mojica and Vincent Ng. 2016. Markov logic networks for text mining: A qualitative and empirical comparison with integer linear programming. In *Proceedings of LREC*, pages 4388–4395.

[Onisko et al.2015] Agnieszka Onisko, Allan Tucker, and Marek J. Druzdzel, 2015. Prediction and Prognosis of Health and Disease, pages 181–188. Springer International Publishing, Cham.

[Pustejovsky and Stubbs2011] James Pustejovsky and Amber Stubbs. 2011. Increasing informativeness in temporal annotation. In *Proceedings of the 5th Linguistic Annotation Workshop*, pages 152–160. Association for Computational Linguistics.

[Pustejovsky et al.2003] James Pustejovsky, Patrick Hanks, Roser Sauri, Andrew See, Robert Gaizauskas, Andrea Setzer, Dragomir Radev, Beth Sundheim, David Day, Lisa Ferro, et al. 2003. The TimeBank corpus. In *Corpus Linguistics*, volume 2003, page 40.

[Savova et al.2010] Guergana K Savova, James J Masanz, Philip V Ogren, Jiaping Zheng, Sunghwan Sohn, Karin C Kipper-Schuler, and Christopher G Chute. 2010. Mayo clinical text analysis and knowledge extraction system (cTAKES): architecture, component evaluation and applications. *Journal of the American Medical Informatics Association*, 17(5):507–513.

[Stacey and McGregor2007] Michael Stacey and Carolyn McGregor. 2007. Temporal abstraction in intelligent clinical data analysis: A survey. *Artificial Intelligence in Medicine*, 39(1):1–24.

[Styler IV et al.2014] William F Styler IV, Steven Bethard, Sean Finan, Martha Palmer, Sameer Pradhan, Piet C de Groen, Brad Erickson, Timothy Miller, Chen Lin, Guergana Savova, et al. 2014. Temporal annotation in the clinical domain. *Transactions of the Association for Computational Linguistics*, 2:143–154.

[Sun et al.2013] Weiyi Sun, Anna Rumshisky, and Ozlem Uzuner. 2013. Evaluating temporal relations in clinical text: 2012 i2b2 challenge. *Journal of the American Medical Informatics Association*, 20(5):806–813.

[UzZaman and Allen2011] Naushad UzZaman and James F Allen. 2011. Temporal evaluation. In *Proceedings of ACL–HLT*, pages 351–356. Association for Computational Linguistics.

[Weiss et al.2015] David Weiss, Chris Alberti, Michael Collins, and Slav Petrov. 2015. Structured training for neural network transition-based parsing. *arXiv preprint arXiv:1506.06158*.

[Yoshikawa et al.2009] Katsumasa Yoshikawa, Sebastian Riedel, Masayuki Asahara, and Yuji Matsumoto. 2009. Jointly identifying temporal relations with Markov logic. In *Proceedings of ACL-IJCNLP*, pages 405–413. Association for Computational Linguistics.