KU LEUVEN

ARENBERG DOCTORAL SCHOOL
Faculty of Engineering Science

From Text to Time: Machine Learning Approaches to Temporal Information Extraction from Text

Artuur Leeuwenberg

Supervisor:
Prof. dr. Marie-Francine Moens

Dissertation presented in partial fulfillment of the requirements for the degree of Doctor of Engineering Science (PhD): Computer Science

October 2019

From Text to Time: Machine Learning Approaches to Temporal Information Extraction from Text

Artuur LEEUWENBERG

Examination committee:
Prof. dr. Adhemar Bultheel, chair
Prof. dr. Marie-Francine Moens, supervisor
Prof. dr. Jesse Davis
Prof. dr. ir. Jan Aerts
Prof. dr. Walter Daelemans
(University of Antwerp)
Dr. Angus Roberts
(King's College London)

Dissertation presented in partial fulfillment of the requirements for the degree of Doctor of Engineering Science (PhD): Computer Science

October 2019

Acknowledgements

The four years of my PhD research have been a truly amazing period of my life, which I look back to with much joy. This would have been very different if it wasn't for a number of people who I would like to thank here.

I start with my promotor Sien Moens, who gave me the opportunity to start this PhD. This thesis would not have been possible without the many interesting discussions we had, and without the advice, support, encouragement and freedom you gave me throughout these four years. I could not have wished for a better supervisor. Thank you Sien!

Secondly, I would like to express my gratitude to my assessors and jury members, Prof. Jan Aerts, and Prof. Jesse Davis, who took the time to assess and advice on my research during the course of the PhD. Also, I would like to thank my external jury members Prof. Walter Daelemans and Dr. Angus Roberts for taking the time to review my thesis and travel to Leuven for this. Also thanks to professor Adhemar Bultheel for serving as the chairman of the committee for both of my defences.

I am truly lucky that I could work among such amazing colleagues and friends at LIIR. Thank you all for such a nice time, and the lunches and walks to Alma 3 (and to De Moete ;)). For me, this was a moment to look forward to every day, and I also learned a lot about Flemish, Walloon, Chinese, Vietnamese, Catalan and Indian culture. A special thanks to my office mates Aparna, Quynh, Guillem, Bregt, Golnoosh, and Thierry for the fun helpful conversations and many teaffee breaks. Thank you Geert for your many very funny jokes about the Netherlands, but also your listening ear and the insightful discussions. You are probably the only person who knows the content of

this entire thesis already while never having read it.

My PhD would not have been the same if it wasn't for you Margot. Thank you for your endless love, understanding, encouragement and support during these four years. You are truly the best girlfriend I could wish for!

Finally, I would like to thank my parents, brothers, and friends, who without choice suddenly had to listen for four years to stories about time, language, and computers. Thank you for supporting me in this!

Abstract

Temporal information extraction is and has been a crucial aspect of automatic language understanding. With the increase in digitization of texts like news papers, but also electronic health records, high quality extraction of temporal information about the described events gives rise to many applications, like question answering, summarization, temporal information retrieval, and automatic timeline visualization.

In this dissertation, we investigate and propose different machine learning approaches for temporal information extraction from text. Our five main contributions show how symbolic knowledge about temporal reasoning and statistical neural network based approaches can be used and combined to improve temporal extraction, in terms of prediction quality, and in terms of coverage.

First, we construct a document-level structured learning approach for temporal relation extraction, incorporating hard and soft temporal constraints during training and prediction. We show that the document-level constraints can help to improve the quality of the model's predictions, and enforce the predicted temporal relations to be more consistent, important for timeline construction.

Secondly, we design a neural temporal relation extraction model, and investigate how we can efficiently optimize its word representations using unlabeled textual data. Multi-task learning is used as a way to train the representations on two objectives, the supervised relation extraction objective based on the labeled data, and a skip-gram objective based on raw text.

Our third contribution is a literature survey on how temporal reasoning can be successfully and efficiently integrated into temporal information extraction models. The survey highlights multiple gaps in the literature, and provides interesting avenues for future work.

Building on insights from temporal reasoning frameworks, as our fourth contribution, we investigate the construction of timelines of events from text. A new method for relative timeline construction from graphs of temporal relations is proposed. More importantly, a new paradigm is introduced to extract timelines without the need for intermediate temporal relation extraction.

Our last contribution extends the previous work with the extraction of implicit and uncertain temporal information. An annotation scheme is proposed to annotate absolute timelines that can be queried in a probabilistic way, based on the annotated uncertainty, and a set of clinical records is annotated. Moreover, a model is put forward to extract such timelines from text.

The main conclusion of this dissertation is the importance of good integration of symbolic temporal reasoning, key to capturing rigid temporal structure, into statistical (neural) models, crucial when dealing with the ambiguous nature and vagueness present in language. The contributions in this thesis act as a case study and starting point for future research into this integration.

Beknopte samenvatting

Temporele informatie-extractie is een cruciale component voor de automatische verwerking van taal. Met de toename in digitalisering van teksten zoals nieuwsartikelen maar ook elektronische patiëntendossiers geeft hoge kwaliteit extractie van temporele informatie over de beschreven gebeurtenissen aanleiding tot veel praktische toepassingen: zoals het automatisch beantwoorden van inhoudelijke vragen over teksten, het automatisch samenvatten ervan, zoeken op basis van temporele queries, en de visualisatie van tekstgebaseerde tijdlijnen.

In deze dissertatie worden verschillende methoden van machinaal leren voor de extractie van temporele informatie uit tekst voorgesteld en onderzocht. Onze vijf hoofdcontributies demonstreren hoe symbolische kennis over het redeneren met tijd en statistische neurale netwerken kunnen worden gebruikt en gecombineerd voor het verbeteren van automatische temporele informatie-extractie.

Eerst construeren we een gestructureerde leermethode voor de extractie van temporele relaties op documentniveau. Hierin integreren we harde (logische) en zachte (statistische) temporele restricties tijdens de trainingsfase en de predictie van het model. We laten zien dat deze restricties op documentniveau de kwaliteit van de voorspellingen gedaan door het model verbeteren en dat ze forceren dat de voorspelde temporele relaties consistenter zijn, een belangrijke eigenschap voor tijdlijnconstructie.

Ten tweede ontwerpen we een neuraal model voor temporele relatie-extractie en onderzoeken hoe de woordrepresentaties van dit model efficiënt kunnen worden geoptimaliseerd gebruik makende van ongelabelde tekst. Multi-task learning wordt gebruikt als methode om de woordrepresentaties te trainen op twee objectieven: het

gesuperviseerde relatie-extractie objectief gebaseerd op gelabelde data, en een skip-gram objectief gebaseerd op ongelabelde tekst.

Onze derde contributie is een literatuuronderzoek naar hoe temporeel redeneren succesvol en efficiënt kan worden geïntegreerd in temporele informatie-extractie modellen. Dit literatuuronderzoek brengt verscheidene gebreken in de bestaande literatuur aan het licht en voorziet interessante richtingen voor verder toekomstig onderzoek.

Bouwend op de inzichten uit de verschillende frameworks voor temporeel redeneren onderzoeken we in onze vierde contributie de constructie van tijdlijnen van gebeurtenissen uit de tekst. Een nieuwe methode wordt voorgesteld voor het opstellen van relatieve tijdlijnen uit grafen van temporele relaties. Belangrijker nog is de introductie van een nieuw paradigma voor de extractie van tijdlijnen waarvoor de gebruikelijke tussenstap van temporele relatie-extractie niet langer noodzakelijk is.

Onze laatste contributie bouwt verder op de vorige met als extensie de extractie van impliciete en onzekere temporele informatie. Een nieuw annotatieschema voor het annoteren van absolute tijdlijnen met een probabilistische interpretatie gebaseerd op de geannoteerde onzekerheid wordt beschreven. Vervolgens is dit schema toegepast in de annotatie van een dataset van Amerikaanse klinische rapporten, gebruikt voor de constructie en evaluatie van een extractie-model voor dergelijke probabilistische tijdlijnen.

De hoofdconclusie van deze dissertatie is het belang van goede integratie van symbolische methoden voor temporeel redeneren, aangewezen voor het beschrijven van de sterke structuur van tijd, in statistische (neurale) modellen, cruciaal voor het omgaan met de ambiguïteit en vaagheid in taal. De contributies in deze dissertatie dienen als een case study en startpunt voor toekomstig onderzoek naar deze integratie.

List of Abbreviations

AI artificial intelligence. 1

ANN artificial neural networks. 9

BF best first. 65

CNN convolutional neural networks. 34

CRF conditional random field. 27

DCT document-creation-time. 4

DCT-R event-document-creation-time relations. 4

EE-R event-event relations. 4

ILP integer linear programming. 4

LSTM long short-term memory networks. 13

ML machine learning. 1

MLN Markov logic networks. 20

NLP natural language processing. 1

NRO natural reading order. 65

POS part-of-speech. 27

RC relation classification. 33

RRHC random restart hill climbing. 67

SG skip-gram. 33

SI stacked inference. 66

SLI sieve-level inference. 65

SP structured perceptron. 17

SVM support vector machine. 26

TE-R temporal expression-event relations. 4

TIE temporal information extraction. 1

timex time expression. 4

TLinks temporal links. 4

TR temporal reasoning. 1

Contents

List of Figures

List of Tables

1 Introduction

> Time is what keeps everything from happening at once.
>
> *Ray Cummings*

Time is a central concept of human perception and cognition. In our daily life, we make many decisions based on time. Our awareness of time has blended into our communication, and our language is filled with temporal cues. Two examples of frequent temporal cues in language are verb tense (*has* vs. *had*), and word order (*he laughed and fell* vs. *he fell and laughed*). In this dissertation, we investigate how we can learn computer models to extract the temporal cues that we convey in language, from textual documents. In other words, how can we learn models to construct a timeline of events from a textual document? This research lies within the broad research area of artificial intelligence (AI), and integrates three subareas: (1) natural language processing (NLP), the study of processing of language using computer models, (2) machine learning (ML), the study of building computer models that can learn to perform certain tasks from a set of examples. And (3), temporal reasoning (TR), the study on how to represent and reason with temporal information.

In this dissertation, we focus primarily, but not exclusively, on temporal information extraction (TIE) in the *clinical* domain, using textual patient records for the majority of the experiments. Being able to automatically construct high quality timelines from clinical documents provides many real world applications, and plays an increasingly important role with the increasing digitization of personal health records. Applications include temporal question answering (QA), where doctors can ask temporal questions

about events occurring in a collection of documents (Llorens et al., 2015; Höffner et al., 2017; Meng et al., 2017; Sun et al., 2018; Pampari et al., 2018). For example, "When was the last time that the patient received narcotics?". Another application is the (semi-)automatic selection of patients for clinical trials, which frequently involves temporal constraints in their inclusion and exclusion criteria for participants (Raghavan et al., 2014). For example, "patients with a distant history (greater than 6 months before study entry) of venous thromboembolic disease are eligible". Other applications include the visualization of patient timelines (Jung et al., 2011), multi-document summarization (Barzilay and McKeown, 2005; Ng et al., 2014), and it has important potential for better prediction of treatment effects and early detection of diagnoses (Augusto, 2005; Zhou and Hripcsak, 2007; Choi et al., 2016b,c).

In the remainder of this chapter we will draw the research context in which this work was conducted, describe the problem statement and research questions, summarize the contributions, and provide the outline of the thesis.

1.1 Research Context

The research in this dissertation was pursued in the context of the MARS project: "MAchine Reading of patient RecordS" (KUL-C22/15/16), and the ACCUMULATE project "Acquiring Crucial Medical information Using Language Technology" (IWT-SBO 150056). Both projects focus on the extraction of important clinical information from textual patient records, including temporal information.

1.2 Problem Statement and Contributions

We start this section by introducing Figure 1.1, which shows typical tasks in temporal information extraction, and acts as a backbone for describing the research questions and contributions made in this dissertation.

To build a timeline of events from text automatically, the first step is to extract the events from the text that will be placed on the timeline. This phase is called event extraction (a $\Rightarrow$ b in Figure 1.1). Events are typically described as "situations that *happen* or *occur*" (Pustejovsky et al., 2003b). Models for event extraction fall in the category of sequence-labeling models, which are very frequently used in NLP, and determine for each word in the text, whether it should be assigned the label *event* or not based on the word itself and its context.

Once events have been extracted, time expressions, like dates (*12/03/2006*), times (*9 am*) or durations (*for five hours*), are detected. Time expressions, often abbreviated as

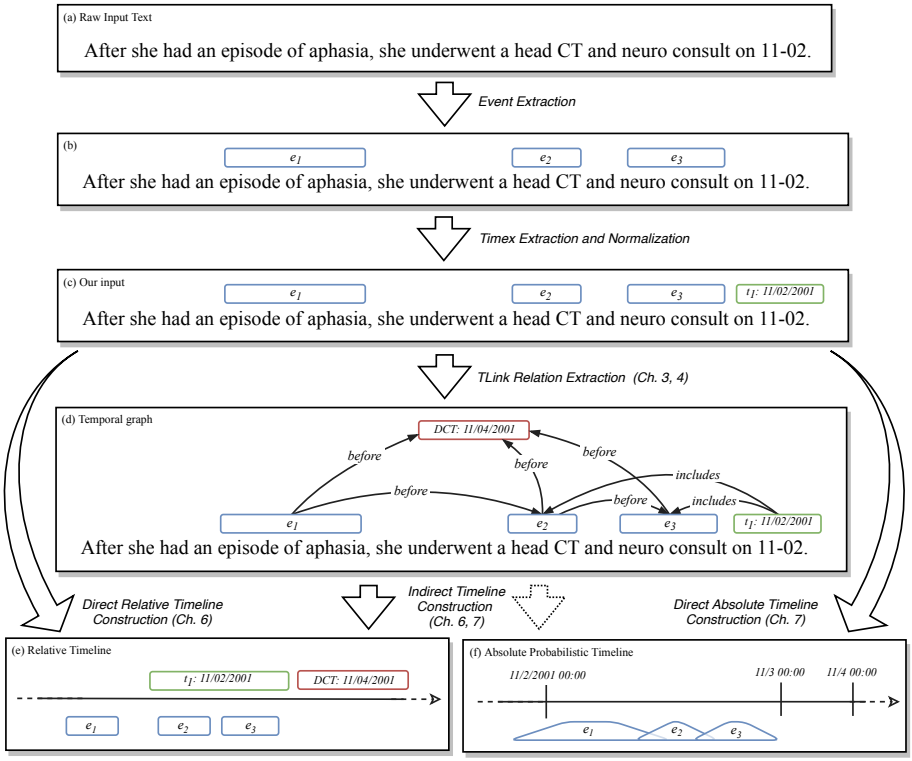

Figure 1.1: Overview of different tasks of temporal information extraction, starting from the extraction and normalization of events and temporal expressions from raw text: (a $\Rightarrow$ b $\Rightarrow$ c). And successively, the construction of an event timeline, the focus of this dissertation: either directly (c $\Rightarrow$ e or c $\Rightarrow$ f), or indirectly, by extracting a graph of temporal relations as an intermediate step: (c $\Rightarrow$ d, followed by d $\Rightarrow$ e or d $\Rightarrow$ f).

timex in the literature and the rest of this dissertation, refer directly to values on the timeline, and function as reference points (or anchors) around which the events can be positioned. Besides the detection of timexes, it is important to know what date or time value is referred to by the time expression, which is not always evident (e.g. for relative cases like *last Sunday*). The process of automatically determining the calendar value for a timex is called *timex normalization*. Figure 1.1 shows an example of timex extraction and normalization (b $\Rightarrow$ c).

Temporal Relation Extraction

The models proposed in this dissertation all start from text in which events and timex are already extracted, and normalized: step (c) in the figure. From this input, the work in this dissertation researches how to further construct the event timeline. The motivation for this starting point is because existing models up to this point already have higher accuracy (around 70-90%), compared to successive stages (around 50-70%).

An important and frequently explored next step towards building timelines is the extraction of temporal relations, or temporal links (TLinks): (1) between the events (EE-R), (2) between the temporal expressions and events (TE-R), and (3) between each event and the document-creation time (DCT-R). The results of the TLink extraction, shown in (d) in Figure 1.1, is a temporal graph with events, timex, and the DCT as nodes (from now on referred to as temporal entities). The graph's edges are the TLinks among them. The types of TLinks are usually a (sub)set of Allen algebra relations (Allen, 1983), which describe the relative positions that intervals can have with regard to each other (e.g. *before*, *during*, *simultaneously*). Many existing TLink extraction models iteratively predict a TLink between each pair of temporal entities, based on the textual context. A problem with this approach is that TLinks are predicted independently of each other. However, TLinks are not independent of each other. In Figure 1.1d: The facts that (1) e_1 lies before e_2 and (2) e_2 lies before the DCT restricts the possibilities for the relation between e_1 and the DCT: it must now be before the DCT to stay temporally consistent. This dependence between TLinks has two sides: on the one-hand we can infer relations from existing relations, but on the other hand, if we predict a wrong relation this may propagate to the rest of the temporal graph. This brings us to our first research question:

Q 1. How can we exploit the dependencies between temporal relations to improve temporal relation extraction models?

C 1. To answer this question we propose and evaluate a structured perceptron model for TLink extraction, enhanced with integer linear programming (ILP) to model temporal inference. The model jointly predicts EE-R and TE-R, DCT-R, constituting a new state of the art on the clinical THYME dataset at the time.

C 2. Evaluating this model provides valuable insights on when and how to make temporal inferences over the candidate relations both during training and prediction and gives an in-depth assessment of the use of constraints and global features.

Training machine learning models for TLink extraction requires a sufficient number of texts with the correct temporal graphs as training examples. The manual annotation of these graphs requires domain specialists, especially for clinical texts. Additionally, for a text with n temporal entities, there are $O(n^2)$ potential relations, making this task very time consuming, and costly. As a consequence, the existing datasets are relatively small, with sizes ranging between 36 and 500 annotated documents. However, raw unannotated text is widely available (even for the clinical domain). Recent developments in NLP provide methods to exploit large collections of raw text to construct word representations that capture a degree of syntactic and semantic similarity. We believe parts of the information that is captured by these representation may be useful to better extract TLinks (e.g. if one event is in future tense and another in past tense, the first happened before the second). This brings us to our second research question:

Q 2. How can we use raw text, in addition to our annotated texts, to improve word representations of TLink extraction models?

C 3. To efficiently exploit the raw text, we proposed a TLink extraction model which optimizes its word-level representations jointly on the TLink extraction task, using annotated data, and on an auxiliary context prediction task, using unannotated data. We show that this multi-task training setting results in better representations for classification, and constitutes a new state of the art for temporal relation extraction on the THYME dataset even without dedicated clinical preprocessing.

Timeline Construction

Up til this point, the discussed research questions involved the extraction of temporal relations, or temporal graphs. However, the final aim of temporal information extraction is to construct a timeline of events from the text. There is a strong relation between temporal graphs and the start and end points of events on a timeline. Formal theories of temporal reasoning can connect these two perspectives, providing powerful and flexible tools to handle temporal information. To obtain further insight into how temporal reasoning can be integrated successfully in models for timeline construction, we formulated the following literature-based research question:

Q 3. How has temporal reasoning been successfully used in the research field of temporal information extraction? And what are promising future directions for further integration?

C 4. To address this literature-based research questions, we conducted a survey on the various ways in which TR has been exploited in TIE models over the past three decades: in annotation, preprocessing, training, prediction, and evaluation. The survey includes a clear explanation of the theory on TR for TIE required to comprehend the latest state-of-the-art TIE models. Moreover, it provides a structured overview of the various ways in which TR has been exploited in TIE models over the past three decades: in annotation, preprocessing, training, prediction, and evaluation. Finally a distillation of the most important conclusions and directions for future work are included.

Through temporal reasoning we can convert temporal relations between events to statements about points on a timeline, and construct a timeline from automatically extracted TLinks. Irrespective of the quality of TLink extraction models, for long documents with many temporal entities the step of extracting TLinks can be computationally very complex because of: (1) the squared relation between the number of entities and the number of TLinks, and (2) the fact that this large number of TLinks needs to be consistent in order to construct timelines from them. Because of these disadvantages, we investigate if it is possible to directly position events on the timeline in the correct order, without first extracting TLinks. As data annotation is expensive, ideally, we would like to do this using existing datasets. This brings us to research question three:

Q 4. How can we train models from annotated temporal graphs, that directly predict timeline positions for events, without extracting TLinks as an intermediate step?

C 5. To answer this question, we developed a new method to construct a relative time-line from a temporal graph (called TL2RTL: d $\Rightarrow$ e in Figure 1.1).

C 6. More importantly, we proposed two new models that, for the first time, directly predict (relative) timelines - in linear complexity - without an intermediate TLink extraction stage (S-TLM & C-TLM: c $\Rightarrow$ e in Figure 1.1).

C 7. To train the proposed models, we constructed three new loss functions based on the mapping between Allen's interval algebra and the end-point algebra to train time-line models from TimeML-style temporal graphs.

A major aspect of temporal information extraction which has not yet been discussed up to this point is temporal under-specification. Temporal under-specification indicates that only partial information about event timing is available, and that pieces of information

possibly have different levels of certainty. For example, in Figure 1.1, we do not know if the episode of aphasia started on 11-02 or already before that date. Temporal information that is explicitly mentioned through temporal expressions in the text is generally considered more certain. This is what is annotated in most existing datasets, and hence also extracted by the models trained on such data (including the models in this thesis up to this point). Implicit information, often more uncertain in nature, is less frequently annotated. However, uncertain information can still be crucial to make a realistic more informative timeline. For example, although not explicitly mentioned in the text, and not captured by the temporal graph, it is most likely that the episode of aphasia happened fairly shortly before the CT scan, and not years in the past. To address dealing with uncertain temporal information in the construction of timelines from text we formulate our last research question:

Q 5. How can we annotate and extract complete absolute timelines, that capture implicit and uncertain temporal information?

C 8. To include implicit information in our data, we propose a novel annotation scheme for absolute timeline annotation, which deals with implicit information, and under-specification by using absolute uncertainty bounds.

C 9. Using this scheme, we annotated a corpus of English patient records and analyzed inter-annotator agreement, and the scheme's relation to the TimeML-style temporal graphs (focusing more on explicit information).

C 10. Finally, using the new annotations, we trained and evaluated a multi-regression model to predict absolute timelines that can be queried in a probabilistic way, based on the temporal uncertainty in the text.

1.3 Outline

The chapters in this dissertation follow the conducted research chronologically.

Chapter 2 starts with introducing the necessary background knowledge on fundamental methods used in the rest of the dissertation.

Chapters 3 and 4 propose models for the task of temporal relation extraction, covering research questions 1, and 2 respectively. Chapter 5 covers a literature survey on the use of temporal reasoning in the research area of temporal information extraction to answer research question 3. Chapter 6 addresses research question 4, and Chapter 7 addresses research question 5. Both discuss the task of timeline construction, the first focusing on relative timelines, and the second focusing on absolute timelines. Figure 1.1 contains references to the chapters in which new methods have been proposed for a certain task.

Chapter 3 introduces a structured perceptron model which incorporates document-level temporal inference during training and prediction. The model is evaluated on a benchmark corpus of American English clinical patient records in the domain of colon cancer (the THYME corpus).

Chapter 4 presents a neural relation extraction model and investigates the joint training of word representations on the temporal relation extraction task and a context prediction task to better exploit unannotated raw textual data. The robustness of the joint training setup is addressed, and the THYME corpus is used to evaluate the approach and to analyze its errors.

In contrast to other chapters, Chapter 5 is a literature-based survey of the different uses of temporal reasoning in the construction of temporal information extraction models. It categorizes existing approaches for integrating temporal reasoning, focusing on efficiency and expressivity, and highlights gaps in the literature that indicate promising areas for future research towards more complete temporal information extraction.

Chapter 6 tackles the task of timeline prediction and offers novel approaches for indirect timeline construction and direct timeline prediction, focusing on relative timelines, which contain the *order* of the start and end points of the events. The chapter makes a comparison of both methods on two English benchmark datasets in the news domain.

Chapter 7 provides a new scheme for absolute timeline annotation, dealing with temporal uncertainty and implicit information. The scheme is used to annotate a corpus of clinical intensive-care-unit reports. The newly created corpus is analyzed and a new model for the prediction of probabilistic absolute timelines is proposed which is trained on the annotations.

The content of the chapters was based on research published in (Chapters 3-6) or submitted to (Chapter 7) well-known peer-reviewed international conferences or journals. In the final chapter, Chapter 8, we conclude the dissertation and summarize the most important findings and discuss limitations and promising future directions.

2 Fundamentals

In this Chapter we introduce four basic methods which are considered background knowledge for the following chapters. We start by giving a short introduction of neural networks, as all models in the thesis can be considered neural models (Chapters 3, 4, 6, 7). Afterwards, we introduce word embeddings, a technique to represent words in neural models. Third, we discuss long short-term memory networks (LSTM), which are a type of neural network that is able to remember information across a sequence of inputs, and is used to encode sequences of words, or sentences. LSTM are used in the proposed models in Chapters 4, 6, and 7. Finally, we discuss integer linear programming (ILP), a method to formulate and solve certain constrained optimization problems, which is used in the context of model prediction in Chapter 3, and discussed further in Chapter 5.

2.1 Neural Networks

Artificial neural networks (ANN) are a subset of machine learning models that consist of interconnected basic components, called artificial neurons, which are loosely inspired by biological neurons in the brain.

A Single Neuron

Each neuron can take a number of weighted inputs $X = x_0, \ldots, x_N$, with corresponding weights $W = w_0, \ldots, w_N$, an optional bias term β, and an activation

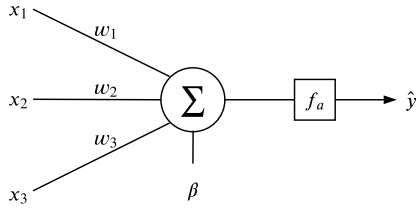

Figure 2.1: A single artificial neuron, with three inputs x_1, x_2, x_3, bias term β, activation function f_a and a predicted output value $\hat{y}$.

function f_a. Together these characterize a forward prediction function $F(X|W, \beta)$ from inputs X to output $\hat{y}$. An example of a single neuron with three inputs is shown in Figure 2.1. The forward pass for a single neuron takes the weighted sum of the inputs, adds the bias term, and applies the activation function to calculate the predicted output. This calculation is given by Equation 2.1.

$$F(X|W, \beta) = f_a(\sum_{i}^{N} w_i x_i + \beta) \tag{2.1}$$

The activation function determines the type of relation between the inputs and the output. There are different options for choosing activation function f_a. A classical option is the use of the logistic sigmoid function given by Equation 2.2, which projects values from $[-\infty, \infty]$ to the unit interval [0,1], and is sometimes used to output probabilities for binary classification problems.

$$\textit{sigmoid}(x) = \frac{1}{1 + e^{-x}} \tag{2.2}$$

Beside the sigmoid function, in this dissertation (Chapters 6 and 7) we also use the ReLU activation function (Nair and Hinton, 2010), and softplus activation function (Dugas et al., 2001), given by Equation 2.3 and 2.4 respectively. Both of these activation functions project from $[-\infty, \infty]$ to $[0, \infty]$ and are used to enforce that the output of networks is positive.

$$\textit{ReLU}(x) = \max(x, 0) \tag{2.3}$$

$$\textit{softplus}(x) = \ln(1 + e^{x}) \tag{2.4}$$

Multilayered Perceptrons

If we stack multiple neurons on top of each other from the same input, we call this a perceptron. A single network where multiple perceptrons are chained to each other by

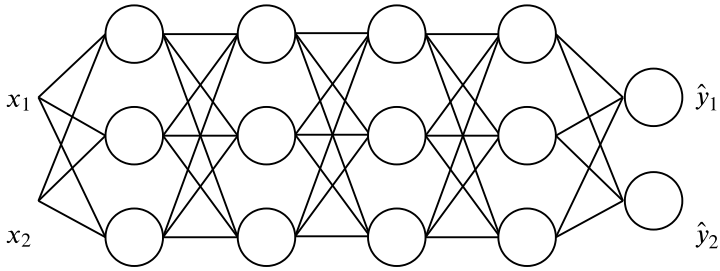

Figure 2.2: An example of a multilayered perceptron, consisting of fours layers, each with three nodes. Bias terms and activation functions are not shown.

connecting the output of each perceptron to the input of the next is called a multilayered perceptron (MLP). Figure 2.2 shows a MLP with five layers. All but the last layer contain three neurons each. The last 2-neuron layer is the output layer. The (four) layers in-between the input and last layer are called hidden layers. If a network has more than one hidden layer, it is called "deep". Calculating the forward pass $F(X)$ for such networks involves many matrix multiplications (one for each layer), which can be efficiently parallelized on Graphical Processing Units (GPUs), greatly reducing computation time. In the experiments in this dissertations we used GPUs to accelerate our calculations.

Gradient-Based Training

Training a network corresponds to finding values for the network's weights and biases (collectively called the network's parameters θ) such that its predictions lie as closely as possible to the desired output. To measure the quality of a prediction $\hat{y}$, compared to the desired output y the amount of error is calculated using a loss function $L(y, \hat{y})$. The goal of training is to minimize loss L on the training data.

If the loss function L is differentiable with regard to the network's parameters θ, which is the case for the neural models used in this dissertation[1], we can calculate how much each parameter $w_i \in \theta$ contributes to the total loss by calculating the parameter's gradient with regard to the loss, i.e., $\frac{\partial L}{\partial w_i}$. Based on its gradient, each parameter is updated with a certain learning rate. In this dissertation we used Adam (Kingma and Ba, 2014) to determine the learning rates at training time.

[1]ReLU is not differentiable at $x = 0$, but this case is handled separately by setting its gradient to 0.

2.2 Word Representations

Informative representation of words in natural language processing models is crucial, as they are the input to the model, and words are often considered the basic components of meaning, carrying important information.

One-Hot Representations

A classical method to represent words as vectors, which can serve as input to machine learning models, is to use one-hot representations. For a vocabulary V of size $|V|$, each word is assigned an index in the vocabulary. The word vector w_i for the word at index i in the vocabulary is defined as a vector of size $|V|$, filled completely with zeros, except at position i, where the value is 1. An advantage of this way of representing words is that each word can be clearly discriminated based on its vector. Typical downsides of this approach are that the word vectors are usually very large (of vocabulary-size), and that no background knowledge about the words is captured in their representation. In Chapter 3 we used one-hot representation to represent words, and other features.

Word Embeddings

An alternative approach to one-hot vectors are word embeddings, which are typically used to induce background knowledge into the word representations. In contrast to one-hot encoded word vectors, word embeddings are dense (containing usually no zeros), and are of low dimensionality (around 10-1000 dimensions), compared to one-hot representations. Word embeddings project each word to a vector in a space $\mathbb{R}^N$, where vector dimension N is a hyperparameter. All word vectors are saved in a matrix of size $|V|$, called the embedding matrix. The values of this matrix are parameters that have to be learned (in contrast to one-hot vectors, where no learning is required). A popular model to learn these weights is the skip-gram model (Mikolov et al., 2013), where the parameters are trained on a word context prediction task, ensuring that words that have similar contexts will get similar vectors. The underlying hypothesis is that words that occur in the same contexts tend to have similar meanings (Harris, 1954). Relying on this hypothesis words with similar meanings get similar vectors. The upside of word embeddings can also be its downside, as the quality of the representations depends fully on the values of the learned vectors. We use different variants of word embeddings in chapters 4, 6, and 7.

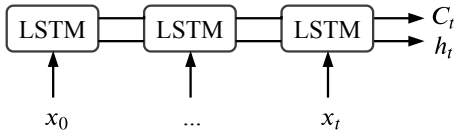

Figure 2.3: An example of a single LSTM processing a sequence of t time steps.

2.3 Long Short-Term Memory Networks

Text can be considered sequential data. Namely, as a sequence of words (or characters). Specialized neural networks have been developed to process sequences, called recurrent neural networks (RNN). In this section we introduce Long Short-Term Memory Networks (LSTM), which have shown to perform well in the literature, and have been used in the research conducted in this dissertation (Chapters 4, 6, and 7).

We read language (mostly) in sequence, remembering the important parts of what we have read thus far. The LSTM architecture (Hochreiter and Schmidhuber, 1997) aims to mimic this behaviour of selectively remembering important information about items in the processed sequence. An example of an LSTM processing up to time step t is shown in Figure 2.3. The memory of a single LSTM cell at each time step t of the sequence is represented by a state vector C_t, and its output by an output vector h_t. Each cell has different gating mechanisms to decide what information from the cell state to forget (forget gate), what to include in the memory from the current input (input gate), and what information to output at the current time step (output gate).

Calculations for the forget gate activations f_t, input gate activations i_t, output gate activations o_t, candidate cell $\tilde{C}_t$, updated cell state C_t, and updated output h_t are given by Equations 2.5-2.10 respectively, where $W_f, W_i, W_o, W_{\tilde{C}}$ and $b_f, b_i, b_o, b_{\tilde{C}}$ are their corresponding weight vectors and bias terms.

$$f_t = \textit{sigmoid}([W_f \cdot [h_{t-1}, x_t]] + b_f) \tag{2.5}$$

$$i_t = \textit{sigmoid}([W_i \cdot [h_{t-1}, x_t]] + b_i) \tag{2.6}$$

$$o_t = \textit{sigmoid}([W_o \cdot [h_{t-1}, x_t]] + b_o) \tag{2.7}$$

$$\tilde{C}_t = tanh(W_{\tilde{C}} \cdot [h_{t-1}, x_t] + b_{\tilde{C}}) \tag{2.8}$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C} \tag{2.9}$$

$$h_t = o_t * tanh(C_t) \tag{2.10}$$

We discuss the equations one by one. First, $\tilde{C}_t$, shown in Equation 2.8, calculates

a newly proposed cell state, conditioned on the previous state's output h_{t-1} and the current input x_t. To construct the new cell state C_t, shown in Equation 2.9, the previous cell state C_{t-1} and the newly proposed cell state $\tilde{C}$ are combined. What information should be kept from the previous state C_{t-1} is weighted per dimension by the output of the forget gate, which assigns a value between 0 and 1 to each state dimension (by means of its sigmoid activation in Equation 2.5). Similarly, the input gate i_t weights each dimension of the newly proposed state vector $\tilde{C}$ to determine what information should be kept. Given the newly calculated memory state C_t, an intermediate output h_t is calculated, where o_t is a gate to determine which information is important to output at this point in time. As calculation of the outputs h_t, and cell state C_t are differentiable, we can use standard gradient-based methods to learn parameter values for the LSTM.

Bidirectional LSTM

An LSTM can only use the previous time steps to construct the output of the current time step. This makes the processing of each word in a sequence oriented towards the left context. However, the right context may be equally important. For this reason, bi-directional LSTM are used (Bi-LSTM), which simply use two LSTMs, one processing the sequence from left to right, and one from right to left. At each time step, the outputs of both LSTMs are concatenated to obtain the overall output vector for that time step. In this dissertation Bi-LSTMs are used to encode word sequences in Chapters 6, and 7.

2.4 Integer Linear Programming

Integer linear programming (ILP) is a powerful mathematical modeling tool. Some computational problems can be modeled as integer linear programs, a class of optimization problems. The benefit of doing so is that, although solving ILPs is NP-complete (Karp, 1972), there exist optimization methods to solve them quite efficiently (Gurobi Optimization, 2015).

An ILP consists of a number of **integer** variables $x \in X$, a number of constants $a \in A$, an objective function O, and a set of constraints between the variables and constants. Solving the ILP corresponds to finding integer values for X such that O is maximal, without violation of the constraints. The objective and the constraints should be **linear** in the variables (meaning addition and subtraction of variables is accepted, but multiplication between variables is not). If these conditions are met, the problem can be relaxed to a linear programming problem (where variables do not necessarily have to be integers), and optimization can make use of the cutting planes method,

which exploits the defined constraints to skip large sections of the possible values for X, without skipping the exact optimal integer solution.

Most often in natural language processing a variant of ILP is used where all variables are constrained to be either 1 or 0. The value of each binary decision variable is used to decide on the assignment of a certain output label. In a machine learning context this can be used to place constraints on the prediction of certain labels, or label combinations. This technique is used in the next chapter (Chapter 3), where we investigate how we can incorporate temporal reasoning, in the form of constraints, into machine learning models, and force the prediction of consistent temporal graphs.

3

Structured Learning for Temporal Relation Extraction from Clinical Records

This chapter was previously published as:

Artuur Leeuwenberg and Marie-Francine Moens. 2017. Structured Learning for Temporal Relation Extraction from Clinical Records. In *Proceedings of the European Chapter of the Association for Computational Linguistics (EACL)*, pp. 1150-1158, Valencia, Spain. ACL.

In this chapter we investigate how we can exploit the dependencies between temporal relations to improve temporal relation extraction models. We introduce a feature-based structured perceptron (SP) model, which models both hard and soft temporal constraints during training and prediction. The constraints can provide an informed reduction of the output space, based on background knowledge about temporal reasoning, potentially improving accuracy. Also, they can ensure that the predicted temporal graphs are more consistent, making them more suitable for further timeline construction. To ensure the efficiency of the complex inference, we formulate the prediction as an integer linear program (ILP), for which efficient solvers exist.

We empirically evaluated different aspects of our model using a benchmark dataset of clinical patient records, and our best setting obtained an improvement over the state-of-the-art temporal relation extraction models at the time for this dataset.

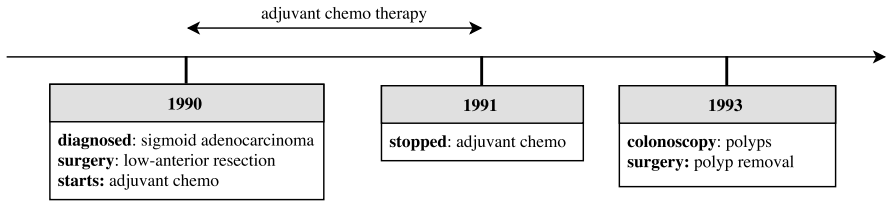

Figure 3.1: Fragment of a (partial) patient timeline.

3.1 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 timeline of the patient's condition (such as in Figure 3.1).

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 chapter, we focus on the last sub-problem, extracting 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 afterwards.

in which we assume that the events *diagnosed* and *surgery*, 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.

This chapter 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 (DCT-R). 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 by using integer linear programming (ILP) constraints with fast solvers, loss augmented sub-sampling, 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 DCT-R task.

3.2 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., 2013b), and more recently the Clinical TempEval Shared Task with two iterations, one in 2015 and one in 2016 (Bethard et al., 2014, 2015, 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 difficult, 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 timeline, 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 (DCT-R), 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 ($E_e \times E_e$), and between events and temporal expressions ($E_e \times E_t$ and $E_t \times E_e$). 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).

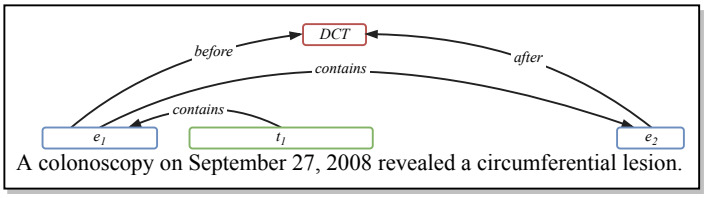

Figure 3.2: Example of inconsistent output labeling. Events are indicated in blue, and temporal expressions in green. Temporal relations are indicated by directed edges.

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 classifier 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 timeline.

An example of inconsistent labeling is given in Figure 3.2. 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, 2008a; 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 Jurafsky (2008a) and Do et al. (2012) modeled label dependencies when predicting TimeBank TLINKS (Pustejovsky et al., 2003b). 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 DCT-R 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 chapter, we aim to jointly predict TLINKS and DCT-R in a structured learning model with inference during training and prediction, to assess inference with temporal constraints of Chambers and Jurafsky (2008a) and 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.

3.3 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 λ 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 Φ, describe the prediction procedure of the model, and describe how we train the model (i.e., learn a good λ).

3.3.1 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 3.1.

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). Φ_{joint} is defined in Equation 3.1, where $C_{tl}(X)$ and $C_{dr}(X)$ are candidate generation functions for the TLINK sub-task and DCT-R sub-task, respectively (further explained in Section 3.3.2).

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

Table 3.1: Features of the local feature functions of each sub-task, ϕ_{tl} for TLINKS, and ϕ_{dr} for DCT-R.

$$\Phi_{joint}(X,Y) \quad = \quad \sum_{x \in C_{dr}(X)} \sum_{l \in L_{dr}} \phi_{dr}(x,l) \;\; \oplus \;\; \sum_{x \in C_{tl}(X)} \sum_{l \in L_{tl}} \phi_{tl}(x,l) \quad (3.1)$$

3.3.2 Local Candidate Generation

For each document X, we create local candidates for both sub-tasks. In this chapter, we assume that event (E_e) and temporal expression (E_t) annotations are provided in the input. The DCT-R-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., $E_e(X) \cup E_t(X) \times E_e(X)$. In our experiments we usually restrict the number of candidates generated by C_{tl} to gain training and prediction speed (without significant loss in performance). This is explained further in Section 3.4.3.

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 specified in Table 3.2. Global features are defined by a feature function $\Phi_{global}(X,Y) \rightarrow \mathbb{R}^r$ and have their corresponding weights in weight vector λ. When using global features Φ_{global} is concatenated with the joint feature function Φ_{joint} to form the final feature function Φ, as show in Equation 3.2.

$$\Phi(X,Y) = \Phi_{joint}(X,Y) \oplus \Phi_{global}(X,Y) \quad (3.2)$$

When not using global features, we use only the joint features, shown in Equation 3.3.

$$\Phi(X,Y) = \Phi_{joint}(X,Y) \quad (3.3)$$

Feature	Description
Φ_{sdr}	Bigram and trigram counts of subsequent DCT-R-labels in the document
Φ_{drtl}	Counts of DCT-R-label pairs of the entities of each TLINK

Table 3.2: Global (document-level) features.

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 defined as the dot product between the learned weight vector λ and the outcome of the joint feature function $\Phi(X, Y)$, as shown in Equation 3.4.

$$S(X, Y) = \lambda\Phi(X, Y) \tag{3.4}$$

The prediction problem for an input document X is finding the label assignment Y that maximizes the score S based on the weight vector λ, shown in Equation 3.5.

$$\hat{Y}_k = \arg\max_{Y} S(X, Y) \tag{3.5}$$

We use integer linear programming (ILP) to solve the prediction problem in Equation 3.5. 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 3.6, is to maximize the sum of the scores of local decisions. In all equations the constant d refers to the document-creation time. The objective is maximized under two sets of constraints, given in Equations 3.7 and 3.8, that express that each candidate is assigned exactly one label, for each sub-task.

$$O = \arg\max_{W} \sum_{x_{i,d} \in C_{dr}(X)} \sum_{l \in L_{dr}} w^l_{i,d} \cdot S(x_{i,d}, y^l_{i,d}) + \sum_{x_{i,j} \in C_{tl}(X)} \sum_{l \in L_{tl}} w^l_{i,j} \cdot S(x_{i,j}, y^l_{i,j}) \tag{3.6}$$

$$\forall_i : \sum_{l \in L_{dr}} w^l_{i,d} = 1 \tag{3.7}$$

$$\forall_{i,j} : \sum_{l \in L_{tl}} w^l_{i,j} = 1 \tag{3.8}$$

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

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

Table 3.3: 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.

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 3.3. Constraints are expressed in terms of the binary decision variables used in the integer linear program.

In Table 3.3, constraints $\mathscr{C}_{Ctrans}$, and $\mathscr{C}_{Btrans}$ model transitivity of CONTAINS, and BEFORE respectively. Constraints $\mathscr{C}_{CBB}$, and $\mathscr{C}_{CAA}$ model the consistency between TLINK relation CONTAINS and DCT-R relations BEFORE, and AFTER respectively (resolving the inconsistent example of $\mathscr{C}_{CBB}$ in section 3.1, and Figure 3.2). Similarly, $\mathscr{C}_{BBB}$, and $\mathscr{C}_{BAA}$ model the consistency between TLINK relation BEFORE and DCT-R relations BEFORE, and AFTER.

Constraints can be applied during training and prediction, as Equation 3.5 is to be solved for both. If not mentioned otherwise, we use constraints both during training and prediction.

3.3.5 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 λ 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 over-fitting. 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 λ_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: $\lambda, \lambda_a, c, I, T$
$c \leftarrow 0$
$\lambda \leftarrow \langle 1, \ldots, 1 \rangle$
$\lambda_a \leftarrow \langle 1, \ldots, 1 \rangle$
for i in I **do**
  for k in T **do**
    $\hat{Y}_k \leftarrow \arg\max_Y \lambda \Phi(X_k, Y)$
    if $\hat{Y}_k \neq Y_k$ **then**
      $\lambda \leftarrow \lambda + \Phi(X_k, Y_k) - \Phi(X_k, \hat{Y}_k)$
      $\lambda_a \leftarrow \lambda_a + c \cdot \Phi(X_k, Y_k) - c \cdot \Phi(X_k, \hat{Y}_k)$
      $c \leftarrow c + 1$
return $\lambda - \lambda_a / c$

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 (with around a factor 2) and address class imbalance at the same time, we sub-sample 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.

Local Initialization

To reduce training time (with a factor 2-3), we do not initialize λ with ones, but we train a perceptron for both local sub-tasks, based on the same local features mentioned in Table 3.1, and use the trained weights to initialize λ 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 3.4.3.

3.4 Experiments

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

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

We will discuss our results in Section 3.4.4. 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.

3.4.1 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 3.4. 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 3.4: Dataset statistics for the THYME sections we used in our experiments.

3.4.2 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 Φ_{joint} of our structured perceptron. We have two competitive state-of-the-art baselines, one for the DCT-R sub-task, and one for the TLINK sub-task. A second baseline is the best performing system of the Clinical TempEval 2016 on the DCT-R task (Khalifa et al., 2016). They experiment with a feature rich support vector machine (SVM) and a sequential

conditional random field (CRF) for the prediction of DCT-R and report the – to our knowledge – highest performance on the DCT-R task. The competitive TLINK baseline is the latest version of the cTAKES Temporal system (Lin et al., 2016b,a). 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.

3.4.3 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 3.3.2, 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 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.

3.4.4 Results

Our experimental results on the THYME test set are reported in Table 3.5. In the table, the abbreviation SP refers to the structured perceptron model described in Section 3.3 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 modified versions to explore the effect of the modifications.

[1] , . / \ " ' = + − ; : ( ) ! ? < > % & $ * | [ ] { }

System	$F^{\text{DCTR}}_{\text{B}}$	$F^{\text{DCTR}}_{\text{A}}$	$F^{\text{DCTR}}_{\text{O}}$	$F^{\text{DCTR}}_{\text{B/O}}$	$F^{\text{DCTR}}_{\textbf{ALL}}$	F^{TL}_{C}	F^{TL}_{B}	F^{TL}_{O}	$F^{\text{TL}}_{\text{BO}}$	$F^{\text{TL}}_{\text{EO}}$	$F^{\text{TL}}_{\textbf{ALL}}$
Baseline: perceptron	77.6	74.4	76.9	52.8	75.9	45.6	14.7	7.3	6.0	2.4	36.4
Best TempEval'16: SVM+CRF	-	-	-	-	84.3	-	-	-	-	-	
cTakes Temporal: 2×SVM	-	-	-	-	-	59.4	-	-	-	-	-
SP	83.7	80.5	86.0	57.5	83.3	**60.8**	29.4	18.5	15.8	23.1	51.8
$\text{SP}_{\text{random sub-sampling}}$	83.7	80.3	85.9	57.5	83.3	56.4	27.5	20.4	15.4	21.8	49.0
$\text{SP}_{disjoint}$	83.5	80.1	85.9	57.6	83.2	60.7	29.0	18.3	14.6	23.2	51.6
$\text{SP}^{cc} + \mathscr{C}_*$	84.3	81.0	86.1	57.3	83.6	60.3	29.2	18.6	14.8	22.2	51.4
$\text{SP}^{uc} + \mathscr{C}_*$	84.3	81.4	86.1	57.4	83.7	60.6	29.1	18.4	15.7	23.6	51.6
SP + Φ_{sdr}	856	830	86.7	56.9	**84.6**	**60.8**	29.1	18.2	15.9	22.2	51.8
SP + Φ_{drtl}	838	811	85.5	56.4	83.1	60.5	28.6	17.6	14.7	21.7	51.4

Table 3.5: Results on the THYME test set, for TLink (tl) and DCT-R (dctr) labels: *before* (b), *after* (a), *overlap* (o), *before/overlap* (b/o), *contains* (c), *begins-on* (bo), *ends-on* (eo). SP refers to our structured perceptron model, without constraints or global features, using local initialization and loss-augmented negative sub-sampling. $\mathscr{C}_*$ refers to using all constraints. Superscript CC and UC refer to using constraints at training and prediction time, or only at prediction time respectively.

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 significantly[2] ($P<0.0001$ for both DCT-R and TLINKS). Furthermore, when comparing loss-augmented sub-sampling (SP) with random sub-sampling of negative TLINKS ($\text{SP}_{\text{random sub-sampling}}$) 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].

Jointly Learning DCT-R and TLINKS

When comparing the disjoint model ($\text{SP}_{disjoint}$) 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 DCT-R). However, joint learning on a document level provides the flexibility to formulate constraints connecting the labels of both tasks, such as the last four constraints in Table 3.3, resulting in a more consistent labeling over both tasks. Similarly, in the joint learning setting, we can define global features that connect both tasks (like Φ_{drtl}).

[2] Significance is based on a paired t-test: pairing the F-scores of both systems per document.

[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.

Integrating Temporal Constraints

We experimented with integrating label constraints in two ways (1) both during training and prediction (SP^{cc} + $\mathscr{C}_*$), or (2) only during prediction (SP^{uc} + $\mathscr{C}_*$). In general it can be noticed that in our experiments using the temporal label constraints from Table 3.3 slightly increases DCT-R performance, but slightly decreases TLINK performance. A reason for this can be that the model generally gives better predictions for DCT-R, 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.

Using Global Structured Features

We have two types of features, Φ_{sdr}, which is only based on DCT-R labels, and Φ_{drtl}, which is based on a combination of DCT-R and TLINK labels. When we add Φ_{sdr} to our model, the overall F-measure on the DCT-R 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 DCT-R labels, also exploited by Khalifa et al. (2016) using the sequential CRF. The second global feature, Φ_{drtl}, in fact models the same type of dependencies as the last four constraints in Table 3.3, relating the TLINK relations with the DCT-R 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.

3.5 Conclusions

In this chapter, we proposed a structured perceptron model for learning temporal relations between events and the document-creation time (DCT-R), 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 timeline construction, we formulated a number of constraints, including those from Chambers et al. (2007) and

Do et al. (2012), and assessed them during inference. Our best system[4] outperforms the state-of-the-art of both the CONTAINS TLINK task, and the DCT-R task.

A slightly extended version of the proposed model Leeuwenberg and Moens (2017a) was used for participation in Clinical TempEval 2017, an international scientific shared task on temporal information extraction from clinical texts, and was ranked 2nd (out of 10 participating systems) in the temporal relation extraction task (Bethard et al., 2017).

This shows that domain knowledge about time, represented as temporal reasoning rules, can be used effectively to improve temporal relation extraction models.

[4]The code for this chapter is available at https://liir.cs.kuleuven.be/software_pages/.

4

Word-Level Multi-Task Learning for Temporal Relation Extraction

This chapter was previously published as:

Artuur Leeuwenberg and Marie-Francine Moens. 2018. Word-Level Loss Extensions for Neural Temporal Relation Classification. In *Proceedings of the International Conference on Computational Linguistics (COLING)*, pp. 3436-3447, Santa Fe, New-Mexico, USA. ACL.

Beside incorporating temporal knowledge in the form of temporal constraints, we can also infuse knowledge about language, or about words. In this chapter, we investigate how we can maximally exploit the available raw text, in addition to our annotated texts, to improve word representations of TLink extraction models. In an earlier error analysis of a state-of-the-art temporal relation extraction system we found that many errors involve words that were unseen at training time (Leeuwenberg and Moens, 2016), indicating the importance of good word representations. Here, we propose a neural temporal relation extraction model which optimizes word-level representations jointly on the temporal relation extraction task, using annotated data, and on an auxiliary context prediction task, using the more easily obtainable unannotated data. By modeling this as a multi-task problem, we aim to balance the information needed for both tasks, and prevent task-specific information loss. We analyze the effectiveness and robustness of this approach, and obtain a further improvement over the state-of-the-art, even without dedicated clinical preprocessing.

4.1 Introduction

Word representations in the form of continuous vectors are often pre-trained on large amounts of raw text to learn general word features, using unsupervised objectives. These representations are then used in supervised models for various classification tasks. However, such tasks sometimes require very specific features that may not have been captured by the unsupervised objective. In other domains such as computer vision, representations are often learned jointly from multiple resources for classification. In this chapter, we explore the possibility to exploit learning signals from both settings to construct better task-oriented word representations, and obtain a better relation classification model.

The main task in this chapter is the extraction of narrative containment relations (CR) from English clinical texts, as annotation of clinical data is costly and it is therefore crucial to fully exploit both the labeled as well as the unlabeled data that is available. The aim of CR extraction is to find if, given events A and B, event A is temporally contained in event B (i.e., if event A happens within the time span of event B). An example of such relation is given in Figure 4.1, where the model should predict all containment edges given the entities (events and temporal expressions), by classifying each pair of entities as containment or no containment. Temporal relation classification in clinical text is a very important task in the secondary use of clinical data from electronic health records. The patient timeline is crucial for making a good patient prognosis and clinical decision support (Onisko et al., 2015). This task has already

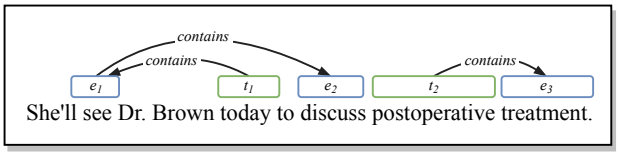

Figure 4.1: A sentence annotated with events (in blue), temporal expressions (in green), and containment relations.

been addressed in three iterations of the Clinical TempEval Shared Task (Bethard et al., 2016). Still there is a gap of more than 0.20 in F-measure between the state-of-the-art CR extraction systems and the annotator-adjunctator agreement (indicating an upper bound for performance). This shows that this task is very challenging.

Following the current trend in NLP, the recent state-of-the-art models for extraction of CR are neural network models. These models all use pre-trained word embeddings as word representations (Tourille et al., 2017a; Dligach et al., 2017; Lin et al., 2017).

Pre-training of the embeddings is done with an auxiliary task (a task where one is not interested in the final predictions, but in the trained model components), like

the skip-gram task (Mikolov et al., 2013). When used for classification tasks in NLP, these pre-trained word representations are often either used as fixed inputs for the classification model, or as initialization for the word representations of the classification model (sometimes called fine-tuned embeddings).

A problem with pre-trained representations in classification models is that solving the main task often requires different information than the auxiliary task. Training word representations only on the auxiliary objective can result in loss of crucial information for the task, and afterwards fine-tuning on the task loss does not influence words that are not in the task's training data.

In the current work, we propose a neural relation classification (RC) model that learns its word representations jointly on the main task (supervised, on labeled data) and on the auxiliary task (unsupervised, on unlabeled data) in a multi-task setting to overcome this problem, and ensure that the embeddings contain valuable information for our main task, while still leveraging the unlabeled data for more general feature learning. As auxiliary task we implement a skip-gram (SG) architecture, similar to Mikolov et al. (2013). Our proposed models use only unlabeled data and a general (news, out of domain) part-of-speech (POS) tagger as external resources, in contrast to the current state-of-the-art models, to ease extension to other languages for which specialized NLP tools for clinical texts might not be available. The main contributions of this chapter are that it:

- Shows that training the word-level representations jointly on its main task and an auxiliary objective results in better representations for classification, compared to using pre-trained variants.
- Shows that the method's increased performance and hyper-parameters are robust across different training set sizes, and that single-loss training settings act as lower bounds on performance.
- Constitutes a new state of the art for temporal relation extraction on the THYME dataset even without dedicated clinical preprocessing.

4.2 Related Work

The model we present draws inspiration from prior research on (temporal) relation classification and neural multi-task learning.

4.2.1 Clinical Temporal Relation Extraction

Temporal relation extraction from clinical texts is a widely studied area in NLP and has been explored through various shared tasks, such as the i2b2 shared task on clinical temporal information (Sun et al., 2013b), and three iterations of Clinical TempEval (Bethard et al., 2015, 2016, 2017). Until recently, most of the top performing systems employed manually constructed linguistic feature sets (Lin et al., 2016a; Lee et al., 2016; Leeuwenberg and Moens, 2017b). In the last few years, there has been a shift towards using neural models, using LSTM (Tourille et al., 2017a) and Convolutional Neural Networks (CNN) models (Dligach et al., 2017; Lin et al., 2017) inspired by the work on relation classification in other domains (Zeng et al., 2014; Zhang and Wang, 2015; Zhou et al., 2016; Nguyen and Grishman, 2015). The top results in clinical temporal relation extraction are still achieved when enhancing the neural models with dedicated clinical NLP tools for preprocessing the clinical texts, often using the English cTAKES system (Savova et al., 2010), which contains tools for clinical POS tagging, named entity recognition, and a dependency parser all trained on clinical data. The main reason for using these dedicated clinical tools is that parsers trained on non-clinical texts perform significantly worse on clinical data (Jiang et al., 2015). Dedicated clinical NLP tools are not available for most languages though, and retraining NLP tools on clinical data is quite resource intensive, because it requires extra annotation effort. Additionally, clinical data is often difficult to obtain or share publicly for patient privacy reasons. Hence, we keep resource intensive preprocessing to a minimum and employ only a general news domain POS tagger (Toutanova et al., 2003), providing important temporal relation extraction cues, such as tense shifts (Derczynski, 2017), and for which training data are available for many languages (Petrov et al., 2012).

4.2.2 Multi-task Learning

Our proposed model training can be seen as multi-task learning (MTL), where the aim is to improve model generalization by leveraging the information from training signals of different related tasks (Caruana, 1998). In earlier work, MTL has shown to be quite effective for different NLP tasks such as machine translation (Dong et al., 2015), sentiment analysis (Peng and Dredze, 2015; Yu and Jiang, 2016), sentence level name prediction (Cheng et al., 2015), semantic role labeling (Collobert and Weston, 2008), and many more. For example, Collobert and Weston (2008) used an auxiliary unsupervised objective for semantic role labeling (SRL). They alternately trained embeddings in a language model and a SRL model. In contrast to their work, we learn both tasks truly jointly, and optimize a single semi-supervised objective. Typically in neural MTL, one or more layers of the network are shared among different models. Two issues in MTL are (1) how to determine if the tasks are related enough to benefit from each other, and (2) what layers to share among the models. Baxter et al. (2000)

theoretically argue that tasks are related when they share an inductive bias. In our model, we expect that the skip-gram task (Mikolov et al., 2013) can act as a reasonable word-level inductive bias for our task, as it has already shown its effectiveness in SRL (Collobert and Weston, 2008) and sentiment analysis (Peng and Dredze, 2015) in MTL, and for many NLP classification tasks when using them as pre-trained embeddings. Hashimoto et al. (2017) showed that even when combining many tasks, considering the task hierarchy (simpler tasks lower in the network) allows them to benefit from each other. In most work on MTL the auxiliary tasks are supervised and specifically chosen for their relatedness to the main task (Ruder, 2017), whereas in our model we chose the unsupervised auxiliary skip-gram task, and share weights of the word embedding layer. This results in a new joint relation classification objective that is semi-supervised on the word-level and provides better generalization for the final classification model.

4.3 The Model

Our model consist of two components: (1) a relation classification component (RC), and (2) a skip-gram component (SG). A high-level schematic overview of our model's training setting is shown in Figure 4.2.

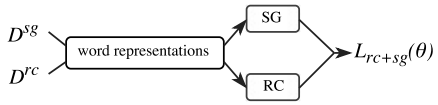

Figure 4.2: High-level overview of our model training setting. D^{rc} and D^{sg} indicate the dataset for the relation classification and skip-gram components respectively, and $L_{rc+sg}(\theta)$ the model's combined loss.

4.3.1 Relation Classification (RC)

To classify relations we employ a long short-term memory (LSTM) (Hochreiter and Schmidhuber, 1997) relation classification model (RC) (Zhang and Wang, 2015). We frame the task as a sequence classification problem, taking as an input: the textual candidate relation description, i.e., the arguments of the candidate relation (the entity pair), and the context words surrounding the arguments, all read as a sequence from left to right. A schematic overview of the RC model component is shown in Figure 4.3.

Generation of candidate entity pairs is described later on in section 4.4.4. The locations of the arguments of each candidate relation are indicated by two types of features taken from the literature: (1) position indicators, which are XML tags added to the original

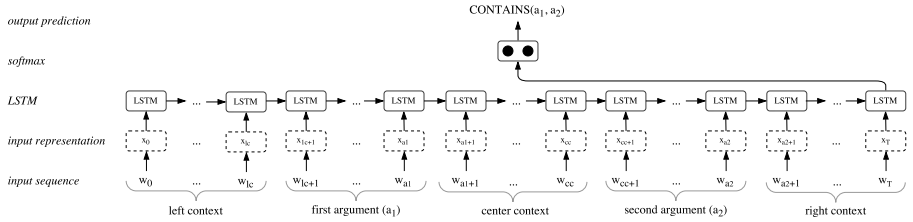

Figure 4.3: Schematic representation of the relation classification (RC) model component. Arrows represent sets of fully connected weights. The dashed box indicates a word input as shown in Figure 4.4.

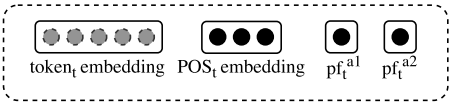

Figure 4.4: Each word input x_t of the RC model (at time step t) is a concatenation of a token embedding, a POS embedding, and two positional features (one for each argument).

input sequence indicating the start and end of the arguments (Zhang and Wang, 2015), and (2) by position features, indicating the relative token-distance of each word to each argument (Zeng et al., 2014). Each word's total input x_t at time step $t \in \langle 0, 1, ..., T \rangle$ consists of the two argument locating features, pf_t^{a1} and pf_t^{a2}, together with a word embedding $x_t^{token} \cdot W_{em}^{token}$, and a POS embedding $x_t^{pos} \cdot W_{em}^{pos}$, where W_{em}^{token} and W_{em}^{pos} are the embedding matrices for tokens and POS respectively, and x_t^{token} and x_t^{pos} their one-hot representations. A schematic overview of the concatenated input x_t for each word to the LSTM unit is shown in Figure 4.4.

The predicted class probabilities $\hat{p}_{rc}(x)$ are given by a softmax classifier placed on top of the LSTM output h at the last time step T (in Eq. 4.1).[1]

$$\hat{p}_{rc}(x) = \textit{softmax}(W_p h_T + b_p) \tag{4.1}$$

The RC model's loss function is the cross-entropy loss, as shown in Eq. 4.2. D^{rc} indicates the supervised relation classification dataset, and θ^{rc} is the collection of all trainable parameters of the model.

$$L_{rc}(\theta^{rc}) = -\sum_{i=1}^{|D^{rc}|} y_i \log \hat{p}_{rc}(x_i) \tag{4.2}$$

[1] We also experimented with bidirectional LSTMs (Zhang et al., 2015) and adding attention (Zhou et al., 2016). In our experiments, this did not result in significant improvements.

4.3.2 Context Prediction (SG)

As the unsupervised auxiliary task, we implemented a feed-forward neural network for a word context prediction task, known as the continuous skip-gram (SG), following Mikolov et al. (2013). As input, the model takes a one-hot encoded input word w_j, which is projected to a word embedding, from which the probability distribution y over its surrounding context words $w_{j-c}, ..., w_{j-1}, w_{j+1}, ..., w_{j+c}$ is predicted, given a context window size c. The full model is given by Eq. 4.3.

$$\hat{p}_{sg}(w_j) = softmax(W_{p_{sg}}(w_j \cdot W_{em}^{token}) + b_{p_{sg}}) \tag{4.3}$$

Like the RC model, we use the cross-entropy loss for our SG model, as shown in Eq. 4.4. D^{sg} indicates the unsupervised dataset, consisting of words and their contexts. θ^{sg} is the collection of all trainable parameters of the model.

$$L_{sg}(\theta^{sg}) = -\sum_{i=1}^{|D^{sg}|} y_i \log \hat{p}_{sg}(w_i) \tag{4.4}$$

Separate Left and Right Context (SGLR)

The skip-gram model is quite rough in its context description and does not take into account word order very well. However, for temporal relations we expect word order to be relevant. For this reason, we also experimented with a variation on the skip-gram model, separating the left and right context, following the intuition of Ling et al. (2015). The context separation is achieved by extending the context words by a 'left' or 'right' prefix depending on their location relative to the sampled word.

4.3.3 Combination (RC + SG)

We train our proposed model on a combination of both loss functions, each with their own dataset D^{rc}, and D^{sg} respectively. The combined loss, shown in Eq. 4.5, is a weighted sum of their cross-entropy losses, where λ_{sg} determines the importance of the SG loss.

$$L_{rc+sg}(\theta) = L_{rc}(\theta^{rc}) + \lambda L_{sg}(\theta^{sg}) \tag{4.5}$$

A crucial part of our model is that although both models sample different types of inputs (the RC: sequences, the SG: single words) from different datasets, and have different classification weights, the word embeddings are shared, i.e., W_{em}^{token} ($\theta^{rc} \cap \theta^{sg} = W_{em}^{token}$), also illustrated in Figure 4.2. So only the word embeddings

are directly influenced by both losses. All other weights (from RC or SG) are only influenced indirectly, through the word embedding weights, as both models are trained simultaneously. Søgaard and Goldberg (2016) showed that, for NLP, sharing representations at the lower levels of the network is most effective: when lower level features are shared, there is room for the model to learn task specific abstractions in higher layers. For this reason we choose our model to share only the word embedding layer, as schematically illustrated in Figure 4.5.

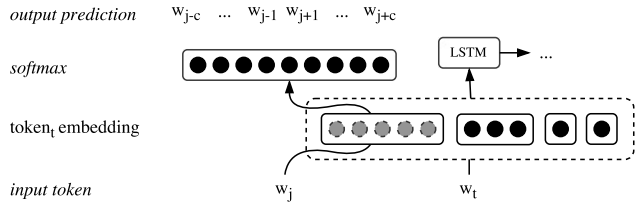

Figure 4.5: Schematic representation of how the SG model component extends the RC model when using the combined loss on input word $w_j \in D^{sg}$, and word w_t at time step t from input sequence $x_i \in D^{rc}$. The gray layer indicates the shared word embedding parameters. The dashed box represents the total word input x_t for RC, as in Figure 4.3 and 4.4.

4.3.4 Training

We train all our models for at least 10 epochs, using Adam (Kingma and Ba, 2014) stochastic gradient descent, with the default parameters from the original paper (lr=0.001), and a batch size of 1024 on a Titan X GPU. As stopping criteria we employ early stopping (Morgan and Bourlard, 1990) with a patience of 20 epochs, based on F-measure on a small validation set of 3 documents (from Train). After each epoch on the main task, we shuffle the data and start the next one. During training we employ a dropout of 0.5 on the input, and on the second to last layer (Srivastava et al., 2014).

For our semi-supervised setting, with the combined loss, we sample 1024 samples in our batch for each task from the corresponding dataset, and do a single weight update on the combined loss. A high-level schematic overview of our semi-supervised training setting is shown in Figure 4.2.

4.4 Experimental Setup

4.4.1 Datasets

We conduct our experiments on the THYME corpus (Styler IV et al., 2014), a temporally annotated corpus of clinical notes in the colon cancer domain, also used in the Clinical TempEval Shared Tasks (Bethard et al., 2015, 2016). We use the provided Train, Dev and Test split so we can directly compare to other approaches from the literature. The Train section consists of 195 documents, with 11,2k annotated candidate relations, the Dev section of 98 documents and 6,2 annotated candidates, and the Test section contains 100 documents and 5,9k candidates. We now refer to this dataset as D^{rc}.

As data for the SG(LR) loss, we use the raw THYME train texts, extended with a MIMIC III (Johnson et al., 2016) section of 500 discharge summaries that contain the terms 'colon' and 'cancer' at least twice. We refer to this dataset as D^{sg} and it is used in all pre-training and joint training settings.

4.4.2 Training Settings

We compare five training settings for the model of which the first three settings act as baselines to compare with our proposed models (settings 4 and 5):

1. **RC (random initialization):** Uses random word embedding initializations (picked from [0.05, 0.05]) and trains on loss L_{rc}.
2. **RC (SG initialization):** Initializes the model with pre-trained SG embeddings, and trains on L_{rc}.
3. **RC (SG fixed):** Initializes the model with pre-trained SG embeddings, and trains the model on L_{rc}, while not updating the word embedding weights, keeping them as fixed features.
4. **RC + SG:** Initializes the model with pre-trained SG embeddings, and trains on L_{rc+sg}.
5. **RC + SGLR:** Initializes the model with pre-trained SG embeddings, and trains on $L_{rc+sglr}$.

4.4.3 Evaluation

As evaluation metrics we use precision, recall, and F-measure, calculated using the evaluation script provided by the Clinical TempEval organizers[2], which evaluates the CR under the temporal closure (UzZaman et al., 2013), taking into account the transitivity properties of the temporal relations.

4.4.4 Preprocessing and Hyper-parameters

As preprocessing of the corpus, we employ very simple tokenization: splitting the text on spaces and considering punctuation[3] and newlines as individual tokens. Additionally, we lowercase the corpus, and conflate digits ($1992 \rightarrow 5555$). To extract POS we use the Stanford POS Tagger v3.7 (Toutanova et al., 2003), using the pre-trained (on WSJ) caseless left-3-words model. Finally, all 1-time occurring tokens in the training dataset are replaced by a <UNK>-token, to represent out-of-vocabulary words at test-time.

We employ the same candidate generation as Leeuwenberg and Moens (2017b), considering all pairs of events (Event×Event, or EE) and events and temporal expressions (Timex×Event, or TE) with a maximum token distance of 30 as candidate relations to be classified (ignoring sentence boundaries, as relations also occur across them). This candidate generation has a maximum recall of 0.87%, and gives a ratio between the positive and negative class of 1:36, also indicating the task's difficulty.

In our experiments, we tuned each model type within the same hyper-parameter search space on the Dev set. The number of LSTM units was chosen from $\{25, 50, 100\}$, and the word embedding dimension from $\{25, 50, 100\}$. This resulted in 100 LSTM units, and a word embedding size of 25. The loss weights λ_{sg} and λ_{sglr} were chosen from $\{0.01, 0.1, 1.0, 10, 100\}$, resulting in λ_{sg}=0.1 and λ_{sglr}=0.1. The context window size of the skip-gram was set to 2, chosen from $\{2, 4, 8\}$. The context size for the RC, and POS embedding dimension size were not tuned and set to 10 (left and right), and 40 respectively.

4.5 Results

4.5.1 Influence of Word-Level Loss

We looked at model performance when increasing the importance of the auxiliary word-level loss (λ_{sg} and λ_{sglr}). The results when changing these hyper-parameters are

[2] https://github.com/bethard/anaforatools

[3] , . / \ " ' = + − ; : ( ) ! ? < > % & $ * | [ ] { }

shown in Figure 4.6.

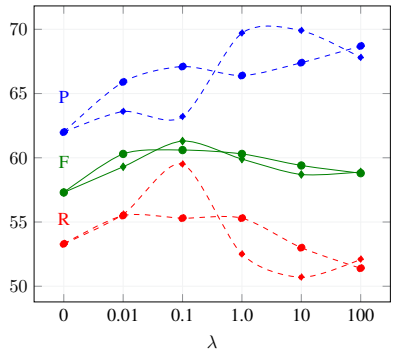

Figure 4.6: Precision (P), Recall (R) and F-measure (F) on the THYME Dev set for different values of λ_{sg} (○) and λ_{sglr} (◇).

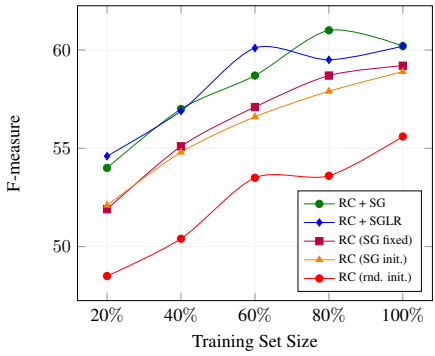

Figure 4.7: F-measure on the THYME Dev set for different training settings, over different training set sizes (in % of the full train set).

When choosing $\lambda = 0$ (λ_{sg} or λ_{sglr}) for our model, we obtain the same model as RC (SG init.), as the auxiliary objective has no influence. For very high values of λ, we hypothesize that the models converge towards RC (SG fixed), because when taking $\lambda \to \infty$, the word embeddings are solely optimized for the auxiliary loss, as the influence of the task loss is proportionally zero, i.e., $\lim_{\lambda \to \infty} \frac{1}{\lambda} = 0$. This property is interesting, as it shows that these baseline models can act as lower bounds for our model performance when choosing a bad λ value. This can be observed in Figure 4.6, where the F-measure is highest for a λ that balances both objectives, whereas for extremes the F-measure decreases.

4.5.2 Comparison Across Training Set Size

We evaluated all model settings for different training set sizes. From Figure 4.7 we can see that the worst model is the one with random word embedding initializations. One improvement is to initialize the model with pre-trained SG embeddings. Fixing the pre-trained embeddings or continued training on the main task objective does not result in very different F-measure scores. However, continued training on the combined objective does seem to give a significant increase in F-measure, consistent over different training sizes for both the RC + SG as well as the RC + SGLR variant. Additionally, it should be noticed that parameters are not returned on each dataset size, but obtained from tuning on the full Dev set. Still the model ranking is consistent.

4.5.3 Evaluation on Subsets of Relations

To get a more detailed insight in what each model learns relative to the others, we evaluated our models on different subsets of the data. First, we split the containment relations based on their argument types and separately evaluated the 3.3k EE relations and the 2.7k TE relations. EE relations are generally found more difficult than TE relations (Lin et al., 2016a,b; Dligach et al., 2017; Lin et al., 2017). In Table 4.1 we can see that also for our model, EE relations are harder to recognize than the TE relations, as all models achieve higher scores for TE compared to EE relations. What is

Table 4.1: Evaluation on subsets of THYME Dev (in F-measure). The subsets of Event×Event (EE) and Timex×Event (TE) relation pairs are of sizes 3.3k and 2.7k respectively. The intervals 0-100, 100-500 and 500+ are subsets reflecting average argument token frequency in the training data (of sizes 2.2k, 2.2k and 1.8k respectively).

Model	EE	TE	0-100	100-500	500+	All
RC (random initializations)	44.5	64.4	40.5	57.8	63.4	53.4
RC (SG initializations)	49.5	68.6	44.1	62.5	67.0	57.3
RC (SG fixed)	48.9	**68.7**	44.1	62.7	67.3	57.6
RC + SG	51.6	67.4	**46.4**	62.5	66.8	58.2
RC + SGLR	**51.7**	68.5	45.3	**63.0**	**68.1**	**58.4**

interesting to see is that when training with the combined loss (SG or SGLR) we obtain a clear improvement on the more difficult EE relations, and perform slightly worse on TE relations compared to using pre-trained embeddings (the three upper settings). The reason could be that EE relations are more diverse in vocabulary, and are consequently more influenced by the quality of the embeddings.

We also analyzed the models w.r.t. total frequency in the training data ($D^{rc} + D^{sg}$) and made three subsets based on the average word frequency of the argument tokens in each relation. The three buckets of relations, 0-100, 100-500, and 500+, are of sizes 2.2k, 2.2k, and 1.8k respectively. What can be observed is that the RC+SG model performs best for low-frequency words, and RC+SGLR performs best for the higher frequency ranges. This can be explained by the fact that the SGLR separates left and right context words, creating sparser and more precise contexts compared to SG. Sparse context descriptions can hurt representations of low frequency words as there may not be enough words that share contexts. But, for more frequent words, more precise context descriptions as in SGLR help to prevents incorrect generalizations (such as cases where word order matters). When evaluating on the full Dev set, both combined loss settings outperform the baselines consistently.

4.5.4 Comparison to the State of the Art

We also compared our proposed models to various state-of-the-art systems from the literature:

The THYME system, by Lin et al. (2016b), consist of separate models for EE relations and TE relations. They employ two feature rich support vector machines (SVM), using POS and dependency parse features from the cTAKES clinical pipeline (Savova et al., 2010) together with augmented training through extended UMLS entities. They later replaced the TE component by a token-based CNN model which improved their model (Lin et al., 2017; Dligach et al., 2017). Also replacing the EE component by a CNN model decreased model performance, showing that the CNN was not able to replace the feature rich SVM. Leeuwenberg and Moens (2017b) used a feature rich structured perceptron, also using cTAKES POS and dependency parse features, jointly learning different relation types on the document level. Tourille et al. (2017a) used two bidirectional LSTM models, one for inter-sentence and one for intra-sentence relations. They used fixed word embeddings pre-trained on the MIMIC III corpus, and also incorporated character level information. To obtain their top results they added ground truth event attribute features enhanced with entity information also obtained from cTAKES.

As can be noticed, all state-of-the-art baselines used dedicated clinical NLP tools to enhance their features in order to obtain their top results, in contrast to our model, which uses only the Stanford POS Tagger (trained on news texts).

Table 4.2: THYME test set results, reporting precision (P), recall (R) and F-measure (F), macro-averaged over three runs. The standard deviation for F is also given.

Model	P	R	F
With specialized resources:			
Best Clinical TempEval (2016)	58.8	55.9	57.3
Lin et al. (2016)	66.9	53.4	59.4
Leeuwenberg et al. (2017)	-	-	60.8
Tourille et al. (2017)	65.7	57.5	61.3
Lin et al. (2017)	66.2	58.5	62.1
No specialized resources:			
RC (random initialization)	67.9	52.1	$58.9_{\pm 0.2}$
RC (SG initialization)	**71.2**	52.0	$60.0_{\pm 1.2}$
RC (SG fixed)	68.9	54.6	$60.9_{\pm 0.8}$
RC + SG	66.2	**59.7**	$\mathbf{62.8}_{\pm 0.2}$
RC + SGLR	68.7	57.5	$62.5_{\pm 0.3}$

Table 4.2 shows that initializing the model with the pre-trained embeddings gives a significant [4] 1.1 point increase in F-measure compared to random initialization, due to an increase in precision. Fixing the embeddings gives slightly better performance than using them as initialization, an increase of 0.9 point in F-measure, mostly due to higher recall. When extending the loss with the SGLR loss, we gain[4] 1.6 in F-measure compared to fixing the word embeddings, and also surpass the state of the art by 0.4 even without specialized resources. If we train our model using the SG loss extension we obtain the best results, and gain[4] 1.9 points in F-measure compared to using pre-trained fixed word embeddings. This setting also exceeds the state of the art (Lin et al., 2017) by 0.7 points in F-measure, due to a gain of 1.2 points in recall, again without using any specialized clinical NLP tools for feature engineering, in contrast to all state-of-the-art baselines.

4.5.5 Manual Error Analysis

Finally, we manually analyzed 50 false positives and 50 false negatives picked randomly from the test set predictions for different settings.

From Table 4.3 we can see that all models have difficulties with distant relations that cross sentence or clause boundaries (CCR). This could be because class imbalance correlates with distance between the arguments of the temporal relations. Furthermore, arguments that are frequent in the supervised data (> 250) are a dominant error category. We suspect this is because frequent events often function both as container and as contained, whereas infrequent events are less ambiguous in their argument position. This hurts RC (SG fixed) most as its embeddings are not influenced by D^{rc}. Furthermore it can be noticed that RC+SG has less errors for infrequent arguments (< 10) in the supervised data. This could be because it leverages the few available instances from both the D^{rc} and D^{sg} data better than the single-loss models.

Table 4.3: Error analysis on 50 FP and 50 FN (random from test) for different settings. Clause boundaries are: newlines and sub-clause or sentence boundaries. Error categories are not mutually exclusive.

Error Type	RC + SG	RC (SG fixed)	RC (SG init.)
Cross-Clause Relations (CCR)	42	39	36
Infrequent Arguments (< 10)	11	15	26
Frequent Arguments (> 250)	37	50	40
Mistake in Ground-Truth	10	8	5
Other	21	15	28

[4] $P < 0.0001$ for a document-level pairwise t-test

4.6 Conclusions

In this chapter, we proposed a neural relation classification model for the extraction of narrative containment relations from clinical texts.[5]

The model trains word representations jointly on the supervised relation classification task and an unsupervised auxiliary skip-gram objective (with separate datasets) through weight sharing to more effectively exploit both the unlabeled and labeled data, as annotated clinical data is costly to create. Our results show that this word-level joint training results in significantly better generalizing classification models compared to using pre-trained word embeddings (either as initialization or fixed embeddings). Furthermore, we show that performance trends and good values for λ (balance between tasks) are robust over different training set sizes, and that even for (badly tuned) extreme values of λ the quality of the model's embeddings is naturally lower-bounded by their pre-trained variants. Additionally, our model sets a new state of the art for temporal relation extraction on the THYME dataset, without using extra dedicated clinical resources, in contrast to current state-of-the-art models.

As future work, it would be interesting to see how well the improvements caused by the word-level joint training generalize to other NLP tasks that typically use pre-trained word embeddings, and other sub-task of temporal information extraction.

[5]The code for this chapter is available at https://liir.cs.kuleuven.be/software_pages/.

5

A Survey on Temporal Reasoning for Temporal Information Extraction

This chapter was previously published as:

Artuur Leeuwenberg and Marie-Francine Moens. 2019. A Survey on Temporal Reasoning for Temporal Information Extraction from Text. In *The Journal of Artificial Intelligence Research (JAIR)*, pp. 341-380. AI Access.

From the previous two chapters we have observed how integration of domain knowledge, either about the structure of time (Chapter 3) or about the structure of language (Chapter 4), can improve temporal relation extraction models. To further progress towards our final goal of building event timelines, we dive deeper into the structure of time, and investigate how temporal reasoning can be used as a tool to manipulate temporal information. For this, we conducted a literature survey on how temporal reasoning has been used successfully used in the research field of temporal information extraction, and provide a clear overview of existing approaches.

The survey explains the required theory on temporal reasoning, and discusses its role in all steps of model construction: annotation, data preprocessing, model training, prediction, and evaluation. Based on the presented overview, we detect gaps in the literature, and highlight promising directions for future research.

5.1 Introduction

The phenomenon of time has a major influence on how people perceive, and communicate through language. Consequently, our language utterances are filled with temporal cues on the events we communicate about. **T**emporal **I**nformation **E**xtraction (TIE) is the process of automatically extracting temporal cues from text, with the final goal to construct a (possibly underspecified) timeline of events from them, as shown in Figure 5.1.

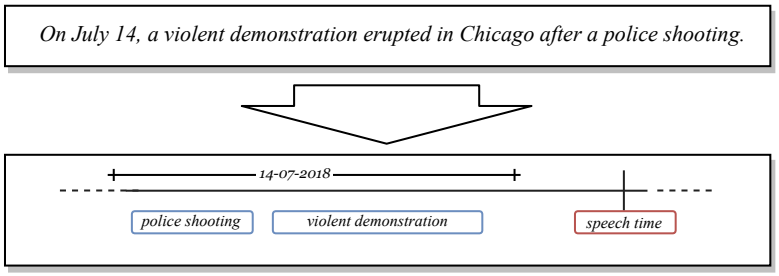

Figure 5.1: An example of temporal information extraction.

Temporal information extraction not only plays a major role in the general problem of natural language understanding (NLU), but is also used in many applications, like information retrieval (Campos et al., 2015), question answering (Llorens et al., 2015; Höffner et al., 2017; Meng et al., 2017; Sun et al., 2018; Pampari et al., 2018), and multi-document summarization (Barzilay and McKeown, 2005; Ng et al., 2014), and has great potential in the clinical domain, for applications like patient timeline visualization (Jung et al., 2011), forecasting treatment effects (Augusto, 2005; Zhou and Hripcsak, 2007; Choi et al., 2016a), better early detection of diagnoses (Choi et al., 2016b), or patient selection for clinical trials (Raghavan et al., 2014). Because of the strong linear structure of time itself, and the great variation in the types of temporal cues we can express in language, a central challenge in temporal information extraction is how to combine all these separate cues into a single coherent temporal ordering of the described events. To obtain this temporal ordering from many different cues temporal reasoning is of vital importance. We define **T**emporal **R**easoning (TR) in the context of TIE as the process of combining different (annotated or extracted) temporal cues to derive additional temporal information about the text. TR is crucial for TIE and has already been exploited widely in the research community in every step in the process of TIE model construction: annotation, pre-processing, model training, inference, and evaluation. Despite the importance of TIE for NLU and the crucial role of TR in TIE, there has not yet been a survey covering the research in this area.

5.1.1 Focus

For successful TR in practical settings two factors are very important: (1) The *expressiveness* of the TR mechanism: How complete is the temporal knowledge that the TR mechanism can infer from the temporal cues? And (2), the *efficiency* of TR: What are the computational costs of TR to infer that new temporal knowledge. These two points will get extra attention in this survey.

Although most research has focused on extraction models for certain types of temporal cues, this survey focuses on the big picture of complete TIE, where the aim is to extract all temporal cues from the text and combine them into a single coherent temporal view, for which a good TR mechanism is crucial. To cover the evolution of TR approaches for TIE in parallel with the popularization of using machine learning (ML) methods for natural language processing (NLP), the focus of this survey lies on the research on TR for TIE systems from the past three decades. We abstract from what linguistic features are successful for TIE systems, as these are discussed in depth by Derczynski (2017), and can be considered complementary to the focus of this survey.

5.1.2 Contributions

In the context of the interesting and important new developments in the past years this survey provides the following contributions:

- A clear explanation introducing the theory and background on TR for TIE required to comprehend the latest state-of-the-art TIE models.
- A structured overview of the various ways in which TR has been exploited in TIE models over the past three decades: in annotation, pre-processing, training, prediction, and evaluation.
- A distillation of the most important conclusions to successfully incorporate TR in a TIE system.
- Directions for future work and on promising unexplored avenues in the research area of TIE.

5.1.3 Structure

The survey is structured as follows: First, in section 5.2, we provide an exemplified overview of what types of temporal information are present in natural language texts. These include relative and absolute cues, definite and indefinite cues, implicitness of

temporal information, world knowledge, and the role of under-specification, as these aspects play an important role in temporal reasoning. In section 5.3, we introduce the theory of temporal reasoning that is required to comprehend and assess the current state-of-the-art models and methods, as discussed in the following sections. Section 5.4 describes the most widely used annotation schemes for temporal information, and discusses how they relate to temporal reasoning frameworks. In section 5.5, we arrive at the core of this survey, and provide a complete and comprehensive overview of the literature on TR for TIE. Then, in section 5.6, we give suggestions on promising directions and less explored areas based on the earlier sections. Lastly, in section 5.7, we summarize the most important findings and conclusions of the survey.

5.2 Temporal Information in Language

In this section, we give a short (exemplified[1]) overview of different temporal cues that can be expressed in language to show what types of temporal information the cues can provide, i.e., in what way the cues may possibly constrain the position of event intervals on the timeline. We focus less on the different ways temporal cues can be expressed, i.e., linguistic variation, as this has no direct impact on TR.

It is important to study the types of temporal information that can be expressed by temporal cues because the different cues need to be combined by a TR system in order to build a complete temporal picture of the text, or construct a timeline.

5.2.1 Timeline Components Captured by Temporal Cues

Temporal cues can bound various components of the event timeline: full positions of *intervals*, but also just the *start*, *end*, or *duration* of intervals.

For instance, in Example 1, the duration of the antibiotics administration is given (10 days), and so is its start time (somewhere on the 2nd of June). While, for the improvement of the respiratory status only the end time is mentioned explicitly (last 2-3 days of the antibiotics administration). This shows that for a fairly simple text fragment, a TR system already needs to be able to combine temporal bounds on at least three different components of the timeline.

Example 1. *Antibiotics were started on 6/2 and continued for 10 days. Respiratory status improved up til the last 2-3 days.*

[1] Examples from the New York Times, and the clinical i2b2 corpus (Sun et al., 2013a).

5.2.2 Relative and Absolute Cues

As shown in the previous example, temporal cues can provide *absolute* references to the timeline, by referring to absolute intervals, like dates or times, or absolute durations, like a certain number of hours. In Example 2, the duration of the third set (*28 minutes*) is an absolute cue. However, additionally quite often the temporal cues provide *relative* information. In the example, there are three explicit relative cues: (1) the duration of the first two sets are *less than* 1 hour, (2) the third set started *after* the first two, And (3) the whole situation took place in the past, i.e., before the speech time (ST) of the sentence, indicated by the past tense of the verbs. The ideal TR system should be able to resolve combinations of absolute and relative cues.

Example 2. *After the grueling duels of the first two sets, which each had taken nearly an hour, Nadal won the third set in 28 minutes.*

5.2.3 Definite and Indefinite Cues

Quantification of timeline components can be definite, referring to clear quantities, or indefinite, using vague quantification. In Example 3, the duration of the patient being HIV positive is definite (*2 years*). Whereas the duration of the left upper quadrant pain is, although explicit, quantified vaguely (*long-standing*), and hence indefinite.

Example 3. *The patient is a 27-year-old woman who is HIV positive for two years. She presented with left upper quadrant pain which is a long-standing complaint.*

5.2.4 Implicitness and World Knowledge

A significant part of the temporal information conveyed in text is implicit. Event durations, and event position are often implicit, or considered common world knowledge. In Example 4, although not mentioned explicitly the *charges* have been made before the judge's question. Also, the man's answer probably lasted only a few seconds, and happened clearly after the judge's question. Whereas, if we would have

replaced *answered* with *trembled*, this would probably have lasted longer, and was possibly already going on during the judge's question.

Example 4. *Wearing a black scarf, slim-fitting navy suit and tortoiseshell glasses, he said little and answered, "I do, your honor", when the judge asked if he understood the charges.*

5.2.5 Underspecification

A major aspect of temporal information extraction is underspecification. Almost in all cases, temporal cues do not provide a fully specified timeline (full absolute positioning of events on the calendar), leaving open multiple temporal interpretations. As can be seen in all previous examples 1-4, it was never mentioned, for example, in what year the events took place, allowing multiple valid timeline interpretations, something that temporal reasoning systems should be able to deal with appropriately.

5.3 Frameworks for Temporal Reasoning

In the previous section we observed that language can contain many different types of temporal information. To combine all these different types of temporal information into a coherent temporal view, or timeline, we require temporal reasoning. To reason with different types of temporal information about events, several frameworks have been developed. The temporal interpretation of an event is generally considered as an interval on the timeline. The span of the interval corresponds to the time that the event takes place. Consequently, TR often addresses reasoning with intervals. As a reference, Fisher et al. (2005) provide an overview of general TR, but do not discuss TIE systems in detail. Here, we review the TR frameworks that have been used for TIE, as a back-bone for section 5.5, where the integration of TR in TIE systems will be discussed.

5.3.1 Allen Interval Relations

One of the most popular TR frameworks used in TIE was proposed by Allen (1983). He proposed a set of thirteen mutually exclusive *basic* interval relations that could be assigned to any pair of definite intervals. These relations and the corresponding

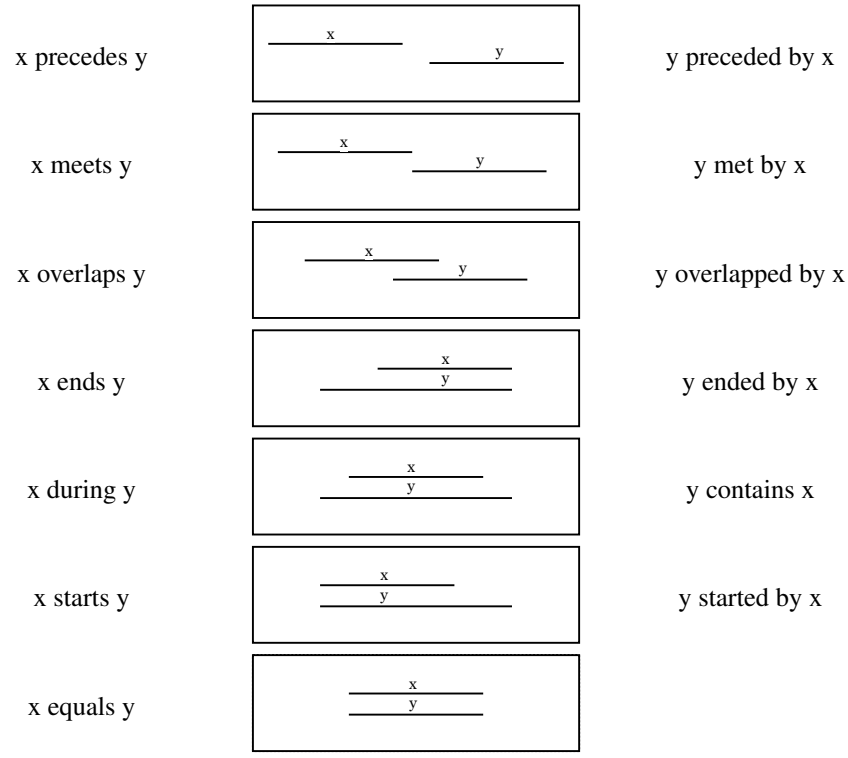

Figure 5.2: Allen's thirteen basic interval relations.

operations are known as Allen's interval algebra. All thirteen basic relations and their visualizations are shown in Figure 5.2. As can be seen, the thirteen basic relations are six pairs of converse relation pairs, and the *equals* relation, which is symmetric. From these basic relations that can only represent relations between *definite* intervals, where relative positions of start and end-points are known, indefinite interval relations can be constructed, where the start or end of the intervals might be incomplete. In Allen's algebra each indefinite relation (called a *general* Allen relation) is represented as a *disjunctive* set of basic relations. The set representation of a general Allen relation between two events contains all basic relations that are possible between the events, given the (incomplete) information about their starts and endings.

Example 5. *y started sometime during x*

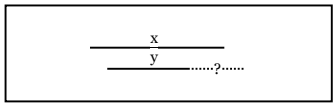

Figure 5.3: Visualization of Example 5 with the general Allen relation: x {contains, overlaps, ended by} y.

For example, the sentence of Example 5 could be represented by Figure 5.3. Only relative information about the start of y is known, namely that it lies within the boundaries of x. However, there is no information given about the end of y, making it impossible to assign a basic Allen relation, as the intervals are indefinite. The correct general Allen relation is: x {*contains*, *overlaps*, *ended by*} y, and indicates that any of these three basic relations in the set, *contains*, *overlap*, or *ended by*, could apply to the situation. The full set of Allen interval relations is the power set of the basic relations, resulting in $2^{13} = 8192$ relations. Notice that when no information about the relation between two intervals (or events) x and y is given, any basic relation is possible. So, in that case the general Allen relation between x and y would be represented by the set containing all basic relations. In other words, the less we know, the more is possible, so the bigger the representation.

Temporal Closure

To infer new relations from a set of general Allen relations relating different events, a composition table is used. The table contains transitivity rules for all basic relations, i.e., it shows for any pair of connected relations r_1(x, y) and r_2(y, z), what relation r_3(x, z) could be inferred. An example for the transitivity for the *precedes* relation is given in Figure 5.4. Using this principle, a temporal closure (called 'Propagate' in the original paper) can be calculated, adding new relations to the existing set. Computing the full closure, which includes all possible inferences that can be made, is NP-complete, making it highly intractable (Vilain et al., 1990). Often, in practice, only a subset of the transitivity rules is used.

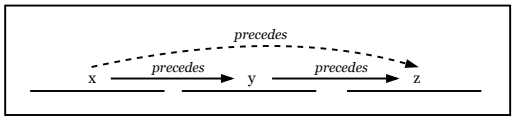

Figure 5.4: An example of an inferable precedes relation (dashed) through transitivity: x precedes z is inferred from the fact that x precedes y and y precedes z.

Temporal Consistency

An important task in TR is checking temporal consistency for a set of relations, as a timeline can only be constructed from a consistent set of relations. In Allen's algebra, temporal consistency can be calculated by checking if, when going through all possible chains of inference, the intersection of all inferable relations for each pair is not empty. In other words, an assignment of general Allen relations is consistent if at least one basic Allen relation can be assigned to each pair of intervals, after closure.

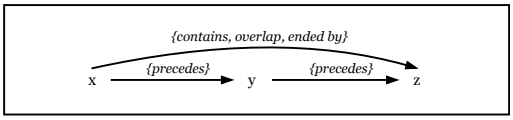

Figure 5.5: An example of an inconsistent assignment of Allen relations.

An example of an inconsistent assignment of relations is shown in Figure 5.5. The example is inconsistent because from the fact that x *precedes* y and y *precedes* z it can be inferred that x {*precedes*} z. And, when taking the intersection for pair (x,z) of the inferred relation x {*precedes*} z and the already assigned relation x {*contains*, *overlap*, *ended by*} z we end up with the empty set. This indicates there are no possible basic relations for pair (x,z). From this we can conclude that the example is not consistent. Calculating temporal consistency is, like the temporal closure, NP complete when using the full Allen algebra (Vilain et al., 1990) as it requires temporal closure. This high computational complexity is not very practical in real applications. For this reason, more efficient solutions have been proposed, which we will cover in the next section.

5.3.2 Subfragments of the Full Allen Algebra and Point Temporal Algebra

Because of the high complexity of calculating temporal closure and consistency for the full Allen algebra, many different more tractable sub-fragments of Allen's interval algebra have been proposed (Vilain et al., 1990; Freksa, 1992a; Nebel and Bürckert, 1995; Ligozat, 1996; Krokhin et al., 2003). Some also focus on integrating quantitative reasoning (Dechter et al., 1991; Meiri, 1996; Dechter and Cohen, 2003) or uncertainty (Schockaert and De Cock, 2008). Although most current research in TIE systems has focused on using Allen relations, representing mostly relative interval cues, the ability to combine quantitative temporal cues and being able to deal with uncertainty is very important, as the variance of cues in language is vast, as seen in the previous section. And all cues need to be taken into account to construct a fully coherent temporal view from the text. We will not discuss all the sub-fragments and extensions in this survey as the vast majority of these extensions have not been used in current TIE systems.

Rather, we provide a theoretical back-bone on which many of these sub-fragments are built. We also use this back-bone to define a categorization of TR-expressiveness in which we will later classify the different TR approaches used for current TIE systems.

A major insight on which many efficient TR algorithms are built is the fact that the basic Allen interval relations can be expressed as sets of point-relations in the *point temporal algebra*. From this perspective, each pair of intervals (x,y) can be seen as a set of four points: the starts of both intervals x^-, and y^-, and their endings: x^+, and y^+. In the point temporal algebra, there are three point-wise relations that can occur between each of these points: $<$, $=$, and $>$. These point-relations can be used to express each basic Allen relation as a *conjunctive* set of point relations on the start and endings of the intervals. For example, x {*equals*} y can be expressed by the conjunctive set $\{x^- = y^-, x^+ = y^+\}$, i.e., iff the start and endings of intervals x and y are equal, then x and y are also equal. As another example, $x\{precedes\}y$ can be expressed as $\{x^+ < y^-\}$, i.e., iff the end of x lies before the start of y then interval x lies before interval y. In general, because the four points describe starts and endings of intervals, it is always the case that each interval's start x^- lies before its end x^+. Based on their

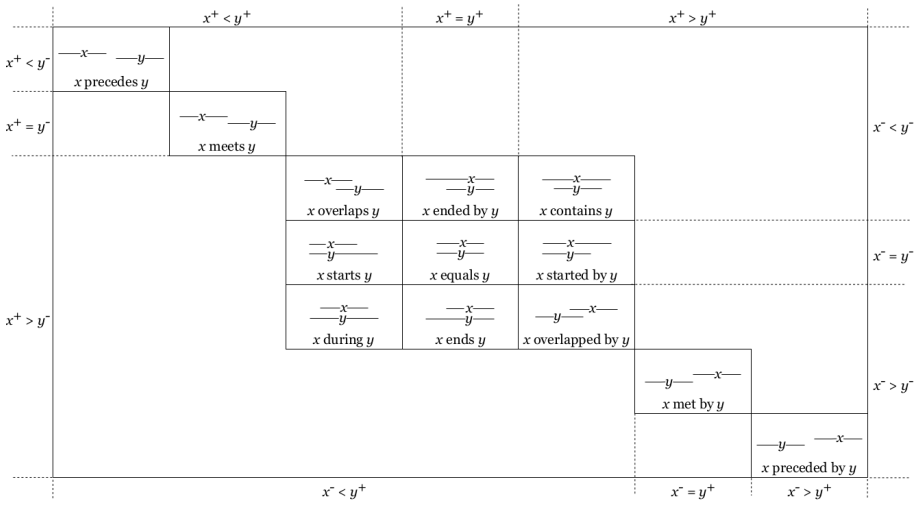

Figure 5.6: The lattice showing the relation between point algebra and the basic Allen interval relations, and the conceptual neighborhood between interval relations (Freksa, 1992a).

point algebraic representations, Allen's basic interval relations can be ordered in a very informative lattice (Freksa, 1992a), as shown in Figure 5.6.

From this lattice we can read several things:

1. How to convert a basic Allen interval relation to a set of point algebraic constraints: start from a basic interval relation box in the lattice, and then take the union of all point-algebraic constraints on the corresponding outsides of the rectangle.

2. How to construct a (general) Allen relation from a set of point-algebraic constraints: start from the point-algebraic constraints on the outside of the rectangle, and take the intersection of the interval relations that are covered by the area inside the rectangle corresponding to the point-algebraic constraints.

3. How to determine the conceptual neighborhood between two basic Allen interval relations: count how many boxes have to be crossed to get from one basic interval relation to the other (i.e., how much do the end-points need to shift to change from one interval relation to the other).

For example, if we have a cue saying that the start of event y happens somewhere during event x, i.e., $\{x^- < y^-, x^+ > y^-\}$ (as in Example 5), we can read from the lattice what relations are covered by the overlapping area of these constraints: {*contains*, *overlaps*, *ended by*}, from which we can conclude that the corresponding general Allen relation is x {*contains*, *overlaps*, *ended by*} y.

Using this link between interval and point relations TR about intervals can be done in the point algebra, which is much more efficient, as it has only three basic relation types, resulting in a much smaller composition table. Additionally, expressing interval relations by conjunctive sets of point relations, instead of disjunctive sets of basic interval relations, ensures that when we have little temporal knowledge, the set representation is smaller, instead of bigger (as with general Allen relations). These two components contribute to the fact that TR in temporal point algebra fragments has only polynomial complexity instead of the NP-completeness of the full Allen algebra (Vilain et al., 1990). This gained efficiency and flexibility in representation are very important when considering practical TIE systems. Even more when combining temporal cues from different documents, or different sources for which complex temporal resolution of the different cues is required to obtain a coherent timeline.

As mentioned in the beginning, we cannot express each general Allen relation as sets of point-algebraic constraints, but only a fragment of them. Point algebra can only express interval relations that can be represented as a conjunction of point-algebraic constraints. For example, the general Allen relation x {*precedes*, *preceded by*} y cannot be expressed as conjunction of point-algebraic constraints. We can see this from the lattice, as we cannot capture these two interval relations in a single rectangle, without including other basic relations as well.

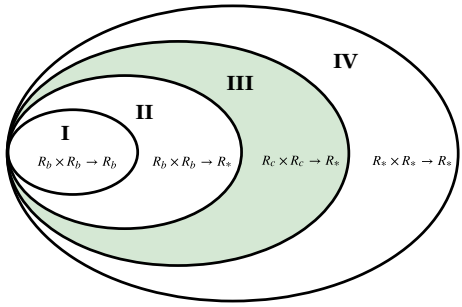

Figure 5.7: Onion diagram showing four classes of temporal reasoning rules, indicating expressiveness of the temporal reasoning. R_b stands for any basic Allen relation, R_c stands for any Allen relation that can be expressed as conjunction of point-algebraic constraints, and R_* can be any Allen relation. Layer III is the most expressive class that still operates in polynomial time, which is important for practical systems.

Expressiveness of temporal reasoning

Later on, in section 5.5, we review the currently existing TR-based TIE systems and categorize them in various ways in order to compare them. To categorize the TIE models with regard to the expressiveness of their TR component we construct four classes based on the part of the transitivity table that is used for TR. This categorization is shown in Figure 5.7: The most inner layer covers rules that only use basic Allen relations (R_b) in the condition and conclusion of the rules. The second layer covers TR rules that have only basic relations in their condition, but can have general Allen relations (R_*) in the conclusion. The third layer covers the sub-fragment that is translatable to point algebra, covering all rules that have relations in the condition of the rule that can be expressed as conjunction of point-algebraic constraints (R_c). Finally, the outer layer is the full Allen algebra, covering the full transitivity table. Outer layers include inner layers in terms of rule sets, and are more expressive, but TR can be more computationally complex (depending on the implementation).

Layer I type reasoning is used quite commonly, as most systems build on basic Allen relations, layer II is more expressive but used less, as it involves reasoning with sets of basic relations instead of only basic relations. For practical TIE systems, layer III is very important for three reasons: (1) It is the most expressive fragment that can be implemented in polynomial time; (2) The full fragment can be mapped to point-algebraic constraints and back, introducing flexibility with regard to combining different types of temporal cues; and (3) It covers all basic Allen relations, which is beneficial as these are often available in the annotated data. Before examining the TR used in TIE systems (section 5.5), we discuss the annotation of temporal information.

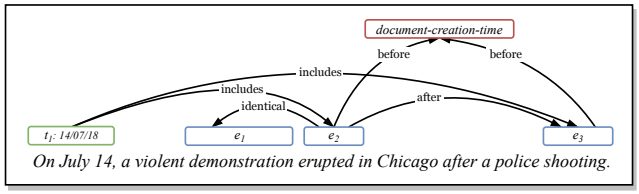

Figure 5.8: An example of TimeML-style annotation: Events in blue, a (normalized) temporal expression in green, the document-creation-time in red, and arrows indicating the temporal links (TLinks) among them.

5.4 Annotation of Temporal Information

The annotation scheme for temporal annotation in text steers the types of temporal cues that can be extracted by TIE systems, and therefore also how TR can be exploited. The most widely used scheme is TimeML (Pustejovsky et al., 2003a), which is an ISO-standard for annotating temporal information in text (Pustejovsky et al., 2010). Most temporal corpora have been annotated with TimeML, or a very similar (sub)scheme (Pustejovsky et al., 2003b; Sun et al., 2013a; Cassidy et al., 2014; Styler IV et al., 2014). And TimeML has been used in TempEval, a series of shared tasks on evaluating temporal information extraction that resulted in many of the existing TIE systems (Verhagen et al., 2007, 2010; UzZaman et al., 2013; Minard et al., 2015; Llorens et al., 2015). We will discuss the shared consequences for TR of the TimeML scheme. In TimeML the core concepts are *event expressions* and *temporal expressions*. Event expressions refer to events in the real world, and can be of different types, like states (e.g., *bankrupt*), actions (e.g., *sailing*), occurrences (e.g., *meeting*), reporting events (e.g., *said*) among other types. Temporal expressions (timex) refer to calendar dates (e.g., *21st of August, 2018*), times (e.g., *1 o'clock*), definite durations (e.g., *two hours*), or sets of times (e.g., *3 times a week*). The function of these timex expressions is to anchor events to the calendar timeline. In TimeML, events and timex expressions are temporally connected through temporal links (TLinks). TLinks can have thirteen types that almost (but not exactly) follow Allen's basic interval relations. An example of a TimeML-annotated sentence in shown in Figure 5.8. The TLink types are shown in Table 5.1. TLinks can be annotated between three categories of candidate pairs:

1. Between events (EE-R).
2. Between temporal expressions and events (TE-R)
3. Between each event and the document-creation-time (DCT-R).

Table 5.1: Temporal link (TLink) types from TimeML and their corresponding basic Allen interval relations.

TimeML TLink	Basic Allen Relation
BEFORE	*precedes*
AFTER	*preceded by*
INCLUDES	*contains*
IS INCLUDED	*during*
IMMEDIATELY BEFORE	*meets*
IMMEDIATELY AFTER	*met by*
BEGINS	*starts*
BEGUN BY	*started by*
ENDS	*ends*
ENDED BY	*ended by*
DURING	*during* | *equals*
DURING_INV	*contains* | *equals*
-	*overlap*
-	*overlapped by*
SIMULTANEOUS	*equals*
IDENTICAL	*equals*

As can be seen in the Table most TLinks match with a basic Allen relation. However, Allen's *overlap*, and *overlapped by* relations are not represented (UzZaman and Allen, 2011), and the temporal interpretation of DURING and DURING_INV relations seem similar to IS INCLUDED and INCLUDES, but are not clearly defined (Chambers et al., 2007; Derczynski et al., 2013; Derczynski, 2016), and are also sometimes interpreted as SIMULTANEOUS (UzZaman et al., 2013). The difference between SIMULTANEOUS and IDENTICAL is that SIMULTANEOUS can apply to two different events happening at the same time, whereas IDENTICAL means two event mentions refer to the exact same event (event co-reference).

In terms of expressiveness, TimeML models a small subset of the full Allen algebra (2^{13} relations), and sticks to modeling the basic Allen relations. This is because temporal annotation is a complex task for annotators, and annotation complexity needs to be taken into account to obtain high-quality annotations with reasonable inter-annotator agreement.

Nevertheless, the expressiveness of TimeML is expanded by also including timex annotations as calendar anchors. Because timexes of types *date* and *time* can be temporally interpreted as absolute intervals with clear positions on the calendar that carry a clear temporal ordering with respect to each other (e.g., *1990* is always before *1991*). And similarly, timex with type *duration* can be interpreted as quantified interval durations that come with an implicit order on durations (e.g., *1 hour* is always

shorter than *2 hours*). Hence, there is a fourth category of temporal links that can be automatically derived from timex annotations:

4 TLinks between temporal expressions (TT-R).

While using timexes as calendar anchors to increase the amount of temporal information captured, the expressivity of TimeML is limited by the expressivity of basic Allen relations. As shown in sections 5.2 and 5.3, the temporal information that can be expressed in language not always concerns definite intervals, for which basic Allen relations are ideal. Underspecification of an event's duration or ending in the text could potentially cause disagreement between annotators, as no basic Allen relation would be suitable. However, including general Allen relations into the annotation scheme, to model temporal uncertainty, could make the annotation task very complex.

As in the construction of many TimeML corpora annotators are not forced to annotate all candidate pairs, this regularly results in sparse TLink annotations. Sparse annotations can (1) make extraction difficult because of class imbalance, and (2) cause problems in evaluation because extraction systems can get penalized for predicting relations that the annotator may have missed. An attempt to address sparse temporal graphs is the TimeBank Dense corpus by Cassidy et al. (2014), who explicitly asked annotators to annotate relations between all events, within a certain token window. Whenever no basic Allen relation could be assigned, annotators are asked to assign the label *vague*. Although *vague* does not tell a lot about the degree of temporal uncertainty, and could in fact be replaced with the general Allen relation including all basic relations in terms of reasoning, it does make very clear what pairs have clear orderings, and what pairs do not. An interesting unexplored avenue might be to annotate the *vague* relations in the TimeBank Dense with their general Allen relations to obtain an even more complete temporal graph.

Ning et al. (2018c) recently addressed the issue of temporal uncertainty as well, and also pose the question whether all events can actually be related to each other temporally. They propose a multi-axis annotation scheme where they first separate events that are anchorable on the timeline, from other events (negated events, opinions, intentions). They assume that if two events are on the same axis, they can be temporally related. In a second step, they ask annotators to annotate temporal point-wise relations (*before*, *after*, *equals*, and *vague*) between the starting points of the anchorable events, instead of the conventional interval relations, obtaining high inter-annotator agreement even on crowd-sourced annotations. They found when also asking annotators to annotate relations between end-points this was found much more difficult by the annotators. This is possibly explained by the difference in interpretation of event durations between annotators, as these are often not explicitly mentioned in the text, and assume background knowledge shared by the readers and writers. This scheme is interesting as it annotates a different set of relations than the ones used by TimeML.

We can see the relation between the two clearly from the lattice in Figure 5.6, as the relations between start points can be separated by the right y-axis of the lattice ($x^- < y^-$, $x^- < y^-$, $x^- > y^-$). This shows that if a TR component would be able to deal with both interval and point-relations, data from both schemes could be combined this way to learn or evaluate TIE systems, which could be very useful.

Although Ning et al. (2018c) do not describe duration annotations, TimeML includes duration annotations. In TimeML, cues on duration need to be explicitly mentioned by a timex in the text in order to be annotated. The fact that a *long meeting* takes longer than a *short meeting* is not annotated as *long* and *short* are not timex expressions. Also background knowledge about typical event duration, in case the duration is not mentioned explicitely by a timex, is not annotated by TimeML. Pan et al. (2006a,b, 2011) proposed an annotations scheme fully dedicated to annotating event durations. They asked annotators to provide quantified bounds on the durations of the events mentioned in the texts (like *1-10 minutes*, or *1-2 days*). This annotation is done on the event level and does not explicitely annotate timex expressions. This way annotators are free to use any cues they can find in the text, or their background knowledge[2].

Another recent scheme including event durations was proposed by Reimers et al. (2016, 2018). They classify events into two coarse duration types: one-day and multi-day events. Similarly to TimeML they annotate timex expressions and link events to these timex expressions when possible. However, when this is not possible, annotators are asked to directly provide calendar bounds on when each event happened (like *after 1992*, and *before 2000*). This way, the events can also be temporally related to calendar anchors that do not occur directly in the text as timex expressions, in contrast to the schemes mentioned earlier. Because this scheme annotates on the event level, and not on the event-pair level, its annotation time is linear. Also, when all events are linked to absolute calendar dates many TLinks can be automatically inferred, by exploiting the order on calendar days. This way a lot of temporal information can be captured by relatively little annotation. Although minor limitations of this scheme are the coarse granularity of the durations, and the fact that now relative ordering statements between events cannot be annotated directly, we believe the direction of annotating on the event level, and allowing non-explicit timeline anchors to be very promising.

We can see that in Allen algebra (and also in point-algebra) it is not possible to directly include quantitative statements about interval durations, even though people clearly express duration information, and have access to it through background knowledge to make temporal inferences. This makes it difficult to combine the duration datasets with the currently interval-based reasoning methods. Combining relative position information, and quantitative duration information during reasoning has - to our

[2]To obtain this background knowledge on event durations automatically, instead of having to annotate it explicitly, there has been work on extracting typical event durations from the web using lexico-syntactic patterns (Kozareva and Hovy, 2011; Williams and Katz, 2012).

knowledge - not been done in the TIE systems in the literature and could be very interesting for future research.

5.5 Temporal Reasoning for Temporal Information Extraction

In this section we will review the development of TIE systems, focusing on approaches that use some form of TR. We do not discuss extraction of events as generally this does not impact TR directly. Extensive research was done on event extraction and timex extraction resulting in strong extraction systems for these tasks (Strötgen and Gertz, 2010; Chang and Manning, 2012; Derczynski et al., 2012; Lee et al., 2014; Strötgen and Gertz, 2015; Miller et al., 2015; Bethard and Parker, 2016; Strötgen and Gertz, 2016; Derczynski et al., 2016; Laparra et al., 2018b,a; Olex et al., 2018). The normalization of timex, determining the calendar value of relative temporal expressions (e.g., *last year* $\rightarrow$ 2018), does involve TR. Almost all state-of-the-art systems resolve timex normalization successfully using hand-crafted rules based on lexical patterns (Mani and Wilson, 2000; Negri and Marseglia, 2004; Verhagen and Pustejovsky, 2008; Strötgen and Gertz, 2010; Kolomiyets and Moens, 2010; Chang and Manning, 2012; Llorens et al., 2012; Lin et al., 2013; Filannino et al., 2013; Sun et al., 2015; Mirza, 2015; Strötgen and Gertz, 2016; Derczynski et al., 2016; Real et al., 2018; Olex et al., 2018). We would like to highlight the scheme by Bethard and Parker (2016) for its explicit use of a more general TR method, called SCATE (Semantically Compositional Annotation Scheme for Temporal Expressions), in which the temporal value (interval or duration) of a timex is composed from the individual words of the timex through interval operations. How the words are to be composed is annotated in the corpus through links, which can also be predicted by a relation extraction model (Laparra et al., 2018b,a; Olex et al., 2018).

For the rest of this section we focus on the task of ordering events, which builds on top of event and timex extraction and normalization, and often involves extensive TR, as information on multiple events and temporal expressions has to be combined. For this task the state-of-the-art event ordering systems still perform below application level (Bethard et al., 2015, 2016, 2017; Ning et al., 2018a; Meng and Rumshisky, 2018). We will discuss the different ways in which TR can be exploited in the different steps of TIE model construction, shown in Figure 5.9, starting with annotation.

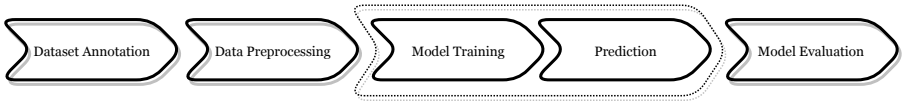

Figure 5.9: Steps for constructing temporal information extraction models. During each step temporal reasoning can be exploited.

5.5.1 Temporal Reasoning for Dataset Annotation (TR-DA)

A widely recognized problem in temporal annotation, is the fact that because of the complex nature of temporal information, annotators tend to miss temporal cues in the text, especially when annotating TLinks between pairs of events (Mani et al., 2006). Verhagen (2005), and Setzer et al. (2005) argue that when annotating temporal relations, not all event-pairs are equally useful to annotate, as some can be deduced from the already annotated TLinks. To save annotation effort they suggest to incorporate TR in the annotation tool, and only ask annotators to label event pairs for which it is not yet possible to infer a relation from the annotations already present. Another motivation to exploit TR during annotation is that annotators sometimes provide temporally inconsistent labelings, which makes it harder to train TIE systems (Gennari and Vittorini, 2016, 2017). Verhagen et al. (2006) used TR with basic Allen relations in their TANGO toolkit. Also, in the construction of the TimeBank-Dense (Cassidy et al., 2014), where annotators are forced to annotate temporal links between all entity pairs within a certain window, after each annotation a temporal closure is calculated. This way, the annotated corpus obtained interesting properties: (1) its graphs are strongly connected, (2) the resulting corpus is consistent, and (3) all required edges are labeled. As not all events can be related through basic Allen relations, the authors introduced a *vague* relation type for when two events are difficult to relate. These works show that TR can be very beneficial during data annotation, making the use of TR for dataset annotation a very interesting direction of research needed to construct better application-level TIE systems. Also the combination of TR with active learning (Settles, 2012) as a way to collect informative training examples might be very interesting, as temporal annotation is generally considered difficult and consequently time consuming (Boguraev and Ando, 2007; Tissot et al., 2015; Derczynski, 2016; Gennari and Vittorini, 2017).

5.5.2 Temporal Reasoning for Data Preprocessing (TR-DP)

Once a dataset has been annotated, a commonly used method is to expand the training data, as annotators frequently miss TLinks. This can be done by applying a transitive

closure to infer new TLink annotations based on the ones already present. The firstly used algorithm to calculate a temporal closure for this purpose is called *SputLink* (Verhagen, 2004). SputLink is based on a subset of Allen's interval algebra and uses a composition table of 745 axioms (or transitivity rules) to infer new relations. It was used already in one of the first machine learning approaches to temporal relation extraction by Mani et al. (2006). In their experiments, using SputLink increases the number of TLinks by a factor 11. The effect of temporal closure on performance for models that use no TR during prediction is ambivalent. There have been cases where temporal closure increases performance (Mani et al., 2006, 2007), but also where it does not (Chambers and Jurafsky, 2008b), or even decreases performance (Tatu and Srikanth, 2008). When using a model that exploits TR during prediction it has been shown that data expansion through a temporal closure can be very beneficial (Chambers and Jurafsky, 2008a). Models that use TR during prediction will be described in the next section.

5.5.3 Temporal Reasoning for Training or Prediction (TR-TP)

To enhance temporal relation extraction models, TR can be integrated during training, and during prediction. Since training and prediction are so closely related, and often prediction is a sub-procedure of training, we discuss the integration of TR in training and prediction together in a single section.

There are various reasons why integration of TR during training and prediction can improve TIE models: (1) ensure temporal consistency among the predicted relations, and (2) to constrain the output space and improve prediction accuracy.

Predicting consistent temporal information is very important for real applications, as it is not possible to construct a proper timeline from a set of inconsistent TLinks. To ensure consistency of TLinks, the expressiveness of TR is crucial. Computational efficiency of TR is also very important for real applications, especially when the TR needs to be performed for each prediction. So, the ideal TR method for a TIE application has a good trade-off between expressiveness and computational efficiency. Many methods have been explored for incorporating TR during model training and prediction, which we discuss one by one.

Best-First (BF) or Natural Reading Order (NRO) Greedy Inference

One of the first approaches to incorporate TR during prediction was proposed by Bramsen et al. (2006a). They addressed the task of temporally ordering text segments in clinical narratives by first detecting segment boundaries, and afterwards classifying each pair of segments into one of three temporal relations: BEFORE, AFTER, or

INCOMPARABLE. To ensure that the finally predicted temporal graph for each document is consistent, they employ a *best-first* (BF) greedy strategy: (1) All segment pairs are sorted by model confidence (based on the score predicted by a pairwise relation classifier), (2) In order of high-to-low confidence, relations are added one-by-one. After adding each relation a temporal closure is applied to expand the graph, until all pairs are connected.

A slight alteration is to order the relations not by model confidence in step 1, but by *natural reading order* (NRO). Afterwards, in step 2, the same strategy is applied as with BF: adding relations one-by-one followed by a transitive closure. Regarding performance, both NRO and BF outperform the baseline models that use no TR during prediction at all, and BF obtains the best results (Bramsen et al., 2006b).

Post-Hoc (P-HOC) Conflict Resolution

Many approaches first use greedy prediction to predict a temporal graph, and resolve conflicts post-hoc. These approaches involve the removal of conflict-causing edges, based on model confidence (Verhagen and Pustejovsky, 2008; Tatu and Srikanth, 2008; Cheng et al., 2013; Sun, 2014; Meng et al., 2017). Verhagen and Pustejovsky (2008) used different models to predict different parts of the temporal graph, and assign different confidences per model to remove the least confident conflict-causing TLinks. Cheng et al. (2013) resolve conflicts by removing the least confident conflict-causing edge for each within-sentence triangle of relations. Meng et al. (2017) remove conflict-causing TLinks such that the sum of their confidences is as low as possible. Although most works using post-hoc conflict resolution report a positive impact of the resolution, they have not been compared to each other in a quantitative manner. This could be an interesting comparison for future research.

Sieve-Level Inference (SLI) and Stacked Inference (SI)

A popular method for TIE is the sieve-based method (Chambers et al., 2014), as it is a flexible way to incorporate both rule-based and machine learning components. The idea of the sieve-based approach is that TLinks are extracted in different consecutive phases by different model components (or sieves). Each sieve extracts TLinks, using the original input text, and the outputs from earlier sieves. TR is incorporated into sieve-based TIE systems by taking a transitive closure on the extracted TLinks after applying each sieve. This way later sieves are prevented from assigning TLinks that are inconsistent with those extracted by earlier sieves. Another way to look at this is that the closure in fact makes the output space smaller after each sieve. Typically the sieves are ordered by precision, making the *sieve-level inference* (SLI) similar to the greedy BF approach, except relations that are not added one-by-one, but in groups,

i.e., sieve-by-sieve. Very similar approaches have been adopted by Mirza and Tonelli (2016) and McDowell et al. (2017). Mirza and Tonelli (2016) separately evaluated the effect of this reasoning approach, and report an increase in recall contributing to an increase in performance when including the sieve-closure.

Another approach that exploits TR in multiple prediction stages is the *stacked inference* (SI) approach of Laokulrat et al. (2014, 2015). In the first stage their model predicts TLinks using pairwise local logistic regression classifiers, as many approaches do. However, on top of that they apply a transitive closure. Then, in a second stage, they learn a new TLink classification model that additionally takes the predicted relations from stage 1, and their corresponding probabilities as input features. An example of a stage 2 feature used to predict a TLink for some given entity pair is the set of all phase-1 TLink-paths connecting the two events. This way their model can learn to predict TLinks in context of other TLinks, resulting in a learned inference procedure.

A recent similar SI approach was used by Meng and Rumshisky (2018), inspired by neural Turing machines (Graves et al., 2014). Instead of using a pre-trained pairwise logistic regression model as local pairwise model in stage 1, like Laokulrat et al. (2015), Meng and Rumshisky (2018) employ a Long Short-Term Neural (LSTM) network classifier (Hochreiter and Schmidhuber, 1997). In the second stage, to classify the TLink relation for a candidate pair, they use the surrounding TLink predictions from the pre-trained model as input features in their final model. Another difference with Laokulrat et al. (2015) is that Meng and Rumshisky (2018) train their two stages jointly.

So for both Laokulrat et al. (2015) and Meng and Rumshisky (2018), the second phase uses no explicit knowledge about temporal reasoning (like reasoning rules). The label-label associations are learned from the data. An advantage of this is that the model can explicitly learn to correct mistakes of the local pairwise model from stage 1, which improves model performance. A disadvantage is that there are no guarantees on consistency.

Random Restart Hill Climbing (RRHC)

A less conventional TR approach was used by McClosky and Manning (2012), who addressed the task of temporal knowledge base population (KBP) slot filling (Ji et al., 2010). Here the focus lies on finding the temporal bounds of a certain subset of events called *fluents*, which are semantic relations between entities that hold for a certain period of time, like *attends-school*, or *has-parent*. Their aim was to find the bounds of such events by detecting the following four types of TLinks (called *meta-relations* in the original paper) between the events and time expressions: BEGUN BY, ENDED BY, DURING (called START, END, and START AND END in the original paper), and a class UNRELATED. They employ local pairwise classifiers to obtain scores for each relation type, that are combined in a joint inference to obtain a globally consistent

prediction using random restart hill climbing (RRHC). They define a global scoring function to score each set of predictions, which takes into account temporal consistency, and also the local scores from the pairwise classifiers. They find good scoring solution by iterating through all TE pairs, and at each pair adding the TLink that results in the setting with the highest score. Since this procedure depends on the initial order through which they randomly restart this procedure ten times, and pick the setting that gives the highest score from the ten final settings. This approach has not yet been evaluated in a more elaborate TIE setting, using a wider range of event types and temporal relation types.

Integer Linear Programming (ILP)

Another, more widely used, method to combine locally predicted TLink scores into a consistent temporal graph is integer linear programming (ILP). This technique was first exploited for TR by Bramsen et al. (2006b). They experiment with two greedy inference strategies: (1) following reading order, and (2) by in order of confidence, as described in the previous section. They also experimented with exact inference using integer linear programming (ILP). To formulate the problem as an integer linear program, it should be represented as a linear objective possibly extended with a set of linear constraints. Solving an ILP is in principle NP-complete, however there exist many efficient (often approximate) solvers (Berkelaar et al., 2004; Makhorin, 2008; Gu et al., 2012). Bramsen et al. (2006b) formulate the objective as the sum of all scores of the pairwise classifiers (for BEFORE, AFTER, and INCOMPARABLE). Additionally, they model three constraints:

1. Each segment pair is assigned only one label. (**mutual exclusivity**)
2. The BEFORE and AFTER relations follow transitivity. (**transitivity**)
3. Each segment is connected through at least one edge other than INCOMPARABLE. (**connectivity**)

Their experimental results show that using ILP performs better than greedy approaches like BF and NRO. Similar results were obtained as well by Mirroshandel and Ghassem-Sani (2012).

Chambers and Jurafsky (2008a) use a similar approach using TimeML-style data, predicting the same TLink types as Bramsen et al. (2006b), and also modeling transitivity through ILP inference, but considering TLinks between events (EE-R) instead of between text segments.

They also show that using TR during prediction is more effective on densely connected temporal graphs. To obtain a densely connected temporal graph from the initially

sparsely annotated annotations, they apply an extensive temporal closure to infer new EE-R. To compute the closure, they also exploit other TLink types (e.g., INCLUDES, and SIMULTANEOUS) and also the other relation categories: TE-R, DCT-R, and TT-R, later also used by others Tatu and Srikanth (2008); Denis and Muller (2011); Laokulrat et al. (2015); Ning et al. (2018a). The importance of densely connected graphs for TR-MI was also recognized by later research (Denis and Muller, 2011; Do et al., 2012; Leeuwenberg and Moens, 2017b).

Both Bramsen et al. (2006b), and Chambers and Jurafsky (2008a) only used two TimeML relations in their joint inference, working on a restricted problem setting, compared to using all twelve TLink types. When taking into account all twelve TLink types TR becomes much more computationally complex. This complexity problem is addressed by Denis and Muller (2011). In contrast to earlier approaches, they propose to formulate TR-based prediction using point-algebra instead of Allen algebra, exploiting the mapping proposed by Vilain et al. (1990), shown earlier in the lattice of Figure 5.6. This is done by translating the TLinks between intervals into relations between start and end points ($>$, $<$, and $=$). Their ILP objective maximizes the score from local pairwise classifiers, similar to Bramsen et al. (2006b), except that the ILP decision variables correspond to decisions about the translated point-wise relations, resulting in four times less variables. Because there are also fewer point-wise relation types (three) compared to interval relation types (twelve), they need a factor fifty fewer constraints to model the same reasoning fragment. After solving the ILP, the resulting point-wise decisions are translated back to TimeML interval relations. This shows that exploiting the connection between interval and point space greatly increases the computational efficiency of TR-based prediction with ILP. We argue that this method is very important for practical TIE systems. Compared to earlier ILP formulations, computational efficiency is gained, while at the same time expressiveness is increased. The full conjunctive sub-fragment discussed in section 5.3 is covered instead of just covering a handful of basic transitivity rules as is often done. Interestingly, more recently Kerr et al. (2014) used a quite large set of transitivity rules, using ILP to construct an ensemble from many local pairwise TLink extraction models, showing clear improvements. Although they used a large set of transitivity rules, they did not exploit the efficient point-algebraic formulation by Denis and Muller (2011).

A second modification to save computation has to do with temporal reasoning on sub-groups of events, instead of all events in the document. The intuition is that people describe events in time frames, also called *narrative containers* (Pustejovsky and Stubbs, 2011), which has been adopted in later annotation works, and their corresponding systems (Styler IV et al., 2014; Bethard et al., 2015, 2016, 2017). This intuition of narrative containers is similar to the segments of Bramsen et al. (2006b), that correspond to larger phrases instead of individual events. Denis and Muller (2011) obtain these sub-graphs from ground-truth structures of connected events, and through events that correspond to the same temporal expression, and show how this can be used

for more efficient TR.

Markov Logic Networks (MLN)

Another important method that has been proposed for TR-based prediction are Markov Logic Networks (MLN) (Richardson and Domingos, 2006). These were first explored for TIE by Yoshikawa et al. (2009), and later also by Ling and Weld (2010), and Ha et al. (2010). An major difference between MLN and the inference methods mentioned earlier, is that instead of combining locally trained models in a global inference setting, MLN also exploit the temporal constraints during training. Also, the weights for TR constraints can be learned, allowing the model to also learn soft correlations between TLinks, instead of hard rules.

To set up a MLN for a discriminative prediction problem, like predicting TLinks between events, one has to: (1) define a set of hidden first-order predicates that are observable during training that you want to predict (e.g., TLinks between different events), (2) define a set of observed predicates, available at both training and test-time (e.g., event features), and (3) define association rules among the predicates, which will get assigned a weight (e.g., feature-label associations, and label-label associations, like transitivity of certain TLinks). And (4), once the MLN is defined, a training and inference regime has to be determined to estimate probabilities for the hidden predicates and the association rules. This last step is often provided by MLN interpreters (Niu et al., 2011). Yoshikawa et al. (2009) model different transitivity rules connecting the EE-R, TE-R, and DCT-R TLinks, and show that using MLN to incorporate TR outperforms local models that use no TR during prediction. MLN have been less popular in more recent works for scalability reasons (Leeuwenberg and Moens, 2017b; Mojica and Ng, 2016). MLN constraints are soft (predicate assignments can become less likely, but not impossible), in contrast to approaches that model hard constraints, like ILP, that allow cutting off large areas from the search space to find solutions efficiently.

Structured Perceptron with Integer Linear Programming (SP+ILP)

Another approach to exploit TR in both training and prediction was proposed by Abend et al. (2015), in the domain of cooking recipes. They learn a global model formulating TIE as a structured learning problem. For this they combine an averaged structured perceptron (Freund and Schapire, 1999) with ILP inference. Abend et al. (2015) focused only on precedence relations between events, which was sufficient for cooking recipes. Similar approaches but for more extensive TimeML-based relations were proposed by Leeuwenberg and Moens (2017b) in the clinical domain, and by Ning

et al. (2017)[3] in the news domain. In these structured learning approaches, a scoring function is learned that scores groups of TLink assignments rather than single TLink assignments, as local models do. Model inference then corresponds to finding the assignment of TLinks with the highest overall score by the model. The naive inference method is to enumerate all possible TLinks assignments for a document, and pick the assignment with the highest score, which is usually highly computationally intractable. To formulate a more efficient inference procedure ILP is used to constrain the search space, similar to the approaches mentioned in the previous section. During training of the structured perceptron, the same ILP-style inference is used.

There are a few differences between the three SP+ILP approaches: Abend et al. (2015) focused on precedence relations between the events only. For this reason, their objective was to find a single chain of relations in which each event is visited only once, whereas the other two works use a more extensive rule set, making inference more complex. Also, Leeuwenberg and Moens (2017b) besides hard-coded transitivity rules, also exploited soft learned label-label constraints. And, Ning et al. (2017) used an even more extensive transitivity table for TR, including more expressive rules that infer also some general Allen relations (disjunctions of TLinks), resulting in more expressive TR (class II). Similar to most approaches all works prune the total set of TLink candidate pairs to reduce computational complexity, and in all cases it was reported that TR during both training and prediction generally performs better compared to combining local classifiers with ILP.

Direct Timeline Models (DTLM)

Recently, a new type of approach to temporal event ordering was proposed by Leeuwenberg and Moens (2018a) (discussed in detail in Chapter 6). Instead of predicting TLinks among events and temporal expressions, their model directly predicts the start and end points of events. An advantage of this approach is that it is fast, as predicting start and end points for each event is linear in the number of events, in contrast to predicting a set of TLinks, which is quadratic or requires pruning. To train their model they exploit TR to convert TLinks to sets of point-algebraic constraints. The loss function to train the model represents the distance that start and end points of events still need to shift to make all annotated temporal order relations valid on the predicted timeline. Another advantage of this approach is that its predicted timelines are consistent by definition. The main limitation of the approach is that there is no probabilistic interpretation of confidence for predictions, which is mentioned as future work.

[3] Which was later extended to deal with sparse annotations (Ning et al., 2018d), jointly reason with causal relations (Ning et al., 2018a), and to include statistical knowledge from other resources (Ning et al., 2018b).

5.5.4 Temporal Reasoning during Model Evaluation (TR-ME)

Initially, temporal information extraction systems were evaluated using either accuracy or F1-measure of extracted TLinks. Setzer et al. (2003) proposed to exploit TR to address a problem of straightforward calculation of accuracy or F1-measure:

- The same temporal situation can be represented using different TLinks (e.g., A BEFORE B represents the same situation as B AFTER A with different labels).

To counter this problem they proposed to apply a temporal closure using all TLink types before evaluating F1-measure. However, this way, all TLinks are weighted equally, predicted and inferred TLinks. Tannier et al. (2008) addressed this problem by evaluating only with regard to *core relations*. From a set of TLinks, the core relations can be obtained by removing relations one-by-one, for as long as the inferable set of relations does not change (i.e., no information is lost). A problem with this approach however is that when comparing only core relations, not all inference information is captured, as already pointed out by Tannier et al. (2008) and Tannier and Muller (2011).

UzZaman and Allen (2011) have proposed a metric that deals with this issue, called temporal-awareness. They calculate a harmonic mean of precision and recall, i.e., an F-score. A crucial difference with Setzer et al. (2003), and Tannier et al. (2008) is that to calculate their precision and recall they do not modify the original relations, but rather change the criterion on whether a relation is correctly classified with regard to the reference or not, using TR. Their precision metric is calculated as the percentage of system relations that can be *verified* from the reference relations using TR. Recall is calculated as the percentage of reference relations that can be verified from the system relations, using TR. To perform TR, they exploit TimeGraph (Miller and Schubert, 1990), an efficient TR algorithm based on the mapping between intervals and point algebra mentioned in Figure 5.6. TimeGraph conducts TR in the non-disjunctive sub-fragment of Allen's algebra, i.e., expressivity class III. The temporal-awareness is now used widely for evaluating TIE systems.

5.5.5 Overview of TR in TIE

There are some striking differences between the usage of TR in the different phases of model construction. During annotation, TR has shown positive results, and can help to reduce annotation work, and ensure consistency in the annotations. However, exploiting TR is not yet common practice in corpus construction.

In Table 5.2, we construct an overview of - to our knowledge - all TIE systems described in the literature that employ some form of interval or point-based reasoning for event ordering in the past three decades. We can see in the overview that in

earlier approaches fewer TLink categories have been used for TR, focusing mostly on precedence relations (PR) between event-event pairs (EE-R). In later approaches more relation types are predicted, like temporal inclusion (IR), temporal equivalence (ER), and overlap relations (OR). This shows that the research focus in the community slowly grows in the direction of the challenge of full TIE, where *all* temporal cues from the text are extracted and combined at the same time, making TR an increasingly important aspect of TIE systems.

If we look at the use of TR to expand the training data (TR-DE) we observe mixed effects. Approaches that report high improvements above 10% improvement (Mani et al., 2006, 2007; Tatu and Srikanth, 2008) do so while splitting the training and test set on the relation instance level after TR. When splitting on the document-level, which is more realistic setting, the improvements are much smaller or even negative (Mani et al., 2007; Tatu and Srikanth, 2008; Nikfarjam et al., 2013; Mirza, 2014).

TR-based prediction approaches (TR-TP) are reported to outperform those that do not exploit TR, where ILP based approaches generally outperform greedy approaches (Bramsen et al., 2006b; Denis and Muller, 2011; Mirroshandel and Ghassem-Sani, 2012). A trend also seen in the table, is that more systems exploit TR not only during prediction (NRO, BF, ILP, SLI, RRHC), but also during training (SI, MLN, SP+ILP, DTLM), as this has shown to improve performance even further (Leeuwenberg and Moens, 2017b; Ning et al., 2017), stressing the importance of integration of TR in TIE models.

The expressivity of the TR-based prediction approaches (in column TR-TP) mostly concerns basic Allen relations (class I). Ning et al. (2017, 2018a) extend this to transitivity rules with disjunctions of Allen relations in the conclusion (class II), but they still perform reasoning with the interval-level transitivity table. The vast majority of TR approaches that go beyond Allen's composition table of basic relations (from class I or II to class III) in TR expressiveness almost all perform reasoning in point algebra to remain tractable. This is observed in all areas of TIE: data expansion (Verhagen, 2005), training and prediction (Denis and Muller, 2011; Leeuwenberg and Moens, 2018a), and for evaluation (UzZaman and Allen, 2011). This indicates the importance of exploiting the point-interval mapping when considering practical systems, where expressivity and efficiency are both important.

It can be seen that during evaluation the expressivity of TR is frequently of class III. This is because the temporal awareness metric by (UzZaman and Allen, 2011) was adopted in the TempEval challenges (UzZaman et al., 2013; Minard et al., 2015; Bethard et al., 2016, 2017), and hence became a standard evaluation metric to use.

Table 5.2: Overview of event ordering TIE systems using interval or point-based TR. The first column shows the reference, the second the types of temporal interval relations that are predicted: precedence relations (PR: *precedes*, *preceded by*, *meets*, *met by*) inclusion relations (IR: *during*, *contains*, *starts*, *started by*, *ends*, *ended by*), overlap relations (OR: *overlap*, *overlapped by*) or equivalence relations (ER: *equals*). The next three columns indicate whether TR was used for data expansion (TR-DE), during training or prediction (TR-TP), or during model evaluation (TR-ME), where roman numerals indicate the expressivity class from section 5.3. The last column shows among what types of entities the relations are predicted (from section 5.4), where † indicates if ground truth relations were used in TR that were not predicted. If a reference directly evaluated absence of TR, for each experiment (separated by /), we report baseline score(s), and the change in score due to TR. Scores and improvements are incomparable across references, as datasets, tasks, and evaluation metrics vary.

Reference	Relation Types	TR-DE	TR-TP	TR-ME	Candidate Pairs
Mani et al. (2006)	PR, IR, ER	✓ $^{76+11\%}$	-	-	EE-R, TE-R
Mani et al. (2007)	PR, IR, ER	✓ $^{60+14\%/-11\%}$	-	-	EE-R, TE-R
Bramsen et al. (2006a)	PR	✓	BF_I	-	SS-R
Bramsen et al. (2006b)	PR	✓	$NRO_I^{72+2\%}$, $BF_I^{+6\%}$, $ILP_I^{+12\%}$	-	SS-R
Chambers and Jurafsky (2008a)	PR	✓	$ILP_I^{72+2\%}$	-	EE-R, TE-R†, TT-R†
Verhagen and Pustejovsky (2008)	PR, IR, ER	✓	$P\text{-}HOC_I$	-	EE-R
Tatu and Srikanth (2008)	PR, IR, ER	✓ $^{57+1/+15\%}$	$P\text{-}HOC_I^{50-3\%}$	-	EE-R, TE-R†, TT-R†
Yoshikawa et al. (2009)	PR, IR, ER	-	$MLN_I^{67+2\%}$	-	EE-R, TE-R, DCT-R
Ling and Weld (2010)	PR, OR	-	MLN_I	-	TE-R
Ha et al. (2010)	PR, OR	-	MLN_I	-	EE-R, TE-R
Denis and Muller (2011)	PR, IR, ER	✓	$NRO_I^{11+14\%}$, $ILP_{III}^{+30\%}$	-	EE-R, TE-R†, TT-R†
Mirroshandel and Ghassem-Sani (2012)	PR, IR, ER	-	$BF_I^{48+3/49+1\%}$, $ILP_I^{+4/+2\%}$	-	EE-R
Do et al. (2012)	PR, OR	✓	ILP_I	-	EE-R, TE-R, DCT-R
McClosky and Manning (2012)	IR	-	$RRHC_I^{71+1\%}$	-	TE-R
Costa and Branco (2013)	PR, OR	✓	NRO_I	-	EE-R, TE-R, DCT-R, TT-R
Nikfarjam et al. (2013)	PR, OR	✓ $^{63+1\%}$	-	-	TE-R
Cheng et al. (2013)	PR, OR	✓	$P\text{-}HOC_I$	✓ III	EE-R,TE-R
Sun (2014)	PR, OR	-	$P\text{-}HOC_I$	✓ III	EE-R,TE-R
Chambers et al. (2014)	PR, IR, ER	-	$SLI_I^{32+8\%}$	-	EE-R, TE-R, DCT-R, TT-R
Mirza (2014)	PR, IR, ER	✓ $^{48-2\%}$	-	✓ III	EE-R, TE-R, DCT-R
Kerr et al. (2014)	PR,IR, ER, OR	-	ILP_{III}	✓ III	EE-R, TE-R, DCT-R
Laokulrat et al. (2015)	PR, IR, ER	✓	$SI_I^{69+1\%}$	✓ III	EE-R, TE-R, DCT-R, TT-R†
Abend et al. (2015)	PR, ER	-	$SP{+}ILP_I^{66+4/+5\%}$	✓ I	EE-R
Mirza and Tonelli (2016)	PR, IR, ER	✓	$SLI_I^{61+1/49+2\%}$	✓ III	EE-R, TE-R, DCT-R
Li et al. (2016)	PR, OR	-	$ILP_I^{62+2/+6\%}$	-	EE-R
Cornegruta and Vlachos (2016)	IR	-	-	✓ III	TE-R
Meng et al. (2017)	PR,IR,ER,OR	-	$P\text{-}HOC_I^{52+1/+4\%}$	✓ III	EE-R, TE-R, DCT-R, TT-R†
Leeuwenberg and Moens (2017b)	PR, IR, OR	✓	$ILP_I^{83+1\%}$, $SP{+}ILP_I^{+2\%}$	✓ III	EE-R, TE-R, DCT-R
Ning et al. (2017)	PR, IR, ER	✓	$ILP_{II}^{57+5\%}$, $SP{+}ILP_{II}^{+10\%}$	✓ III	EE-R, TE-R, DCT-R
McDowell et al. (2017)	PR, IR, ER	-	$SLI_I^{49+2\%}$	-	EE-R, TE-R, DCT-R
Ning et al. (2018a)	PR, IR, ER	✓	$SP{+}ILP_{II}^{46+4/+5\%}$	✓ III	EE-R, TE-R, DCT-R, TT-R†
Meng and Rumshisky (2018)	PR, IR, ER	-	$SI_I^{53+1\%}$	-	EE-R, TE-R, DCT-R, TT-R†
Leeuwenberg and Moens (2018a)	PR, IR, ER	-	$DTLM_{III}$	✓ III	EE-R, TE-R, DCT-R

5.6 Future Directions and Discussion

In this section we discuss the results from the survey and aim to point out areas that have been relatively unexplored or that we believe are promising for future work.

In the previous section, we observed that TR during prediction improves over using no TR, global methods outperform local greedy methods, and integrating TR in both prediction and training can improve model performance even further.

Efficiency of TR has not been compared explicitly across models in the existing TIE literature. However, it is used as motivation for many works to choose for a certain method. In general, greedy methods and sieve-based methods (BF, NRO, SI), which look for local optimal solutions, are faster than global methods like ILP, MLN, and RRHC, possibly at the cost of performance. There has been some work on comparing efficiency of ILP-style inference and MLN, which suggests that ILP is more efficient for currently existing solvers (Mojica and Ng, 2016). Also, SP+ILP methods are generally slower in training time than ILP-based methods, as their more complex inference procedure is also performed during training, however as seen in Table 5.2, it also often provides further improvements. To choose the degree of TR may depend on your dataset as well, as predicting densely connected graphs generally benefits more from TR. As mentioned in section 5.3, irrespective of the degree of TR, point-based TR methods are generally faster than equally expressive interval-based TR methods.

In this light, we observe that more recent works focus more on annotating and predicting relations between start and end points of events, rather than relations between events or intervals (Reimers et al., 2016, 2018; Ning et al., 2018c,e; Leeuwenberg and Moens, 2018a). We believe this is a promising and important change in perspective also with regard to TR. To increase expressivity of a TR component up to class III while remaining tractable reasoning in point-algebra is crucial. Additionally, representing the temporality of events by their start and end points provides flexibility when combining different annotation schemes, as most schemes can be converted to point-algebra.

Also, this flexibility could be very beneficial when incorporating other types of reasoning. Many questions that involves temporality can not be solved using TR alone, but require other types of semantic reasoning about events and entities (Höffner et al., 2017; Pampari et al., 2018; Suster and Daelemans, 2018), including (but not limited to) co-reference (Do et al., 2012), spatial reasoning, which has strong similarities with temporal reasoning (Guesgen, 1989; Mukerjee and Joe, 1990; Freksa, 1992b; Walsh, 2003), and causal reasoning and extraction (Bethard et al., 2008; Mirza and Tonelli, 2014; Mirza, 2014; Mirza and Tonelli, 2016; Mostafazadeh et al., 2016; Dunietz et al., 2017; Ning et al., 2018a).

To incorporate these different types of information into single TIE models we believe neural networks could be a suitable model class, as they are very flexible in combining

multiple tasks, and incorporating background knowledge like typical event orders (Chklovski and Pantel, 2004; Pichotta and Mooney, 2016), durations (Vempala et al., 2018), or times of the day (Noro et al., 2006). Currently, many neural TIE approaches follow the pairwise TLink classification paradigm, and are based on LSTM (Tourille et al., 2017b; Cheng and Miyao, 2017; Leeuwenberg and Moens, 2018b; Meng et al., 2017; Choubey and Huang, 2017; Lin et al., 2018, 2019), convolutional neural networks (CNN) (Dligach et al., 2017; Lin et al., 2017), tree-based LSTM networks (Galvan et al., 2018), and attention networks (Liu et al., 2019). However, currently the exploitation of TR for neural TIE has been very limited in neural TIE systems leaving room for future research.

Another area we would like to address is the challenging area of cross-document TIE. For TR in cross-document TIE temporal cues from multiple documents need to be combined, stressing the importance of computational efficiency. Cross-document TIE has not been discussed elaborately in this survey as the amount of TR in this research area has been very limited, possibly for this very reason of computational efficiency. Barzilay and McKeown (2005) were one of the first to dive into this area of research, and used *iconicity*, the heuristic that in narrative texts events are often mentioned in chronological order, to temporally order sentences in a multi-document summarization task. There has been work on generating course grained entity-focused timelines from multiple news articles as a means to do multi-document summarization (Yan et al., 2011; Zhao et al., 2013; Lin et al., 2014; Wang et al., 2015; Althoff et al., 2015). However, the focus of most of the research in this area lies mainly on the informativeness of sentences for the summary, and less on the temporal aspect. In most cases, TIE is very limited, and it is simply assumed that all events mentioned in each text occur at the document-creation time, leaving many opportunities for TIE and TR. Besides computational efficiency, a challenge in a cross-document TIE setting is that event mentions of the same event should be linked across documents, called cross-document event co-reference. Event co-reference is strongly connected to TIE as a single event can only occur at a single time. Do et al. (2012) modeled this principle connecting TIE and event co-reference in their TR-based prediction using ILP. However, this was in a within-document setting. This could be a good starting point for cross-document ILP formulations. Minard et al. (2015) provided a setup for evaluating this problem in a shared task. However, TR has not yet been explored by the participants of this shared task. Also the ECB+ corpus (Cybulska and Vossen, 2014), which contains event co-reference and temporal information annotations across documents could be an interesting resource.

With regard to general TIE, considering both cross-document but also within-document relations, we can observe from the overview in the previous section that the state-of-the-art systems using TR focus mainly on *ordering* events, and *anchoring* events to temporal expressions, using TimeML-style data. Since TimeML only annotates duration cues that are definite, quantified and explicit it does not contain implicit or

background event duration information, nor explicit indefinite quantification (*long meeting*). There have been TIE systems that also use implicit duration information (Pan et al., 2006b; Reimers et al., 2018). However, these systems do not exploit TR, nor TimeML-style annotations, leaving a gap in the literature: systems that are able to combine different data sources, like TimeML and event duration annotations (Pan et al., 2011; Reimers et al., 2016). Combining different types of information and learning from different data sources presents new challenges for TR. For the TR aspect it might be interesting to explore temporal constraint networks, that are able to deal with quantification (Dechter and Cohen, 2003). Two systems for which we believe further integration of event duration information would be straightforward are: (1) the direct timeline models by Leeuwenberg and Moens (2018a), and (2) the work of Reimers et al. (2018), as both approaches already predict and combine event position and duration.

5.7 Conclusions

We presented a comprehensive survey on how temporal reasoning mechanisms can be exploited for temporal information extraction, covering the literature of the last three decades in this research area.

To explain the complexity of the temporal information that is present in language we provided an exemplified overview of the different types of temporal information present in language: absolute v.s. relative cues, definite and indefinite cues, implicit cues, and background knowledge. Many types of temporal cues, like explicit event position cues, and event durations can already be extracted by different types of temporal information extraction systems from the literature.

There appears to be a trend towards more complete TR, going from only reasoning about EE-R, towards including also TE-R, DCT-R, and even TT-R. Although still no systems seem to combine all temporal cues as of yet. For example, information about the relative ordering of events and quantified duration information have not yet been combined in TR, although data is available, accommodating room for future work on joint models.

To provide a back-bone for the state-of-the-art TIE systems using TR, a comprehensive explanation of the most widely used temporal reasoning frameworks in temporal information extraction systems was given. We reviewed Allen's interval algebra and its relation to point algebra in detail, giving insight into the considerations with regard to expressiveness of temporal reasoning and computational efficiency. We argue that for obtaining practical systems point-algebraic approaches for TR are preferred, as they strike a good balance between expressiveness and efficiency instead of reasoning directly with Allen's interval relations.

In the core of the survey, we reviewed the different methods to exploit temporal reasoning for constructing temporal information extraction models and distilled the most widely confirmed conclusions. TR during annotation has been proposed already early on, and is effective for ensuring connectivity and consistency in annotations but has been used to a limited degree in existing corpora. To expand the often only sparsely annotated TimeML data, a transitive closure is used frequently to densify the annotated temporal graphs. This has been found mostly beneficial when TR is also used during model inference, as TR during model inference appears to work better on densely connected temporal graphs. Generally, usage of TR for model inference appears to increase model performance. Some approaches use TR only during training, and some both during training and inference. Inference-only approaches include: BF, ILP, RRHC, and SLI, and approaches that use TR also during training include: MLN, SI, SP+ILP, and DTLM. The last category of approaches has been researched most recently and has been reported to perform better than inference-only approaches. For evaluation the research community has converged on using the TR-based temporal awareness measure.

In closing, it is clear that TR is crucial for TIE, and widely used in all aspects of model construction. However, most current research on TIE still addresses sub-fragments of the complete TIE problem, focusing on extraction of specific types of temporal cues, instead of extracting all cues jointly which would allow them to complement each other. Consequently, it remains an open research question how to perform efficient and expressive TR involving all types of temporal cues. We believe to answer this question, a flexible, expressive and efficient reasoning framework is required. For this, we believe important directions of research are point-based reasoning approaches, striking a good balance between efficiency and expressiveness, and deep learning methods, that facilitate flexibility in model construction, multi-task learning, and sharing of representations.

6

Direct Prediction of Relative Timelines

This chapter was previously published as:

Artuur Leeuwenberg and Marie-Francine Moens. 2018. Temporal Information Extraction by Predicting Relative Timelines. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1237-1246, Brussels, Belgium. ACL.

Chapters 3 and 4 have focused on the extraction of temporal relations. In this chapter, we address the successive step of timeline construction, using the powerful link between the interval-based temporal relations and point-based reasoning, highlighted in the previous chapter (Chapter 5). We investigate how we can construct timelines from temporal graphs, but more interestingly whether we can train models from annotated temporal graphs that directly predict timeline positions for events, without extracting temporal relations as intermediate step. The motivation for this is that temporal relation extraction generally involves a squared complexity with regard to the number of events in the text.

This chapter introduces an indirect timeline construction method, two direct timeline models, and three loss functions to train the models. The approaches are empirically evaluated on two benchmark datasets in the news domain, and provide promising results, particularly for direct timeline prediction.

6.1 Introduction

The current leading perspective on temporal information extraction regards three phases: (1) a *temporal entity recognition* phase, extracting events (blue boxes in Fig. 6.1) and their attributes, and extracting temporal expressions (green boxes), and normalizing their values to dates or durations, (2) a *relation extraction* phase, where temporal links (TLinks) among those entities, and between events and the document-creation time (DCT) are found (arrows in Fig. 6.1, left). And (3), construction of a timeline (Fig. 6.1, right) from the extracted temporal links, if they are temporally consistent. Much research has concentrated on the first two steps, but very little research looks into step 3, timeline construction, which is the focus of this chapter.

In this chapter, we propose a new timeline construction paradigm that avoids phase 2, the relation extraction phase, because in the classical paradigm temporal relation extraction comes with many difficulties in training and prediction that arise from the fact that for a text with n temporal entities (events or temporal expressions) there are n^2 possible entity pairs, which makes it likely for annotators to miss relations, and makes inference slow as n^2 pairs need to be considered. Temporal relation extraction models consistently give lower performance than those in the entity recognition phase (UzZaman et al., 2013; Bethard et al., 2016, 2017), introducing errors in the timeline construction pipe-line.

The ultimate goal of our proposed paradigm is to predict from a text in which entities are already detected, for each entity: (1) a probability distribution on the entity's starting point, and (2) another distribution on the entity's duration. The probabilistic aspect is crucial for timeline based decision making. Constructed timelines allow for further quantitative reasoning with the temporal information, if this would be needed for certain applications.

As a first approach towards this goal, in this chapter, we propose several initial timeline models in this paradigm, that directly predict - in a linear fashion - start points and durations for each entity, using text with annotated temporal entities as input (shown in Fig. 6.1). The predicted start points and durations constitute a *relative timeline*, i.e., a total order on entity start and end points. The timeline is relative, as start and duration values cannot (yet) be mapped to absolute calender dates or durations expressed in seconds. It represents the relative temporal order and inclusions that temporal entities have with respect to each other by the quantitative start and end values of the entities. Relative timelines are a first step toward our goal, building models that predict statistical absolute timelines. To train our relative timeline models, we define novel loss functions that exploit TimeML-style annotations, used in most existing temporal corpora.

This chapter leads to the following contributions:

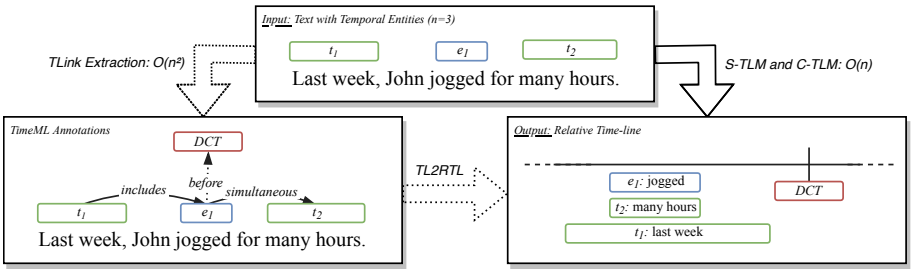

Figure 6.1: An overview of two paradigms: (1) The indirect approach (dashed arrows), where first TLinks are predicted from which we can build a relative timeline using TL2RTL. And (2), the direct approach (solid arrow), where a relative timeline is predicted directly from the input by S-TLM or C-TLM.

- A new method to construct a relative timeline from a set of temporal relations (TL2RTL).
- Two new models that, for the first time, directly predict (relative) timelines - in linear complexity - from entity-annotated texts without doing a form of temporal relation extraction (S-TLM & C-TLM).
- Three new loss functions based on the mapping between Allen's interval algebra and the end-point algebra to train timeline models from TimeML-style annotations.

In the next sections we will further discuss the related work on temporal information extraction. We will describe the models and training losses in detail, and report on conducted experiments.

6.2 Related Work

6.2.1 Temporal Information Extraction

The way temporal information is conveyed in language has been studied for a long time. It can be conveyed directly through verb tense, explicit temporal discourse markers (e.g. *during* or *afterwards*) (Derczynski, 2017) or temporal expressions such as dates, times or duration expressions (e.g. *10-05-2010* or *yesterday*). Temporal information is also captured in text implicitly, through background knowledge about, for example,

duration of events mentioned in the text (e.g. even without context, *walks* are usually shorter than *journeys*).

Most temporal corpora are annotated with TimeML-style annotations, of which an example is shown in Fig 6.1, indicating temporal entities, their attributes, and the TLinks among them.

The automatic extraction of TimeML-style temporal information from text using machine learning was first explored by Mani et al. (2007). They proposed a multinomial logistic regression classifier to predict the TLinks between entities. They also noted the problem of missed TLinks by annotators, and experimented with using temporal reasoning (temporal closure) to expand their training data.

Since then, much research focused on further improving the pairwise classification models, by exploring different types of classifiers and features, such as (among others) logistic regression and support vector machines (Bethard, 2013; Lin et al., 2016a), and different types of neural network models, such as long short-term memory networks (LSTM) (Tourille et al., 2017a; Cheng and Miyao, 2017), and convolutional neural networks (CNN) (Dligach et al., 2017). Moreover, different sieve-based approaches were proposed (Chambers et al., 2014; Mirza and Tonelli, 2016), facilitating mixing of rule-based and machine learning components.

Two major issues shared by these existing approaches are: (1) models classify TLinks in a pairwise fashion, often resulting in an inference complexity of $O(n^2)$, and (2) the pair-wise predictions are made independently, possibly resulting in prediction of temporally inconsistent graphs. To address the second, additional temporal reasoning can be used at the cost of computation time, during inference (Chambers and Jurafsky, 2008a; Denis and Muller, 2011; Do et al., 2012), or during both training and inference (Yoshikawa et al., 2009; Laokulrat et al., 2015; Ning et al., 2017; Leeuwenberg and Moens, 2017b). In this chapter, we circumvent these issues, as we predict timelines - in linear time complexity - that are temporally consistent by definition.

6.2.2 Temporal Reasoning

Temporal reasoning plays a central role in temporal information extraction, and there are roughly two approaches: (1) Reasoning directly with Allen's interval relations (shown in Table 6.1), by constructing rules like: If event X occurs before Y, and event Y before Z then X should happen before Z (Allen, 1983). Or (2), by first mapping the temporal interval expressions to expressions about interval end-points (start and endings of entities) (Vilain et al., 1990). An example of such mapping is that If event X occurs before Y then the end of X should be before the start of Y. Then reasoning can be done with end-points in a point algebra, which has only three point-wise relations

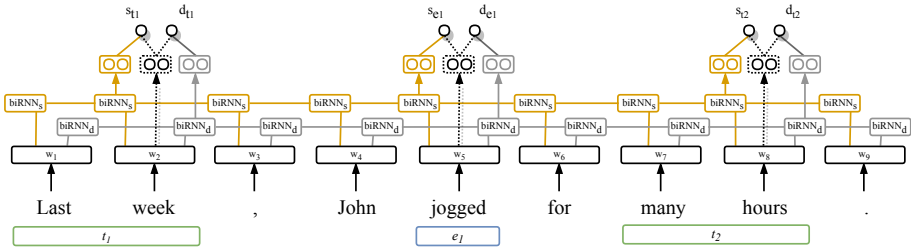

Figure 6.2: Schematic overview of our two timeline models: C-TLM (solid edges), exploiting entity context, and the simpler S-TLM (dotted edges), which is context independent. The models predict a starting point (s) and duration (d) for each given temporal entity (t_1, e_1, and t_2) in the input.

$(=, <, >)$, making reasoning much more efficient compared to reasoning with Allen's thirteen interval relations.

Mapping interval relations to point-wise expressions has been exploited for model inference by Denis and Muller (2011), and for evaluation by UzZaman and Allen (2011). In this chapter, we exploit it for the first time for model training, in our loss functions.

6.3 Models

We propose two model structures for direct timeline construction: (1) a simple context-independent model (S-TLM), and (2) a contextual model (C-TLM). Their structures are shown in Fig. 6.2. Additionally, we propose a method to construct relative timelines from a set of (extracted) TLinks (TL2RTL). In this section we first explain the first two direct models S-TLM and C-TLM, and afterwards the indirect method TL2RTL.

6.3.1 Direct Timeline Models

Word representation

In both S-TLM and C-TLM, words are represented as a concatenation of a word embedding, a POS embedding, and a Boolean feature vector containing entity attributes such as the type, class, aspect, following Do et al. (2012). Further details on these are given in the experiments section.

Simple Timeline Model (S-TLM)

For the simple context-independent timeline model, each entity is encoded by the word representation of the last word of the entity (generally the most important). From this representation we have a linear projection to the duration d, and the start s. S-TLM is shown by the dotted edges in Fig 6.2. An advantage of S-TLM is that it has very few parameters, and each entity can be placed on the timeline independently of the others, allowing parallelism during prediction. The downside is that S-TLM is limited in its use of contextual information.

Contextual Timeline Model (C-TLM)

To better exploit the entity context we also propose a contextual timeline model C-TLM (solid edges in Fig 6.2), that first encodes the full text using two bi-directional recurrent neural networks, one for entity starts (BiRNN$_s$), and one for entity durations (BiRNN$_d$).[1] On top of the encoded text we learn two linear mappings, one from the BiRNN$_d$ output of the last word of the entity mention to its duration d, and similarly for the start time, from the BiRNN$_s$ output to the entity's start s.

Predicting Start, Duration, and End

Both proposed models use linear mappings[2] to predict the start value s_i and duration d_i for the encoded entity i. By summing start s_i and duration d_i we can calculate the entity's end-point e_i.

$$e_i = s_i + \max(d_i, d_{min}) \tag{6.1}$$

Predicting durations rather than end-points makes it easy to control that the end-point lies after the start-point by constraining the duration d_i by a constant minimum duration value d_{min} above 0, as shown in Eq. 6.1.

Modeling Document-Creation Time

Although the DCT is often not found explicitly in the text, it is an entity in TimeML, and has TLinks to other entities. We model it by assigning it a text-independent start s_{DCT} and duration d_{DCT}.

[1] We also experimented with sharing weights among BiRNN$_d$ and BiRNN$_s$. In our experiments, this gave worse performance, so we propose to keep them separate.

[2] Adding more layers did not improve results.

Start s_{DCT} is set as a constant (with value 0). This way the model always has the same reference point, and can learn to position the entities w.r.t. the DCT on the timeline.

In contrast, DCT duration d_{DCT} is modeled as a single variable that is learned (initialized with 1). Since multiple entities may be included in the DCT, and entities have a minimum duration d_{min}, a constant d_{DCT} could possibly prevent the model from fitting all entities in the DCT. Modeling d_{DCT} as a variable allows growth of d_{DCT} and averts this issue.[3]

Training Losses

We propose three loss functions to train timeline models from TimeML-style annotations: a regular timeline loss L_τ, and two slightly expanded discriminative timeline losses, $L_{\tau ce}$ and $L_{\tau h}$.

Regular Timeline Loss (L_τ)

Ground-truth TLinks can be seen as constraints on correct positions of entities on a timeline. The regular timeline loss L_τ expresses the degree to which these constraints are met for a predicted timeline. If all TLinks are satisfied in the timeline for a certain text, L_τ will be 0 for that text.

As TLinks relate entities (intervals), we first convert the TLinks to expressions that relate the start and end points of entities. How each TLink is translated to its corresponding point-algebraic constraints is given in Table 6.1, following Allen (1983).

As can be seen in the last column there are only two point-wise operations in the point-algebraic constraints: an order operation ($<$), and an equality operation ($=$). To model to what degree each point-wise constraint is met, we employ hinge losses, with a margin m_τ, as shown in Eq. 6.2.

To explain the intuition and notation: If we have a point-wise expression ξ of the form $x < y$ (first case of Eq. 6.2), then the predicted point $\hat{x}$ should be at least a distance m_τ smaller (or earlier on the timeline) than predicted point $\hat{y}$ in order for the loss to be 0. Otherwise, the loss represents the distance $\hat{x}$ or $\hat{y}$ still has to move to make $\hat{x}$ smaller than $\hat{y}$ (and satisfy the constraint). For the second case, if ξ is of the form $x = y$, then point $\hat{x}$ and $\hat{y}$ should lie very close to each other, i.e., at most a distance m_τ away from each other. Any distance further than the margin m_τ is counted as loss. Notice that

[3] Other combinations of modeling s_{DCT} and d_{DCT} as variable or constant decreased performance.

[4] No TLink for Allen's overlap relation is present in TimeML, also concluded by UzZaman and Allen (2011).

Table 6.1: Point algebraic interpretation (I_{PA}) of temporal links used to construct the loss function. The start and end points of event X are indicated by s_x and e_x respectively.

Allen Algebra	**Temporal Links**	**Point Algebra (I_{PA})**
X precedes Y Y preceded by X	X before Y Y after X	$e_x < s_y$
X starts Y Y started by X	X begins Y Y begun by X	$s_x = s_y$ $e_x < e_y$
X finishes Y Y finished by X	X ends Y Y ended by X	$e_x = e_y$ $s_y < s_x$
X during Y Y includes X	X is included in Y Y includes X	$s_y < s_x$ $e_x < e_y$
X meets Y Y met by X	X immediately before Y Y immediately after X	$e_x = s_y$
X overlaps Y Y overlapped by X	absent[4] absent[4]	$s_x < s_y$ $s_y < e_x$ $e_x < e_y$
X equals Y	X simultaneous Y X identity Y	$s_x = s_y$ $e_x = e_y$

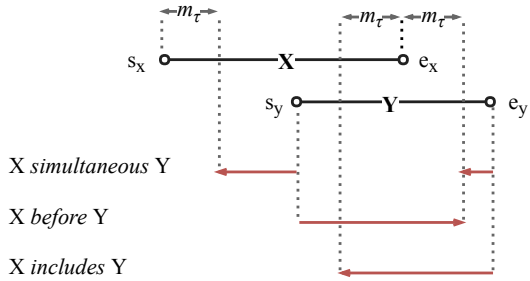

Figure 6.3: Visualization of the timeline loss L_τ with margin m_τ, for two events X and Y, and TLinks *simultaneous*, *before*, and *includes*. The red arrows' lengths indicate the loss per relation, i.e., how much the points should be shifted to satisfy each relation.

if we set margin m_τ to 0, the second case becomes an L1 loss $|\hat{x} - \hat{y}|$. However, we use a small margin m_τ to promote some distance between ordered points and prevent confusion with equality. Fig. 6.3 visualizes the loss for three TLinks.

$$L_p(\xi|t,\theta) = \begin{cases} \max(\hat{x} + m_\tau - \hat{y}, 0) & \text{iff } x < y \\ \max(|\hat{x} - \hat{y}| - m_\tau, 0) & \text{iff } x = y \end{cases} \tag{6.2}$$

The total timeline loss $L_\tau(t|\theta)$ of a model with parameters θ on text t with ground-truth TLinks $R(t)$, is the sum of the TLink-level losses of all TLinks $r \in R(t)$. Each TLink-level loss $L_r(r|t,\theta)$ for TLink r is the sum of the point-wise losses $L_p(\xi|t,\theta)$ of the corresponding point-algebraic constraints $\xi \in I_{PA}(r)$ from Table 6.1.[5]

$$L_r(r|t,\theta) = \sum_{\xi \in I_{PA}(r)} L_p(\xi|t,\theta) \tag{6.3}$$

$$L_\tau(t,\theta) = \sum_{r \in R(t)} L_r(r|t,\theta) \tag{6.4}$$

Discriminative Timeline Losses

To promote a more explicit difference between the relations on the timeline we introduce two discriminative loss functions, $L_{\tau ce}$ and $L_{\tau h}$, which build on top of L_r. Both discriminative loss functions use an intermediate score $S(r|t,\theta)$ for each TLink r based on the predicted timeline. As scoring function, we use the negative L_r loss, as shown in Eq. 6.5.

$$S(r|t,\theta) = -L_r(r|t,\theta) \tag{6.5}$$

Then, a lower timeline loss $L_r(r|t,\theta)$ results in a higher score for relation type r. Notice that the maximum score is 0, as this is the minimum L_r.

Probabilistic Loss ($L_{\tau ce}$)

Our first discriminative loss is a cross-entropy based loss. For this the predicted scores are normalized using a softmax over the possible relation types (TL). The resulting probabilities are used to calculate a cross-entropy loss, shown in Eq. 6.6. This way, the loss does not just promote the correct relation type but also distantiates from the other relation types.

$$L_{\tau ce}(t|\theta) = \sum_{r \in R(t)} r \cdot \log\left(\frac{e^{S(r|t,\theta)}}{\sum_{r' \in TL} e^{S(r'|t,\theta)}}\right) \tag{6.6}$$

[5]The TLink *during* and its inverse are mapped to *simultaneous*, following the evaluation of TempEval-3.

Ranking Loss ($L_{\tau h}$)

When interested in discriminating relations on the timeline, we want the correct relation type to have the highest score from all possible relation types TL. To represent this perspective, we also define a ranking loss with a score margin m_h in Eq. 6.7.

$$L_{\tau h}(t|\theta) = \sum_{r \in R(t)} \sum_{r' \in TL \setminus \{r\}} \max(S(r'|t,\theta) - S(r|t,\theta) + m_h, 0) \quad (6.7)$$

Training Procedure

S-TLM and C-TLM are trained by iterating through the training texts, sampling mini-batches of 32 annotated TLinks. For each batch we (1) perform a forward pass, (2) calculate the total loss (for one of the loss functions), (3) derive gradients using Adam[6] (Kingma and Ba, 2014), and (4) update the model parameters θ via back-propagation. After each epoch we shuffle the training texts. As the stopping criteria we use early stopping (Morgan and Bourlard, 1990), with a patience of 100 epochs and a maximum number of 1000 epochs.

6.3.2 From TLinks to Relative Timelines (TL2RTL)

To model the indirect route, we construct a novel method, TL2RTL, that predicts relative time lines from a subset of TLinks, shown in Fig 6.1. One can choose any method to obtain a set of TLinks $R(t)$ from a text t, serving as input to TL2RTL. TL2RTL constructs a relative timeline, by assigning start and end values to each temporal entity, such that the resulting timeline satisfies the extracted TLinks $R(t)$ by minimizing a loss function that is 0 when the extracted TLinks are satisfied. TL2RTL on itself is a method and not a model. The only variables over which it optimizes the loss are the to be assigned starts and duration values.

In detail, for a text t, with annotated entities $E(t)$, we first extract a set of TLinks $R(t)$. In this chapter, to extract TLinks, we use the current state-of-the-art structured TLink extraction model by Ning et al. (2017). Secondly, we assign a start variable s_i, and duration variable d_i to each entity $i \in E(t)$. Similar to S-TLM and C-TLM, for each $i \in E(t)$, d_i is bounded by a minimum duration d_{min} to ensure start s_i always lies before end e_i. Also, we model the DCT start s_{DCT} as a constant, and its duration d_{DCT} as a variable. Then we minimize one of the loss functions L_τ, $L_{\tau ce}$, or $L_{\tau h}$ on the extracted TLinks $R(t)$, obtaining three TL2RTL variants, one for each loss.

[6] Using the default parameters from the paper.

If the initially extracted set of TLinks $R(t)$ is consistent, and the loss is minimized sufficiently, all s_i and d_i form a relative timeline that satisfies the TLinks $R(t)$, but from which we can now also derive consistent TLinks for any entity pair, also the pairs that were not in $R(t)$. To minimize the loss we use Adam for 10k epochs until the loss is zero for each document.[7]

6.4 Experiments

6.4.1 Evaluation and Data

Because prediction of relative timelines trained on TimeML-style annotations is new, we cannot compare our model directly to relation extraction or classification models, as the latter do not provide completely temporally consistent TLinks for all possible entity pairs, like the relative timelines do. Neither can we compare directly to existing absolute timeline prediction models such as Reimers et al. (2018) because they are trained on different data with a very different annotation scheme.

To evaluate the quality of the relative timeline models in a fair way, we use TimeML-style test sets as follows: (1) We predict a timeline for each test-text, and (2) we check for all ground-truth annotated TLinks that are present in the data, what would be the derived relation type based on the predicted timeline, which is the relation type that gives the lowest timeline loss L_r. This results in a TLink assignment for each annotated pair in the TimeML-style reference data, and therefor we can use similar metrics. As evaluation metric we employ the temporal awareness metric, used in TempEval-3, which takes into account temporal closure (UzZaman et al., 2013). Notice that although we use the same metric, comparisons against relation classification systems would be unfair, as our model assigns consistent labels to all pairs, whereas relation classification systems do not.

For training and evaluation we use two data splits, TE‡ and TD‡, exactly following Ning et al. (2017). Some statistics about the data are shown in Table 6.2.[8] The splits are constituted from various smaller datasets: the TimeBank (TB) (Pustejovsky et al., 2003b), the AQUANT dataset (AQ), and the platinum dataset (PT) all from TempEval-3 (UzZaman et al., 2013). And, the TimeBank Dense (Chambers et al., 2014) , and the Verb-Clause dataset (VC) (Bethard et al., 2007).

[7] For some documents the extracted TLinks were temporally inconsistent, resulting in a non-zero loss. Nevertheless, $> 96\%$ of the extracted TLinks were satisfied.

[8] We explicitly excluded all test documents from training as some corpora annotated the same documents.

Table 6.2: Dataset splits used for evaluation (indicated with ‡).

Split	Training data	#TLinks	#Documents	Test data	#TLinks	#Documents
TD‡	TD (train+dev)	4.4k	27	TD (test)	1.3k	9
TE3‡	TB, AQ, VC, TD (full)	17.5k	256	PT	0.9k	20

6.4.2 Hyper-parameters and Preprocessing

Hyper-parameters shared in all settings can be found in Table 6.3. The following hyper-parameters are tuned using grid search on a development set (union of TB and AQ): d_{min} is chosen from $\{1, 0.1, 0.01\}$, m_τ from $\{0, 0.025, 0.05, 0.1\}$, α_d from $\{0, 0.1, 0.2, 0.4, 0.8\}$, and α_{rnn} from $\{10, 25, 50\}$. We use LSTM (Hochreiter and Schmidhuber, 1997) as RNN units[9] and employ 50-dimensional GloVe word-embeddings pre-trained[10] on 6B words (Wikipedia and NewsCrawl) to initialize the models' word embeddings.

Table 6.3: Hyper-parameters from the experiments.

Hyper-parameter	Value
Document-creation starting time (s_{DCT})	0
Minimum event duration (d_{min})	0.1
Timeline margin (m_τ)	0.025
Hinge loss margin (m_h)	0.1
Dropout (α_d)	0.1
Word-level RNN units (α_{rnn})	25
Word-embedding size (α_{wemb})	50
POS-embedding size	10

We use very simple tokenization and consider punctuation[11] or newline tokens as individual tokens, and split on spaces. Additionally, we lowercase the text and use the Stanford POS Tagger (Toutanova et al., 2003) to obtain POS.

6.5 Results

We compared our three proposed models for the three loss functions L_τ, $L_{\tau ce}$, and $L_{\tau h}$, and their linear (unweighted) combination L_*, on TE3‡ and TD‡, for which the results are shown in Table 6.4.

[9] We also experimented with GRU as RNN type, obtaining similar results.

[10] https://nlp.stanford.edu/projects/glove

[11] , . / \ " ' = + − ; : ( ) ! ? < > % & $ * | [ ] { }

Table 6.4: Evaluation in terms of precision (P), recall (R), and F1-measure (F) of relative timelines for each model and loss function, where L_* indicates the (unweighted) sum of L_τ, $L_{\tau ce}$, and $L_{\tau h}$.

	TE3‡			TD‡		
Model	**P**	**R**	**F**	**P**	**R**	**F**
Indirect: $O(n^2)$						
TL2RTL (L_τ)	53.5	51.1	52.3	59.1	61.2	60.1
TL2RTL ($L_{\tau ce}$)	53.9	51.7	**52.8**	61.2	60.7	60.9
TL2RTL ($L_{\tau h}$)	52.8	51.1	51.9	57.9	60.6	59.2
TL2RTL (L_*)	52.6	52.0	52.3	62.3	62.3	**62.3**
Direct: $O(n)$						
S-TLM (L_τ)	50.1	50.4	50.2	57.8	59.5	**58.6**
S-TLM ($L_{\tau ce}$)	50.1	50.0	50.1	53.4	53.5	53.5
S-TLM ($L_{\tau h}$)	51.5	51.7	**51.6**	55.1	56.4	55.7
S-TLM (L_*)	50.9	51.0	51.0	56.5	55.3	55.9
C-TLM (L_τ)	56.2	56.1	**56.1**	57.1	59.7	**58.4**
C-TLM ($L_{\tau ce}$)	54.4	55.4	54.9	52.4	57.3	54.7
C-TLM ($L_{\tau h}$)	55.7	55.5	55.6	55.3	54.9	55.1
C-TLM (L_*)	54.0	54.3	54.1	54.6	53.5	54.1

A trend that can be observed is that overall performance on TD‡ is higher than that of TE3‡, even though less documents are used for training. We inspected why this is the case, and this is caused by a difference in class balance between both test sets. In TE3‡ there are many more TLinks of type *simultaneous* (12% versus 3%), which are very difficult to predict, resulting in lower scores for TE3‡ compared to TD‡. The difference in performance between the datasets is probably also be related to the dense annotation scheme of TD‡ compared to the sparser annotations of TE3‡, as dense annotations give a more complete temporal view of the training texts. For TL2RTL better TLink extraction[12] is also propagated into the final timeline quality.

If we compare loss functions L_τ, $L_{\tau ce}$, and $L_{\tau h}$, and combination L_*, it can be noticed that, although all loss functions seem to give fairly similar performance, L_τ gives the most robust results (never lowest), especially noticeable for the smaller dataset TD‡. This is convenient, because L_τ is the fastest to compute during training, as it requires no score calculation for each TLink type. L_τ is also directly interpretable on the timeline. The combination of losses L_* shows mixed results, and has lower performance for S-TLM and C-TLM, but better performance for TL2RTL. However, it is slowest to compute, and less interpretable[13], as it is a combined loss.

[12] F1 of 40.3 for TE3‡ and 48.5 for TD‡ (Ning et al., 2017)

[13] The combined loss does not directly represent the distance on the timeline that points still have to shift to satisfy all TLinks, as L_τ does.

Moreover, we can see that on TE3‡, C-TLM performs better than the indirect models across all loss functions ($p = 0.07$[14]). This is a very interesting result, as C-TLM is an order of complexity faster in prediction speed compared to the indirect models ($O(n)$ compared to $O(n^2)$ for a text with n entities).[15] We further explore why this is the case through our error analysis in the next section.

	B	A	II	I	S
B	24.8%	4.7%	2.8%	1.6%	0.1%
A	5.0%	15.8%	3.2%	0.5%	0.0%
II	3.2%	3.2%	13.0%	0.6%	0.1%
I	4.0%	1.2%	1.0%	3.2%	0.0%
S	4.4%	3.0%	2.6%	1.3%	0.4%

	B	A	II	I	S
B	23.0%	8.2%	1.3%	0.9%	0.8%
A	4.7%	17.1%	1.8%	0.3%	0.5%
II	4.3%	4.4%	11.1%	0.4%	0.0%
I	1.6%	5.4%	0.5%	1.3%	0.5%
S	4.3%	4.1%	1.8%	0.6%	0.9%

Figure 6.4: On the **left**, the confusion matrix of C-TLM (L_τ), and on the **right** of TL2RTL ($L_{\tau ce}$), on TE3‡ for the top-5 most-frequent TLinks (together 95% of data): BEFORE (B), AFTER (A), IS INCLUDED (II), INCLUDES (I), and SIMULTANEOUS (S). Predictions are shown on the x-axis and ground-truth on the y-axis.

On TD‡, the indirect models seem to perform slightly better ($p = 0.12$). We suspect that the reason for this is that C-TLM has more parameters (mostly the LSTM weights), and thus requires more data (TD‡ has much fewer documents than TE3‡) compared to the indirect methods. Another result supporting this hypothesis is the fact that the difference between C-TLM and S-TLM is small on the smaller TD‡, indicating that C-TLM does not yet utilize contextual information from this dataset, whereas, in contrast, on TE3‡, the larger dataset, C-TLM significantly ($p < 0.01$) outperforms S-TLM across all loss functions, showing that when enough data is available C-TLM learns good LSTM weights that exploit context substantially.

6.6 Error Analysis

We compared predictions of TL2RTL(L_τ) with those of C-TLM (L_τ), the best models of each paradigm. In Table 6.4, we show the confusion matrices of both systems on TE3‡.

When looking at the overall pattern in errors, both models seem to make similar confusions on both datasets (TD‡ was excluded for space constraints).

Overall, we find that *simultaneous* is the most violated TLink for both models. This can be explained by two reasons: (1) It is the least frequent TLink in both datasets. And

[14] We calculate p using a document-level paired t-test.

[15] We do not directly compare prediction speed, as it would result in unfair evaluation because of implementation differences. However, currently, C-TLM predicts at ∼100 w/s incl. POS tagging, and ∼2000 w/s without. When not using POS, overall performance decreases consistently with 2-4 points.

Table 6.5: Example events from the top-shortest/longest durations and top-earliest/latest start values assigned by the model.

Short d	**Long** d	**Early** s	**Late** s
started	going	destroyed	realize
meet	expects	finished	bring
entered	recession	invaded	able
told	war	pronounced	got
arrived	support	created	work
allow	make	took	change
send	think	appeared	start
asked	created	leaving	reenergize

(2), simultaneous entities are often co-referring events. Event co-reference resolution is a very difficult task on its own.

We also looked at the average token-distance between arguments of correctly satisfied TLinks by the timelines of each model. For TL2RTL (L_τ) this is 13 tokens, and for C-TLM (L_τ) 15. When looking only at the TLinks that C-TLM (L_τ) satisfied and TL2RTL (L_τ) did not, the average distance is 21. These two observations suggest that the direct C-TLM (L_τ) model is better at positioning entities on the timeline that lie further away from each other in the text. An explanation for this can be error propagation of TLink extraction to the timeline construction, as the pairwise TLink extraction of the indirect paradigm extracts TLinks in a contextual window, to prune the $O(n^2)$ number of possible TLink candidates. This consequently prevents TL2RTL to properly position distant events with respect to each other.

To get more insight in what the model learns we calculated mean durations and mean starts of C-TLM (L_τ) predictions. Table 6.5 contains examples from the top-shortest, and top-longest duration assignments and earliest and latest starting points. We observe that events that generally have more events included are assigned longer duration and vice versa. And, events with low start values are in the past tense and events with high start values are generally in the present (or future) tense.

6.7 Discussion

A characteristic of our model is that it assumes that all events can be placed on a single timeline, and that it does not assume that unlabeled pairs are temporally unrelated. This has big advantages: it results in fast prediction, and missed annotation do not act as noise to the training, as they do for pairwise models. Ning et al. (2018c) argue

that actual, negated, hypothesized, expected or opinionated events should possibly be annotated on separate time-axis. We believe such multi-axis representations can be inferred from the generated single timelines if hedging information is recognized.

6.8 Conclusions

This chapter leads to the following three main contributions[16]: (1) Three new loss functions that connect the interval-based TimeML-annotations to points on a timeline, (2) A new method, TL2RTL, to predict relative timelines from a set of predicted temporal relations. And (3), most importantly, two new models, S-TLM and C-TLM, that – to our knowledge for the first time – predict (relative) timelines in linear complexity from text, by evading the computationally expensive (often $O(n^2)$) intermediate relation extraction phase in earlier work. From our experiments, we conclude that the proposed loss functions can be used effectively to train direct and indirect relative timeline models, and that, when provided enough data, the – much faster – direct model C-TLM outperforms the indirect method TL2RTL.

As a direction for future work, it would be very interesting to extend the current models, diving further into direct timeline models, and learn to predict absolute timelines, i.e., making the timelines directly mappable to calender dates and times, e.g. by exploiting complementary data sources such as the EventTimes Corpus (Reimers et al., 2016) and extending the current loss functions accordingly. The proposed models also provide a good starting point for research into probabilistic timeline models, that additionally model the (un)certainty of the predicted positions and durations of the entities.

[16]Code is available at: liir.cs.kuleuven.be/software.php

7

Probabilistic Absolute Timeline Extraction

This chapter has been submitted as:

Artuur Leeuwenberg and Marie-Francine Moens. 2019. Extracting Bounded Calendar Timelines from English Clinical Reports. *IEEE/ACM Transactions on Audio, Speech, and Language Processing (TASLP)*. IEEE Press.

To build towards a more complete extraction of the temporal information from text, it is important to take into account explicit information, which has been the focus of most research thus far, but also implicit information. Implicit information often involves a higher degree of uncertainty. In this chapter, we investigate how we can annotate and extract complete absolute timelines that also capture implicit and uncertain information, and can provide a probabilistic interpretation of the timeline.

We propose a timeline annotation scheme, dealing with uncertain temporal information by means of temporal bounds which we use to annotate a set of clinical intensive-care-unit records. On top of our scheme, we suggest a probabilistic interpretation allowing us to query the absolute temporal information in a probabilistic way. Finally, we construct and evaluate a first set of models for this new task, which predict absolute probabilistic timelines.

7.1 Introduction

In this chapter, we address the new task of bounded absolute timeline construction from text. Although temporal language understanding is essential for general natural language understanding, information retrieval, question answering and document summarization (Campos et al., 2015; Höffner et al., 2017; Ng et al., 2014), we focus here on the clinical domain, for which having precise temporal information is vital, and high quality temporal extraction from text could be an important enrichment of the structured electronic health record, with much potential for applications (Shahar, 1999; Jung et al., 2011). Our work in the medical domain forms a pilot for other domains.

Many temporal annotation schemes have been developed, all focusing on different aspects of temporality: relative event order (Pustejovsky et al., 2003a, 2010; Styler IV et al., 2014; Ning et al., 2018c), event durations (Pan et al., 2006a, 2011), explicit temporal cues like temporal expressions (Setzer, 2002; Ferro et al., 2005; Pustejovsky et al., 2010; Bethard and Parker, 2016).

However, for a majority of events existing schemes provide only partial event time information, leaving many event times unbounded. With bounded event time, we mean that a closed interval on the calendar timeline is given within which the annotator is sure the event must have happened (e.g., *between 2018 and 2019*). Absence of completely bounded annotations, often a result of implicitness and uncertainty of the temporal information, makes positioning of events on the absolute calendar timeline very difficult. In this chapter, we aim to deal with temporal uncertainty and integrate various types of temporal information into a single scheme to annotate fully bounded absolute timelines, with complete information about the possible calendar times and durations for each event, based on the text.

Observe an example of our proposed scheme in Figure 7.1. The bounds in our scheme model temporal uncertainty. They indicate how precisely the temporal information can be determined based on the text, which is very important to deal with implicit information, often underrepresented in current schemes, and for timeline evaluation. For example, in Figure 7.1, if we replace the word *fever* for *smoking*, the timeline should look very different, because it is more likely that *smoking* happened for a much longer time period than *fever*, and may have started, or ended further in the past (even years). Nevertheless, the existing TimeML annotations are the same for both cases, ignoring such differences in absolute position and duration. Additionally, by assuming a probability distribution on the bounds (explained further in section 7.3), our scheme allows answering probabilistic temporal questions like the probability on whether an event was taking place at, started, or ended at a particular time, but also more complex queries like the most probable time period between two events, which could be useful in practical applications and for timeline visualization.

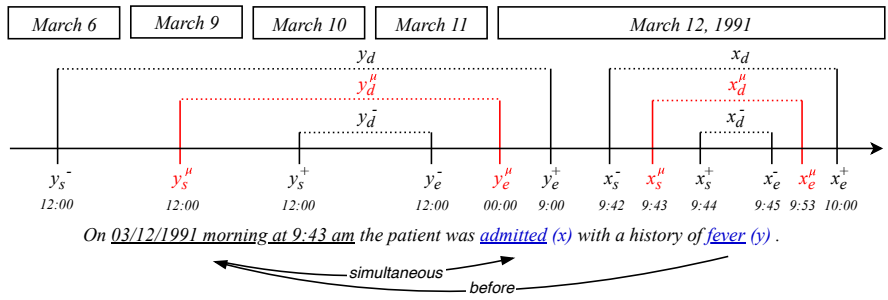

Figure 7.1: An example sentence annotated with our scheme: containing events x (admitted), and y (fever), with most likely start x_s^μ, duration x_d^μ (dotted line), and end x_e^μ (all in red), and their corresponding lower and upper bounds ($^-$, and $^+$ in black). And similarly for event y. Below the sentence the existing temporal links of TimeML are shown.

This chapter makes the following contributions[1]:

- We propose a novel annotation scheme, to annotate bounded absolute timelines, integrating various existing temporal annotation schemes efficiently.
- We annotate an English clinical corpus with our scheme, and analyze inter-annotator agreement, and its relation to TimeML.
- We propose and evaluate a multi-regression model to predict bounded absolute timelines.

Firstly, we will discuss how the work covered by this chapter relates to existing research on temporal annotation and timeline extraction. Secondly, we will discuss the annotation scheme and analyze the annotated clinical reports. Third, we will introduce our proposed model. And finally, we will describe and analyze our experiments, and discuss the conclusions we draw from them.

7.2 Related Work

Event Position

The currently most widely used annotation scheme is TimeML (Pustejovsky et al., 2003a, 2010), in which events (e.g., *a meeting*), and temporal expressions (e.g.,

[1]The dataset, annotation tool, guidelines, evaluation scripts, and model code will be made available.

yesterday, or *02/02/2001*) are temporally linked by basic Allen interval relations[2] (Allen, 1983). Adaptations of this scheme were also annotated on several clinical corpora (Sun et al., 2013a; Styler IV et al., 2014), from which we use the i2b2 temporal corpus as a starting point of our work (Sun et al., 2013a)[3]. To extract TimeML style temporal graphs, multiple shared tasks have been organized, resulting in many systems (Verhagen et al., 2007, 2010; UzZaman et al., 2013; Bethard et al., 2015, 2016, 2017). Current state-of-the-art systems are mostly neural network based models (Tourille et al., 2017a; Ning et al., 2017; Meng and Rumshisky, 2018; Liu et al., 2019). Leeuwenberg and Moens (2018a) construct relative timelines from TimeML-style predictions, where each event is modeled as a timeline interval. We adopt this method to construct absolute interval-based timelines from TimeML as a baseline.

Recently, there have been interesting developments in annotating news texts with relative temporal information (Ning et al., 2018c; Vashishtha et al., 2019), which are out of the scope of this work as we focus on extracting *absolute* timelines, which can be interpreted directly on the calendar.

TimeML links events to the absolute timeline through explicit temporal expressions, for which temporal uncertainty has been studied using fuzzy sets (Tissot et al., 2016). However, most events cannot be directly linked to such expressions, giving them no absolute anchors to the timeline. Reimers et al. (2016) address this issue and re-annotated the 36 news articles from TimeBank Dense (Cassidy et al., 2014) with a new scheme and propose a corresponding system (Reimers et al., 2018), based on a neural decision tree. Their annotations provide calendar dates for all within-day events. This way, within-day events receive absolute position bounds; the start and end of that day. For multi-day events, annotators can choose to annotate a left or right position bound, or both. This way all events are related with at least one link to the absolute timeline. However, the majority of events in their annotations remain unbounded[4]. In our scheme, we address this by providing full bounds for all events. As their scheme was annotated on news data, and is not directly derivable from available clinical annotations, we cannot compare with their work empirically.

Event Duration

TimeML covers explicit duration annotations through temporal expressions. However, it does not cover implicit durations. Because of this, for many events no annotation

[2] E.g., *before*, *simultaneous*, *during*, *overlap*, and *meets*.

[3] These documents are a subset of MIMIC III (Johnson et al., 2016), and besides temporal TimeML annotations, also carry relation annotations (Sun et al., 2013b), co-reference (Uzuner et al., 2012), and question answering information (Pampari et al., 2018) (including temporal questions), increasing the potential of this dataset for future research.

[4] In their annotations we found that 60% of events had open bounds (no left start bound or no right end bound).

of duration is present.[5] Pan et al. (2006a, 2011) added explicit and implicit duration annotations to all events of the 58-document TimeBank corpus (Pustejovsky et al., 2003b). They assigned a lower and upper duration bound to each event. As bounds on duration are most often not symmetric with regard to the most likely (mode) duration (see Sec. 7.4.1), we extend Pan et al. (2006a) by also annotating mode durations. This makes the current work the first to allow analysis of symmetry of temporal uncertainty, and the first to annotate such complete durations. Although several methods have been proposed to predict course-grained event duration (Pan et al., 2011; Gusev et al., 2011; Williams and Katz, 2012), the state-of-the-art event duration prediction model on the dataset by Pan et al. (2006a) is a Long Short-Term Memory (LSTM) network ensemble (Vempala et al., 2018), which we retrain on our data and adopt as a baseline.

Our scheme annotates on the event level. For each event mention annotators annotate two types of information: (1) the most likely event time, and (2) the temporal bounds based on the text, and the annotator's background knowledge. We start by defining the components of a timeline.

7.2.1 Mode Event Time Components

The **timeline** is interpreted as the calendar timeline, discretized on minute level. We define the event time for an event x as an interval $[x_s, x_e]$ on the timeline, ranging from its start point x_s, to its end x_e (with $x_s < x_e$).[6] The duration x_d of each event is the difference between its start and end:

$$x_d = x_e - x_s \tag{7.1}$$

So, each event time can be fully specified by any pairwise combination of event **components** x_s, x_d, or x_e. As we work on a minute scale, each point (start or end) is represented by the format: YYYY-MM-DD-hh-mm, and each duration by the format: YY-MM-DD-hh-mm.[7]

We ask annotators to annotate two out of the three event time components, in the given formats. Annotators are free to choose which two components to annotate, as the third will be derived from Equation 7.1. From now on, we refer to the most likely value of each component as its *mode*, indicated by x_s^μ, x_d^μ, and x_e^μ. We use $\boldsymbol{x}^\mu$ to refer to the mode event time, comprised of all three mode components (red in Figure 7.1).

[5] Around 83 % of all i2b2 events could not reach any TIMEX or SECTIME via simultaneous, or inclusion relations, or a combination of a before and after relation (after extensive temporal closure), indicating open absolute bounds, and absence of any duration information.

[6] Negated events are interpreted as the time during which the negation holds. Event mentions referring to multiple sub-events (e.g., *some slight headaches*) are interpreted as the smallest interval covering all sub-events (convex hull).

[7] Resulting in a maximum duration of almost 100 years, sufficient for our purposes. Calendar calculations are done with *python-datetime* (accounting for leap years).

7.2.2 Temporal Bounds

As temporal information is often underspecified in language, and exact minute-level times are most often not inferable from the text, besides annotating the mode event time, our scheme defines two temporal bounds for every event component (so six in total): a **lower bound** (indicated by $^-$), and an **upper bound** (indicated by $^+$). For each component, its two bounds provide the range of possible values, indicating the degree of uncertainty on that component.[8] The bounds have the following properties:

$$x_s^- \leq x_s^\mu \leq x_s^+ \quad \text{(start bounds)} \tag{7.2}$$

$$x_e^- \leq x_e^\mu \leq x_e^+ \quad \text{(end bounds)} \tag{7.3}$$

$$d_{min} \leq x_d^- \leq x_d^\mu \leq x_d^+ \quad \text{(duration bounds)} \tag{7.4}$$

$$x_s^\mu \leq x_e^\mu \quad \text{(start before end)} \tag{7.5}$$

A minimum duration d_{min} is introduced to prevent zero or negative durations. Notice that if we have the bounds for any two out of the three components (start, duration or end) we can infer the bounds for the third. Hence, annotators only need to annotate two components to fully specify the mode event time and all bounds.

7.2.3 Annotation Steps

To obtain mode event times, and their bounds, annotators iterate through the following steps per document: (1) Select the most certain event, (2) select its two most certain components, (3) annotate mode x_c^μ, and the lower bound x_c^- and upper bound x_c^+ for both components. Overall, annotators give 6 values per event, resulting after inference in the complete 9 values, shown in Figure 7.1.

7.2.4 Calendar Points to Numerical Values

To ease calculation with calendar values, we convert points and durations to numerical values. The numerical value for a time point t is the number of minutes it lies after a fixed reference point ρ in the past. For Figure 7.2, the reference point ρ is January 1, 1990, meaning that point x_s^μ, March 12, 1991 at 9:43 am, is 626,922 minutes later than the reference point. Using this mapping we can easily go between numerical values

[8] A small range of values between the bounds shows that the annotator believes a component can be determined quite precisely, indicating high confidence, and vice versa.

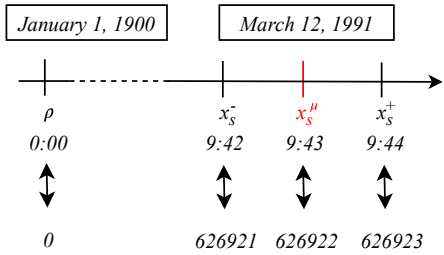

Figure 7.2: Calendar times have a one-to-one mapping to regression values, which represent the number of minutes since a reference point ρ, lying in the past.

and actual calendar dates. For all our models and analyses, the reference point was the first of January 1900, as all events in the corpus happen after this date.

7.3 Probabilistic Timelines

As our scheme captures the uncertainty of the annotated temporal information, we can construct a probabilistic interpretation of the scheme, allowing for probabilistic temporal querying.

7.3.1 Two-Piece Normal Distributions

For each timeline component x_c (start, duration, or end), consisting of lowerbound x_c^-, mode x_c^μ, and upperbounds x_c^+, we assume a two-piece normal (TPN) distribution (Wallis, 2014). As an example, two TPN distributions are shown in Figure 7.4. A TPN distribution is a combination of two half normal distributions, joint at the mode, and its probability density function (*pdf*) can be defined by a left standard deviation σ_l, a right standard deviation σ_r, and the mode μ as:

$$pdf(t) = \begin{cases} A \exp\big[-(t-\mu)^2/2\sigma_l^2\big], & t \leq \mu \\ A \exp\big[-(t-\mu)^2/2\sigma_r^2\big], & t \geq \mu \end{cases} \tag{7.6}$$

with scaling factor

$$A = \left(\sqrt{2\pi}\,(\sigma_l + \sigma_r)\,/2\right)^{-1} \tag{7.7}$$

7.3.2 Annotations as Distributions

Because the TPN distribution is an asymmetric distribution that can be parameterized by exactly three values: σ_l, σ_r, and μ, they align well with our asymmetric bound annotations[9], consisting of mode x^{μ}, lower bound x^{-} and upper bound x^{+}. For each event component c we convert its annotations to a TPN distribution by setting:

$$\mu := x_c^{\mu}$$

$$\sigma_l := x_c^{\mu} - x_c^{-}$$

$$\sigma_r := x_c^{+} - x_c^{\mu}$$

This means that for each event, three TPN distributions are obtained: for the start, duration, and end components. These distributions form the probabilistic interpretation of our bounded annotations. Our proposed models, introduced later in section 7.5, predict mode component values, and their deviations. Hence, we can construct the corresponding TPN distributions for predicted event components in the same way.

7.3.3 Probabilistic Querying

The *pdf* distribution models the probability density for an event component c across time t (e.g., $pdf_s(t)$ gives the probability density for the start of the event). We can use the cumulative functions of the start and end components to determine whether an event has started or ended before a certain point t. The cumulative function of a TPN distribution is given by Equation 7.8, with erf$(\cdot)$ as the Gaussian error function.

$$cdf(t) = \begin{cases} \frac{\left(1+\text{erf}\left[\frac{t-\mu}{\sqrt{2}\sigma_l}\right]\right)\sigma_l}{\sigma_l+\sigma_r} & t \leq \mu \\ \frac{\sigma_l+\text{erf}\left[\frac{t-\mu}{\sqrt{2}\sigma_r}\right]\sigma_r}{\sigma_l+\sigma_r} & t \geq \mu \end{cases} \tag{7.8}$$

From the cumulative functions, we can calculate the probability that an event is actively taking place at time t as the probability that the event has started, minus the probability that it has ended, i.e., $cdf_s(t) - cdf_e(t)$, as shown in Figure 7.3.

7.4 The Annotated Clinical Dataset

Three annotators with > 3 years of study in Biomedicine annotated in total 169 English clinical reports from the i2b2 temporal challenge (Sun et al., 2013a). Dataset statistics

[9] Other asymmetric distributions may also be viable alternatives (e.g., two-piece Laplace distributions).

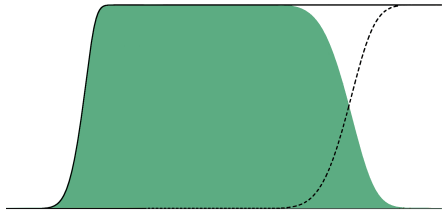

Figure 7.3: Probability that some event: (1) has started before time t ($cdf_s(t)$: solid black line), and (2) that it has ended before time t ($cdf_e(t)$: dashed line), and (3) is happening at time t ($cdf_s(t) - cdf_e(t)$, in green).

	$A_1 \cup A_2 \cup A_3$	A^2
Documents	169	37
Events	12,882	2,451

Table 7.1: Statistics on full dataset[11] $A_1 \cup A_2 \cup A_3$, and the subset annotated by at least two annotators A^2.

are given in Table 7.1. The documents are already annotated with TimeML from which we adopt event span annotations, on which we annotate our scheme.

We have built a new annotation tool, which, besides inference, provides insight to the annotator about their own annotations by visualization of the mode timeline, and which includes short keys to reuse start, duration, or end annotations of already annotated events. Using this tool, the average annotation time per document is around 60 minutes, which is comparable to 55 minutes per documents of the TimeML annotations (Sun et al., 2013a). The annotators regularly discussed difficulties in person with the adjunctator, and used a shared document to establish agreement on difficult cases.

7.4.1 Dataset Analysis

We analyzed the annotated values in terms of order of magnitude. This is shown in Table 7.2. Firstly, 100% of events have very high mode start and end values; this is because they lie multiple decades from the used reference point $\rho = 1900$. More interestingly, we can see that most bounds have a width of hours or days for all components. Also, the vast majority of events have a duration in the range of hours or days. Another very interesting observation we can make is that the deviations seem very asymmetric. For all components right deviations are generally larger than left deviations. We speculate for start and end points that this is because readers go through the text linearly, and because the narrative clinical records generally are partially chronologically. This can result in the fact that while reading, they have more knowledge about past events,

	Start			End			Duration		
	μ	σ_l	σ_r	μ	σ_l	σ_r	μ	σ_l	σ_r
ho	0	67	0	0	66	0	63	66	43
da	0	16	70	0	16	68	17	19	36
we	0	6	13	0	6	14	7	3	7
mo	0	6	6	0	6	6	7	6	5
ye	0	3	6	0	3	5	4	3	4
de	100	3	5	100	4	6	3	3	5

Table 7.2: For all nine annotated components, we show the distribution (in %) of the number of events that was annotated with a value of a certain order of magnitude (hours, days, weeks, months, years, decades).

	$P_s^{\cap}$	$P_s^{\in}$	$P_s^{<}$	$P_e^{\cap}$	$P_e^{\in}$	$P_e^{<}$	$P_d^{\cap}$	$P_d^{\in}$	P^{tl}
I	42 (46)	65 (70)	82 (87)	39 (42)	63 (65)	74 (80)	32 (36)	60 (71)	77 (81)
II	60 (61)	88 (84)	82 (87)	54 (56)	86 (83)	74 (80)	59 (65)	87 (88)	77 (81)

Table 7.3: Agreement percentages for the different proposed metrics on the raw annotations (I), and agreement after extending the bounds to day, week, month, year, and decade level, which are the final annotations used in the experiments (II). In-between brackets, the score for the 17% subset of events is given that were already course-bounded by the existing TimeML-annotations.

which can provide more certainty on the left bound, whereas about future events less information is given at that point, resulting in larger bounds. The fact that the past, even in the real world, is generally more certain than the future can also influence the writer of the document, and his/her way of incorporating temporal cues. For durations, we believe the asymmetric uncertainty is because events have a minimum duration: they cannot be shorter than 0 minutes. So in cases of high uncertainty the left deviation approaches 0, while the right deviation can grow, in principle, for ever. These results show that temporal uncertainty is best modeled by an asymmetric distribution.

To calculate inter-annotator agreement (IAA) we use all 37 documents that are annotated by at least two annotators. We calculate agreement as a weighted mean of pairwise agreements of all three pairwise combinations of the three annotators, where the weight is in proportion with the number of annotated events shared by each pair of annotators. To analyze the annotations in detail, we calculate several metrics of agreement. Their results are given in Table 7.3. We will now discuss the used metrics one-by-one.

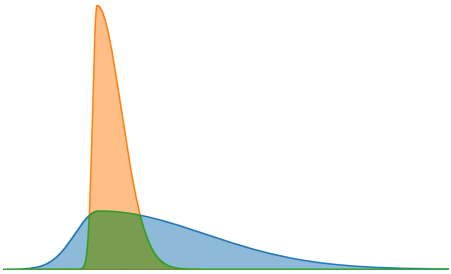

Figure 7.4: Two two-piece normal distributions for the same event's start time: overlap $P_s^{\cap}$=0.38 (in green).

7.4.2 Overlap Agreement ($P^{\cap}$)

Our first metric to calculate IAA between two annotators is calculated as the proportion of overlap between the TPN-distributions for each component (intersection over union).[12] A visualization of this metric is shown in Figure 7.4. This metric takes into account all components of the annotations in a single score (left bound, mode, and right bound), and is therefor quite strict, but complete. On the raw annotations, we obtain a $P^{\cap}$ =32% for duration, $P^{\cap}$ =42% for start points, and $P^{\cap}$ =39% for event endings. At first these scores seem quite low. However, it should be taking into account that these numbers cannot be interpreted in the same way as for a classification task where annotators choose between a fixed set of classes. As for this task annotators are free to annotate any value on the timeline (for a time period of 200 years; approximately 10^8 minutes). We discuss this further in section 7.4.6.

7.4.3 Inclusion Agreement ($P^{\in}$)

As mentioned earlier, $P^{\cap}$ is very strict: even if two annotators agree on almost the exact mode value, the $P^{\cap}$ score can be low, as they might disagree on the width of the bounds (as for Figure 7.4), and vice versa. To account for this we also calculate the percentage of times the mode of one of the annotators is included within the bounds of the other. In other words, how often does one annotator believe the other's most likely timing is possible. This is visualized in Figure 7.5.

[12] Tissot et al. (2016) used a similar metric, based on fuzzy sets instead of TPN distributions, to study imprecise timexes.

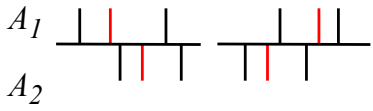

Figure 7.5: An example of inclusion $P^{\in}$: on the left, the mode value (red) of A_2 is included in the bounds of A_1 (agreement). And on the right, neither mode value is included in the bounds of the other (disagreement).

7.4.4 Agreement on Temporal Order ($P^{<}$)

To analyze IAA with regard to relative event order we inspect all pairs of events in each document per annotator, and inspect the order relation between the start points of the event pairs ($>, =$, or $<$). Agreement corresponds to the percentage of event pairs that are assigned the same order relation. In 82% of cases annotators agree on the order of start points, and in 74% they agree on the order of endings.

Like Ning et al. (2018c) observed for news articles, we observe that IAA on the order of start points is higher than that of end points, which could be caused by uncertainty on event duration.

7.4.5 Agreement with TimeML (P^{tl})

To be able to better compare our timeline annotations with the existing TimeML annotations, we follow the strategy of Leeuwenberg and Moens (2018a) to evaluate relative timelines using TimeML, and assign each TimeML-annotated event pair a temporal link (TLink), based on the timeline, and calculate accuracy with the originally annotated TLinks. For this we use the merged TLinks present in the data (*before*, *after*, and *overlap*) by Sun et al. (2013a). Following their annotation guidelines as close as possible, we use the following classification function to assign TLink types to event-event and event-timex pairs:

$$R(x,y) = \begin{cases} \textit{before} & \text{iff } x_s^\mu < y_s^\mu \\ \textit{after} & \text{iff } x_e^\mu > y_e^\mu \\ \textit{overlap} & \text{iff } x_s^\mu \geq y_s^\mu \wedge x_e^\mu \leq y_e^\mu \end{cases} \tag{7.9}$$

When classifying the TimeML TLink types based on our timeline annotations we obtain an accuracy of 0.77. This score is a lower bound on the agreement between the two schemes, as there is no exact mapping between the merged TLinks and the timeline in the guidelines.

7.4.6 Changing Bound Granularity

As can be seen from Table 7.3, the agreement on metrics that are influenced by the width of the bounds are fairly low ($P^{\cap}$ and $P^{\in}$). One important reason for this is the fine minute-level granularity of the annotations. When inspecting the cases of disagreement, we found that annotators have different judgments of amount of uncertainty, even though they often agree quite precisely on the event's timing. To increase agreement, we decrease the granularity of the bounds. We extend bounds that lie within one day, to the start and end of that day. We also do this for bounds within one week, and similarly for months, years, and decades. We do not change the minute-level mode annotations, ensuring that the order of the events does not change, even within days. If we analyze agreement again, shown in the second row of Table 7.3, we observe much higher agreement, especially for inclusion agreement $P^{\in}$, indicating that on a more course grained level the annotators agree well on event position, and duration. We use these course bounded timelines, with high agreement, as our final data for model construction.

7.5 Absolute Timeline Model (ATLM)

For each event, our model predicts the mode start time, and mode duration and their corresponding bounds (from which the end time and bounds follow automatically). Its input is the text with ground truth event spans, and normalized temporal expressions, as this is not the focus of this paper. Our model is shown in Figure 7.6. It is constructed of four modules: (1) word representation (2) anchoring, (3) shifting, and (4) a duration module. We will discuss each module below.

7.5.1 Word Representations

We experiment with two types of word representations: (1) 300-dimensional GloVe embeddings (Pennington et al., 2014) trained on 300M words from the clinical MIMIC III dataset (Johnson et al., 2016), and (2) ELMo embeddings (Peters et al., 2018), in particular the embeddings by Zhu et al. (2018), which are trained on the same clinical dataset. We use ELMo for its ability to capture character-level information, important for encoding temporal expressions (Xu et al., 2019).

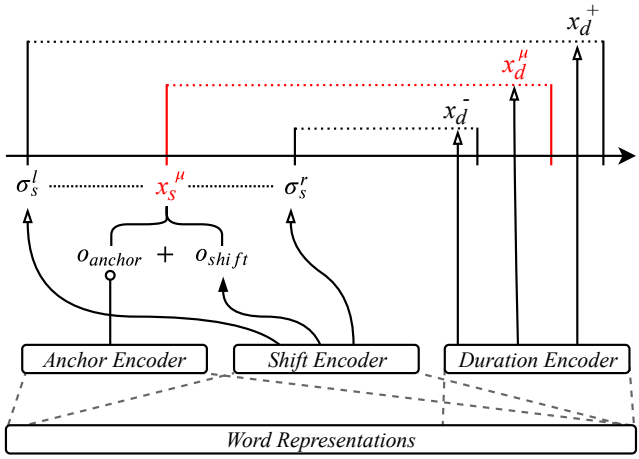

Figure 7.6: A schematic overview of our model, which predicts the start and duration modes of each event, and the corresponding bounds from the input sentence.

7.5.2 Event Durations

To predict event durations, we use a simple model, taking as input the event, and its local left and right context (size: 1), as this has shown to be effective features for estimating event duration (Vempala et al., 2018). We encode the event and its context using either an LSTM (Hochreiter and Schmidhuber, 1997) or CNN[13] (Fukushima, 1980), and from the encoding we directly predict the mode event duration x_d^μ, and its bounds x_d^-, and x_d^+, through a regression layer (detailed in section 7.5.4).

7.5.3 Start Times: Anchoring and Shifting

For each event, we predict start times in two steps: First, we find the temporally closest relevant date/time expression, and use its normalized value as an anchor (o_{anchor}): We use the first left and right date/time expression from the event as candidate anchors, and classify which one is temporally closest based on the context between the event and each candidate anchor, encoded using the anchor encoder.

[13]For LSTM we used 75 dimensions and for CNN we used 75 filters, with window sizes 2, 4, and 6.

Second, based on the encoded context between the event and the found anchor, we predict a shift value o_{shift}, indicating how much the event's start time is shifted with regard to the anchor, such that $o_{anchor} + o_{shift} = x_s^\mu$. Additionally, from the same encoded context, we estimate the left and right start time deviations σ_s^l, and σ_s^r, to obtain the lower and upper start bounds via $x_s^- = x_s^\mu - \sigma_s^l$, and $x_s^+ = x_s^\mu + \sigma_s^r$. Now that we have the start and duration component predictions, we can infer those of the end component, and obtain the predicted TPN distributions following section 7.3.2.

7.5.4 Regression Layers

In this section, we explain the meaning of arrows in Figure 7.6. To predict an output value o from some input encoding i, we use a feed-forward layer with one hidden layer (of half the input dimension, and Leaky ReLU activation), and a single output node with: (1) linear activation (closed-headed arrow), or (2) a softplus activation, $ln(1+e^x)$, to enforce output values to be positive (open-headed arrows). The ball-headed arrow indicates the binary logistic anchor classifier, followed by the action of setting o_{anchor} as the normalized datetime value of the predicted anchor.

7.5.5 Model Training

To train the anchor encoder to choose the left or right closest temporal expression, we use a binary cross entropy loss.

For training the prediction of modes and deviations we used the L_1 loss as given by Equation 7.10. The total loss is the averaged loss across all N events. The event-level loss $l(\cdot)$ for each event x in turn is the sum of the component-wise losses for event time components C: start, duration and end.[14]

$$L_1 = \frac{1}{N} \sum_{i=1}^{N} l(x_i) \tag{7.10}$$

$$l(x) = \sum_{c}^{C} |\hat{x}_c^\mu - x_c^\mu| + |\hat{x}_c^{\sigma_l} - x_c^{\sigma_l}| + |\hat{x}_c^{\sigma_r} - x_c^{\sigma_r}| \tag{7.11}$$

For minimization we used Adam (Kingma and Ba, 2014), with default parameters. As high regression values make training unstable, we rescale the timeline such that years 1900-2100 lie in the interval [0,1] by dividing all values by scaling factor 10^8. All models are trained for a maximum of 200 epochs using a held-out 15-document

[14] We have experimented with some alternative loss functions, but these did not result in improvements.

validation set for early stopping (Morgan and Bourlard, 1990), with a patience value of 20.

7.6 Experiments

In this section we describe the evaluation of our anchor and shift-based absolute bounded timeline extraction model (ATLM), using either LSTM or CNN as encoder components.

7.6.1 Evaluation

Our annotated corpus is split into a 132-document training set (10,431 events) and 37-document test set (2,451 events). We chose the test to consist of all documents that have been annotated by more than one annotator. This way the agreement measures give a realistic indication of the upper bound for system performance. We create ground-truth annotations by taking the mean values of all the annotators. From the mean values, we create the corresponding TPN distributions as explained in section 7.3.2. For evaluation we calculate measures proposed in section 7.4.1. Hyper-parameters are tuned on a small 15-document development set (from training).

7.6.2 Baselines

As there is not yet a model which predicts absolute bounded timelines, we construct baselines from existing state-of-the-art models.

Event Duration Baseline (D-LSTM)

As this is the first clinical corpus to annotate full event durations, as a baseline for predicting event duration we implemented the current state-of-the-art model for news texts by Vempala et al. (2018). Their model is an LSTM-ensemble build on top of GloVe embeddings. To adapt their model to the clinical domain, we retrain the GloVe embeddings on 100M words of clinical reports from MIMIC III (Johnson et al., 2016). As Vempala et al. (2018) only classify events into two duration categories: $\leq$ a day, and $>$ a day, instead of a binary softmax output on top of its event encoder, we use three regression layers, as in section 7.5.4, to predict the duration mode, and its left and right deviations.

TLinks to Timeline (TL2ATL)

As the TLinks in TimeML anchor some of the events to the absolute timeline, we can also construct a TLink-extraction-based baseline. First, to extract TLinks, we retrained a publicly online available neural state-of-the-art clinical TLink extraction model (Leeuwenberg and Moens, 2018b) on our data split, using the existing TimeML annotations for training. From the extracted TLinks of types *before*, *after*, and *overlap*, we position the events on the timeline following the TLinks-to-Timeline method by Leeuwenberg and Moens (2018a): (1) Each event is assigned an interval with start variable x_s^μ and end variable x_e^μ. (2) The variables are set such that the predicted TLinks between the events are satisfied on the timeline[15]. Determining the variable values is done by minimizing a loss function that reflects the degree to which the TLinks are satisfied. We interpret the TLinks as given in Equation 7.9, modeling pointwise order ($a < b$) as a margin-based hinge loss, $max(a+m-b, 0)$, with a margin m of 1 minute, and equality ($=$) with an L1 loss, $|a - b|$. As events are also linked to TIMEXES, we assign two fixed constants x_s^μ, and x_e^μ to each TIMEX following their annotated ground-truth normalized values. This way the TIMEXES function as anchors on the timeline. For optimization we use Adam (Kingma and Ba, 2014).

7.7 Results and Analysis

	$P_s^\cap$	$P_s^\in$	$P_s^<$	$P_e^\cap$	$P_e^\in$	$P_e^<$	$P_d^\cap$	$P_d^\in$	P^{tl}
IAA	60 (61)	88 (84)	82 (87)	54 (56)	86 (83)	74 (80)	59 (65)	87 (88)	77 (81)
Baselines:									
TL2ATL	-	13 (9)	52 (53)	-	15 (11)	49 (55)	-	30 (40)	**68** (**67**)
D-LSTM	-	-	-	-	-	-	11 (13)	**97** (**97**)	-
Proposed:									
ATLM-CNN-GLOVE	36 (42)	67 (62)	62 (61)	35 (38)	65 (62)	59 (61)	13 (14)	**97** (**97**)	55 (59)
ATLM-CNN-ELMO	**48** (**56**)	**83** (**81**)	56 (66)	39 (45)	72 (73)	62 (**67**)	30 (36)	91 (93)	59 (63)
ATLM-LSTM-GLOVE	44 (53)	79 (79)	57 (67)	**42** (**47**)	**75** (74)	62 (64)	9 (11)	96 (94)	56 (59)
ATLM-LSTM-ELMO	37 (37)	44 (32)	**67** (**71**)	29 (30)	**75** (**75**)	**65** (65)	**47** (**51**)	78 (76)	60 (62)

Table 7.4: Results of the different absolute timeline models on the test set. With in-between brackets, the scores for the 17% subset of events that were already bounded by the existing TimeML-annotations. In short: $P^\cap$ evaluates the entire predicted distributions (including mode and bound prediction), $P^\in$ evaluates the predicted mode values (while taking into account uncertainty), $P^<$ evaluates temporal order of the mode values, and P^{tl} evaluates TLink accuracy.

[15] In our experiments the predicted TLinks were satisfied for 95%. This is due to inconsistency of the predicted TLinks.

From Table 7.4, we observe that the timelines predicted by TL2ATL better satisfy the existing TLinks in the test set compared to the ATLM models (8-13% higher in P^{tl}, with $p < 0.0001$[16]). This can be expected, as TL2ATL used the TimeML Tlinks as training data, causing the model to focus more on these relations.

For all other metrics, we can see that the ATLM models perform significantly better (between 10-50% depending on the metric, with at least $p < 0.01$[16]). We believe the primary reason for this is that our scheme, on which the ATLM models were trained, provides more complete temporal information for more events, compared to the TLinks of TimeML, which provide complete information for some events, but hardly any information on others.

For the ATLM models, with regard to event starts, the best model combines the CNN with ELMo embeddings. However, we do not observe clear general trends when comparing CNN with LSTM, or when comparing GloVe embeddings with ELMo embeddings across encoders.

When looking at predicted durations, the best model in terms of overlap ($P^{\cap}$) combines the LSTM with ELMo embeddings. There is a clear improvement for both the CNN and LSTM model when using ELMo embeddings instead of GloVe embeddings ($p < 0.0001$[16]), which suggests that ELMo embeddings seem better at capturing duration information. Since for both start and duration, the best model used ELMo embeddings, we argue that this representation is generally more informative. A reason for this can be the availability of a wide context for the ELMo language model, compared to GloVe representations. We also believe this is the primary reason that our models perform better than the state-of-the-art D-LSTM baseline for durations, as this model uses GloVe embeddings.

Another observation is that for most metrics, in general most models perform slightly better on the TimeML-bounded subset. We believe this is due to the slightly higher IAA on these events, which can in turn be the result of the fact that TimeML focuses on explicit temporal information, whereas we focus on both explicit and implicit information. Overall, mostly for event position (start and end), we find a significant gap between system performance and IAA, indicating much room for improvement. From manual inspection of the timelines predicted by ATLM-LSTM, we found that the model best predicts events with smaller durations, and events lying *temporally* close to a temporal expression (the majority of events, as shown in Table 7.2). It indicates that the models have more difficulty with events with longer durations, and events for which the shifts are higher. We believe this can be explained by the fact that these events are a minority of the cases.

Finally, if we compare the best models against the inter-annotator agreement scores, we observe that the inclusion metrics ($P^{\in}$) already lie quite close to the IAA, particularly

[16]Significance is based on a document-level paired t-test.

for durations. This shows that the predicted mode values are already within the bounds of the ground truth annotations. It should be mentioned that the vast majority of events happen within a single day, making this the easiest sub-task. For all overlap metrics ($P^{\cap}$), which evaluate the complete predicted distributions, we observe that the best systems perform reasonably well, given the fact that this is a new and very challenging task. However, there is still a significant gap between the best performing systems and the IAA, indicating room for future work into further model development.

7.8 Conclusions

In this chapter, we address the task of complete absolute timeline construction from text, accommodating for temporal uncertainty and implicit temporal information. Extraction of high quality timelines not only gives important insights in general language understanding, but also carries important potential for applications in for example the clinical domain.

We propose a novel annotation scheme, to extract completely bounded absolute event timelines from text, based on both explicit and implicit temporal information. We annotate an English clinical corpus, and analyze inter-annotator agreement, and our scheme's relation to TimeML. Finally, we propose and evaluate a multi-regression model to extract the absolute timelines. Results show the asymmetry of temporal uncertainty, indicate the difficulty of this new task, and highlight the value of our approach compared to existing approaches, providing a solid benchmark for further development in this area of research.

8 Conclusions

> People assume time is a strict progression of cause to effect, but actually, . . . , it's more like a big ball of wibbly wobbly timey wimey stuff.
>
> *Steven Moffat*

8.1 Thesis Summary

In this dissertation, we investigated how we can build and further improve computer models that extract event timelines from textual documents, focusing on the clinical domain. Construction of high quality timelines from clinical documents plays an increasingly important role with the increasing digitization of personal health records.

We first introduced the research questions and contributions discussed in the thesis (Chapter 1). Two different approaches were proposed to improve current state-of-the-art temporal relation extraction models, by integrating temporal rules (Chapter 3), and by efficiently combining labeled and unlabeled data (Chapter 4). To advance towards the prediction of event timelines, new methods were developed to learn models that can predict timelines with or without intermediate relation extraction stage, based on currently existing data and annotation schemes (Chapter 6). Extending on this work, a new scheme and dataset were created to construct models that predict more complete and informative timelines, that take into account implicit and uncertain temporal information (Chapter 7). Intermediately, a thorough literature study was

conducted on the successful integration of temporal reasoning in temporal information extraction models (Chapter 5).

Chapter 3 investigated how we can exploit the dependencies between temporal relations to improve temporal relation extraction models. A structured perceptron model was proposed to learn to predict temporal relations between events and the document-creation time, and between temporal entities in the text. The model incorporates global features and temporal constraints, allowing it to benefit from the dependencies between the different temporal relations. Our best system outperformed the state of the art on the clinical THYME dataset at the time.

Manual annotation of textual data with temporal information is costly. To maximally utilize the data available, we investigated how can we use raw text, in addition to our annotated texts, to improve word representations of temporal relation extraction models. In Chapter 4, we proposed a neural multi-task model for the extraction of temporal containment relations from clinical texts. The model trains word representations jointly on the supervised relation classification task (using annotated texts) and an unsupervised auxiliary skip-gram objective (using raw text) through weight sharing. This resulted in significantly better generalizing classification models compared to using pre-trained word embeddings, and further improved the state of the art for temporal relation extraction on the THYME dataset, even without using extra dedicated clinical resources, in contrast to existing state-of-the-art models.

After studying temporal relation extraction models, extensively covered in the literature, we prepared for the next less explored step: timeline construction by making a thorough inquiry into how temporal reasoning can be used efficiently for manipulation of temporal information. We investigated how temporal reasoning has been used in the general research field of temporal information extraction, and conducted a literature survey (Chapter 5). The survey provides an overview of how temporal reasoning has been exploited in all steps of model construction: annotation, data preprocessing, training, prediction, and evaluation. It highlighted several underexposed research areas, like the handling of temporal uncertainty for durations and event positions, and stressed the importance of point-based reasoning.

Based on these insights, we addressed the task of timeline construction (in Chapter 6). Our investigation led to the development of a new method, TL2RTL, to predict relative time-lines from a set of predicted temporal relations, and two new models, (S-TLM, and C-TLM) that – to our knowledge for the first time – predict (relative) time-lines in linear complexity from text, by evading the computationally expensive (often $O(n^2)$) intermediate relation extraction phase in earlier work.

As relative timelines provide information on the order of events, but provide no absolute calendar interpretation of the event times, and because the TimeML annotations used in most experiments focus on explicit temporal cues, providing only a partial temporal

picture of the text, we arrive at our final investigation into the extraction of uncertain, often implicit, temporal information. In chapter 7, we addressed the task of complete absolute timeline extraction, including both explicit and implicit information. For this, we proposed a novel annotation scheme and annotated an English clinical corpus. Finally, we developed and evaluated a multi-regression model to extract the absolute timelines. Results show the value of including implicit information compared to existing approaches, but also indicate the difficulty of this new task, and the need for further model development in this new area of research.

8.2 Opportunities, Perspectives and Future Work

Based on the research covered in this dissertation, a number of avenues for future work have opened up.

We start by extending on the previous chapter. The development of bounded absolute timeline models is still very new, which means that there is still much potential in improvement due to adaptation of the basic model architecture. Attention-based encoders could be very interesting extensions to explore (Vaswani et al., 2017), but also the exploration of alternatives to the anchor-and-shift strategy to improve the estimation of uncertainty bounds could be interesting follow-up directions.

We continue by mentioning a large challenge in temporal information extraction: the cost of annotation. Although recent annotation schemes have focused on event-level annotations instead of pairwise relation annotations to make the annotation more efficient (e.g., Reimers et al. (2016) and our work in Chapter 7), manual annotation of texts with timelines remains a difficult, and time-consuming annotation task. It is well known that incorporation of background knowledge on the language side can reduce the need for annotated data (as in Chapter 4). Recently, the research in this field has advanced rapidly, and introduced new language models (Peters et al., 2018; Devlin et al., 2018), that have shown to improve generalization for a wide range of tasks, and have recently become available for the clinical domain (Alsentzer et al., 2019).

Alongside insertion of domain knowledge about the structure of language, extraction of temporal knowledge about events from already available structured data is a promising direction for future work. With the introduction of our absolute timeline annotation scheme, it becomes a possibility to directly link manually annotated data with possibly available structured event time data. For example, in most hospitals most procedures are time-stamped. This means that we can directly relate procedures mentioned in the annotated texts, with the structured data, and possibly learn an alignment function between events in the text and events in the structured data inheriting their time stamps. With the availability of the structured time stamp information from the MIMIC III

dataset (Johnson et al., 2016), from which the reports in our corpus are taken, this research line could be an interesting next step to explore.

Beside the creation or extraction of new data, combining existing data sources through use of temporal reasoning could help bridge the gap in performance. Based on our literature survey in Chapter 5, we observed that temporal reasoning is already used in all stages of model construction, although most of these models focus on the extraction of only certain types of cues. We have seen (in Chapters 3, 5, 6, 7) that it is possible to use temporal reasoning frameworks to combine different (temporal) extraction tasks, and that they provide tools to translate information across different temporal representations (intervals, points and durations), facilitating even more efficient use of the currently available temporal data.

Combining temporal data with data from related NLP tasks could also possibly be done more efficiently. One example of a very related task is event coreference, the task of clustering event mentions that refer to the same event. Event coreference is related to temporal information extraction as each event (with possibly multiple mentions) can happen only at one time, so all temporal cues that apply to event mentions of the same event should be temporally coherent. This was already pointed out and exploited by (Do et al., 2012), using TimeML-style data and interval-based reasoning. We believe that the absolute timeline prediction models (and annotations) proposed in Chapter 7 may facilitate easier integration with event coreference compared to using TimeML-based temporal graphs: Checking if different (probabilistic) absolute intervals (one for each event mention) are coherent with each other and could possibly refer to the same event is fairly easy (e.g., one way to do this could be by looking at the probability that they overlap in time). However, checking if two nodes in a temporal graph could be unified is not, especially if the graphs are temporally inconsistent, incomplete, or when multiple documents are involved, and the temporal graph may be very large.

8.3 Epilogue

This dissertation stresses the importance of good integration of symbolic temporal reasoning and statistical (neural) models for temporal information extraction from text. Symbolic reasoning plays a key role in capturing the rigid structure of time, whereas statistical (neural) models are pivotal when dealing with the ambiguous nature and vagueness of language.

The contributions in this thesis provide examples of this integration, and can be a starting point for future research into the combination of these fields.

Bibliography

Omri Abend, Shay B Cohen, and Mark Steedman. 2015. Lexical event ordering with an edge-factored model. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 1161–1171. ACL.

James F Allen. 1983. Maintaining knowledge about temporal intervals. In *Communications of the ACM*, pages 832–843. ACM.

Emily Alsentzer, John Murphy, William Boag, Wei-Hung Weng, Di Jindi, Tristan Naumann, and Matthew McDermott. 2019. Publicly available clinical bert embeddings. In *Proceedings of the Clinical Natural Language Processing Workshop*, pages 72–78.

Tim Althoff, Xin Luna Dong, Kevin Murphy, Safa Alai, Van Dang, and Wei Zhang. 2015. TimeMachine: Timeline generation for knowledge-base entities. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 19–28. ACM.

Juan Carlos Augusto. 2005. Temporal reasoning for decision support in medicine. In *Artificial Intelligence in Medicine*, volume 33, pages 1–24. Elsevier.

Regina Barzilay and Kathleen R McKeown. 2005. Sentence fusion for multidocument news summarization. In *Computational Linguistics*, volume 31, pages 297–328. MIT Press.

Jonathan Baxter et al. 2000. A model of inductive bias learning. *Journal of Artificial Intelligence Research*, 12(149-198):3.

Michel Berkelaar, Kjell Eikland, Peter Notebaert, et al. 2004. lpsolve: Open source (mixed-integer) linear programming system. In *Eindhoven University of Technology*.

Steven Bethard. 2013. ClearTK-TimeML: A minimalist approach to TempEval 2013. In *Proceedings of the International Workshop on Semantic Evaluation (SemEval)*, volume 2, pages 10–14. ACL.

Steven Bethard, William J Corvey, Sara Klingenstein, and James H Martin. 2008. Building a corpus of temporal-causal structure. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, pages 908–915. ELRA.

Steven Bethard, Leon Derczynski, James Pustejovsky, and Marc Verhagen. 2014. Clinical TempEval. *arXiv preprint arXiv:1403.4928*.

Steven Bethard, Leon Derczynski, Guergana Savova, James Pustejovsky, and Marc Verhagen. 2015. Semeval-2015 Task 6: Clinical TempEval. In *Proceedings of the International Workshop on Semantic Evaluation (SemEval)*, pages 806–814. ACL.

Steven Bethard, James H. Martin, and Sara Klingenstein. 2007. Timelines from text: Identification of syntactic temporal relations. In *Proceedings of the International Conference on Semantic Computing*, pages 11–18.

Steven Bethard and Jonathan Parker. 2016. A semantically compositional annotation scheme for time normalization. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, pages 3779–3786, Paris, France. ELRA.

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 International Workshop on Semantic Evaluation (SemEval)*, pages 1052–1062. ACL.

Steven Bethard, Guergana Savova, Martha Palmer, and James Pustejovsky. 2017. SemEval-2017 Task 12: Clinical TempEval. In *Proceedings of the International Workshop on Semantic Evaluation (SemEval)*, pages 565–572, Vancouver, Canada. ACL.

Branimir Boguraev and Rie Kubota Ando. 2007. Effective use of TimeBank for TimeML analysis. In *Annotating, Extracting and Reasoning about Time and Events*, pages 41–58. Springer.

Philip Bramsen, Pawan Deshpande, Yoong Keok Lee, and Regina Barzilay. 2006a. Finding temporal order in discharge summaries. In *Proceedings of the AMIA Annual Symposium*, volume 2006, page 81:5. American Medical Informatics Association.

Philip Bramsen, Pawan Deshpande, Yoong Keok Lee, and Regina Barzilay. 2006b. Inducing temporal graphs. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 189–198. ACL.

Ricardo Campos, Gaël Dias, Alípio M Jorge, and Adam Jatowt. 2015. Survey of temporal information retrieval and related applications. *ACM Computing Surveys*, 47(2):15.

Rich Caruana. 1998. Multitask learning. In *Learning to Learn*, pages 95–133. Springer.

Taylor Cassidy, Bill McDowell, Nathanael Chambers, and Steven Bethard. 2014. An annotation framework for dense event ordering. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, volume 2, pages 501–506. ACL.

Nathanael Chambers, Taylor Cassidy, Bill McDowell, and Steven Bethard. 2014. Dense event ordering with a multi-pass architecture. In *Transactions of the Association for Computational Linguistics*, volume 2, pages 273–284.

Nathanael Chambers and Dan Jurafsky. 2008a. Jointly combining implicit constraints improves temporal ordering. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 698–706. ACL.

Nathanael Chambers and Dan Jurafsky. 2008b. Unsupervised learning of narrative event chains. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 789–797.

Nathanael Chambers, Shan Wang, and Dan Jurafsky. 2007. Classifying temporal relations between events. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 173–176. ACL.

Angel X Chang and Christopher D Manning. 2012. SUTime: A library for recognizing and normalizing time expressions. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, volume 2012, pages 3735–3740. ELRA.

Fei Cheng and Yusuke Miyao. 2017. Classifying temporal relations by bidirectional LSTM over dependency paths. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, volume 2, pages 1–6. ACL.

Hao Cheng, Hao Fang, and Mari Ostendorf. 2015. Open-domain name error detection using a multi-task RNN. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 737–746. ACL.

Yao Cheng, Peter Anick, Pengyu Hong, and Nianwen Xue. 2013. Temporal relation discovery between events and temporal expressions identified in clinical narrative. In *Journal of Biomedical Informatics*, volume 46, pages S48–S53. Elsevier.

Timothy Chklovski and Patrick Pantel. 2004. VerbOcean: Mining the Web for fine-grained semantic verb relations. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 33–40. ACL.

Edward Choi, Mohammad Taha Bahadori, Jimeng Sun, Joshua Kulas, Andy Schuetz, and Walter Stewart. 2016a. Retain: An interpretable predictive model for healthcare using reverse time attention mechanism. In *Advances in Neural Information Processing Systems*, pages 3504–3512.

Edward Choi, Andy Schuetz, Walter F Stewart, and Jimeng Sun. 2016b. Using recurrent neural network models for early detection of heart failure onset. In *Journal of the American Medical Informatics Association*, volume 24, pages 361–370. Oxford University Press.

Edward Choi, Andy Schuetz, Walter F Stewart, and Jimeng Sun. 2016c. Using recurrent neural network models for early detection of heart failure onset. *Journal of the American Medical Informatics Association*, 24(2):361–370.

Prafulla Kumar Choubey and Ruihong Huang. 2017. A sequential model for classifying temporal relations between intra-sentence events. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1796–1802. ACL.

Michael Collins. 2002. Ranking algorithms for named-entity extraction: Boosting and the voted perceptron. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 489–496. ACL.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the International Conference on Machine Learning*, pages 160–167. ACM.

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.

Savelie Cornegruta and Andreas Vlachos. 2016. Timeline extraction using distant supervision and joint inference. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1936–1942. ACL.

Francisco Costa and António Branco. 2013. Temporal relation classification based on temporal reasoning. In *Proceedings of IWCS*, pages 59–70.

Agata Cybulska and Piek Vossen. 2014. Using a sledgehammer to crack a nut? Lexical diversity and event coreference resolution. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, pages 4545–4552. ELRA.

Rina Dechter and David Cohen. 2003. Constraint processing. Morgan Kaufmann.

Rina Dechter, Itay Meiri, and Judea Pearl. 1991. Temporal constraint networks. In *Artificial Intelligence*, volume 49, pages 61–95. Elsevier.

Pascal Denis and Philippe Muller. 2011. Predicting globally-coherent temporal structures from texts via endpoint inference and graph decomposition. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 1788–1793. AAAI Press.

Leon Derczynski. 2016. Representation and learning of temporal relations. In *Proceedings of the International Conference on Computational Linguistics (COLING)*, pages 1937–1948. ACL.

Leon Derczynski, Hector Llorens, and Estela Saquete. 2012. Massively increasing TIMEX3 resources: A transduction approach. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, pages 3754–3761. ELRA.

Leon Derczynski, Hector Llorens, and Naushad UzZaman. 2013. TimeML-strict: clarifying temporal annotation. In *arXiv preprint arXiv:1304.7289*.

Leon Derczynski, Jannik Strötgen, Diana Maynard, Mark A Greenwood, and Manuel Jung. 2016. GATE-Time: Extraction of temporal expressions and event. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, pages 3702–3708. ELRA.

Leon RA Derczynski. 2017. *Automatically Ordering Events and Times in Text*. Springer.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dmitriy Dligach, Timothy Miller, Chen Lin, Steven Bethard, and Guergana Savova. 2017. Neural temporal relation extraction. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, volume 2, pages 746–751.

Quang Xuan Do, Wei Lu, and Dan Roth. 2012. Joint inference for event timeline construction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 677–687. ACL.

Daxiang Dong, Hua Wu, Wei He, Dianhai Yu, and Haifeng Wang. 2015. Multi-task learning for multiple language translation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1723–1732. ACL.

Charles Dugas, Yoshua Bengio, François Bélisle, Claude Nadeau, and René Garcia. 2001. Incorporating second-order functional knowledge for better option pricing. In *Advances in Neural Information Processing Systems (NIPS)*, pages 472–478.

Jesse Dunietz, Lori Levin, and Jaime Carbonell. 2017. The BECauSE corpus 2.0: Annotating causality and overlapping relations. In *Proceedings of the Linguistic Annotation Workshop*, pages 95–104.

Lisa Ferro, Laurie Gerber, Inderjeet Mani, Beth Sundheim, and George Wilson. 2005. Tides 2005 standard for the annotation of temporal expressions. Technical report, MITRE.

Michele Filannino, Gavin Brown, and Goran Nenadic. 2013. ManTIME: Temporal expression identification and normalization in the TempEval-3 challenge. In *Proceedings of the International Workshop on Semantic Evaluation (SemEval)*, volume 2, pages 53–57. ACL.

Michael David Fisher, Dov M Gabbay, and Lluis Vila. 2005. Handbook of temporal reasoning in artificial intelligence. volume 1. Elsevier.

Christian Freksa. 1992a. Temporal reasoning based on semi-intervals. In *Artificial Intelligence*, volume 54, pages 199–227. Elsevier.

Christian Freksa. 1992b. Using orientation information for qualitative spatial reasoning. In *Theories and Methods of Spatio-temporal Reasoning in Geographic Space*, pages 162–178. Springer.

Yoav Freund and Robert E Schapire. 1999. Large margin classification using the perceptron algorithm. In *Machine Learning*, volume 37, pages 277–296. Springer.

Kunihiko Fukushima. 1980. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36(4):193–202.

Diana Galvan, Naoaki Okazaki, Koji Matsuda, and Kentaro Inui. 2018. Investigating the challenges of temporal relation extraction from clinical text. In *Proceedings of the Ninth International Workshop on Health Text Mining and Information Analysis*, pages 55–64.

Rosella Gennari and Pierpaolo Vittorini. 2016. Qualitative temporal reasoning can improve on temporal annotation quality: How and why. In *Applied Artificial Intelligence*, volume 30, pages 690–719. Taylor & Francis.

Rosella Gennari and Pierpaolo Vittorini. 2017. Time out of joint in temporal annotations of texts: Challenges for artificial intelligence and human computer interaction. In *Proceedings of the AI*IA Workshop on Deep Understanding and Reasoning*, pages 50–55. CEUR-WS. org.

Alex Graves, Greg Wayne, and Ivo Danihelka. 2014. Neural Turing machines. In *arXiv preprint arXiv:1410.5401*.

Zonghao Gu, Edward Rothberg, and Robert Bixby. 2012. Gurobi optimizer reference manual, version 5.0. In *Gurobi Optimization Inc., Houston, USA*.

Hans Werner Guesgen. 1989. *Spatial Reasoning Based on Allen's Temporal Logic*. International Computer Science Institute Berkeley.

Inc. Gurobi Optimization. 2015. Gurobi optimizer reference manual.

Andrey Gusev, Nathanael Chambers, Pranav Khaitan, Divye Khilnani, Steven Bethard, and Dan Jurafsky. 2011. Using query patterns to learn the duration of events. In *Proceedings of the Ninth International Conference on Computational Semantics*, pages 145–154. ACL.

Eun Young Ha, Alok Baikadi, Carlyle Licata, and James C Lester. 2010. NCSU: Modeling temporal relations with Markov logic and lexical ontology. In *Proceedings of the International Workshop on Semantic Evaluation (SemEval)*, pages 341–344. ACL.

Zellig S Harris. 1954. Distributional structure. *Word*, 10(2-3):146–162.

Kazuma Hashimoto, Caiming Xiong, Yoshimasa Tsuruoka, and Richard Socher. 2017. A joint many-task model: Growing a neural network for multiple NLP tasks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1923–1933. ACL.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. In *Neural Computation*, volume 9, pages 1735–1780. MIT Press.

Konrad Höffner, Sebastian Walter, Edgard Marx, Ricardo Usbeck, Jens Lehmann, and Axel-Cyrille Ngonga Ngomo. 2017. Survey on challenges of question answering in the semantic web. *Semantic Web*, 8(6):895–920.

Heng Ji, Ralph Grishman, Hoa Trang Dang, Kira Griffitt, and Joe Ellis. 2010. Overview of the TAC 2010 knowledge base population track. In *Proceedings of the Third Text Analysis Conference*, volume 3, pages 3–3.

Min Jiang, Yang Huang, Jung-wei Fan, Buzhou Tang, Josh Denny, and Hua Xu. 2015. Parsing clinical text: how good are the state-of-the-art parsers? *BMC Medical Informatics and Decision Making*, 15(1):S2.

Alistair EW Johnson, Tom J Pollard, Lu Shen, Li-wei H Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. 2016. MIMIC-III, a freely accessible critical care database. In *Scientific Data*, volume 3, page 160035. Nature Publishing Group.

Hyuckchul Jung, James Allen, Nate Blaylock, Will De Beaumont, Lucian Galescu, and Mary Swift. 2011. Building timelines from narrative clinical records: initial results based-on deep natural language understanding. In *Proceedings of the Workshop on Biomedical Natural Language Processing*, pages 146–154. ACL.

Richard M Karp. 1972. Reducibility among combinatorial problems. In *Complexity of Computer Computations*, pages 85–103. Springer.

Catherine Kerr, Terri Hoare, Paula Carroll, and Jakub Marecek. 2014. Integer-programming ensemble of temporal-relations classifiers. In *arXiv preprint arXiv:1412.1866*.

Abdulrahman Khalifa, Sumithra Velupillai, and Stephane Meystre. 2016. Utahbmi at SemEval-2016 task 12: Extracting temporal information from clinical text. *Proceedings of the International Workshop on Semantic Evaluation (SemEval)*, pages 1256–1262.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations*.

Oleksandr Kolomiyets and Marie-Francine Moens. 2010. KUL: recognition and normalization of temporal expressions. In *Proceedings of the International Workshop on Semantic Evaluation (SemEval)*, pages 325–328. ACL.

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.

Zornitsa Kozareva and Eduard Hovy. 2011. Learning temporal information for states and events. In *Proceedings of the International Conference on Semantic Computing*, pages 424–429. IEEE.

Andrei Krokhin, Peter Jeavons, and Peter Jonsson. 2003. Reasoning about temporal relations: The tractable subalgebras of Allen's interval algebra. In *Journal of the ACM*, volume 50, pages 591–640. ACM.

Natsuda Laokulrat, Makoto Miwa, and Yoshimasa Tsuruoka. 2014. Exploiting timegraphs in temporal relation classification. In *Proceedings of the Workshop on Graph-based Methods for Natural Language Processing*, pages 6–14.

Natsuda Laokulrat, Makoto Miwa, and Yoshimasa Tsuruoka. 2015. Stacking approach to temporal relation classification with temporal inference. In *Journal of Natural Language Processing*, volume 22, pages 171–196.

Egoitz Laparra, Dongfang Xu, and Steven Bethard. 2018a. From characters to time intervals: New paradigms for evaluation and neural parsing of time normalizations.

In *Transactions of the Association of Computational Linguistics*, volume 6, pages 343–356.

Egoitz Laparra, Dongfang Xu, Ahmed Elsayed, Steven Bethard, and Martha Palmer. 2018b. SemEval 2018 Task 6: Parsing time normalizations. In *Proceedings of the International Workshop on Semantic Evaluation (SemEval)*, pages 88–96. ACL.

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 the International Workshop on Semantic Evaluation (SemEval)*, pages 1292–1297.

Kenton Lee, Yoav Artzi, Jesse Dodge, and Luke Zettlemoyer. 2014. Context-dependent semantic parsing for time expressions. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, volume 1, pages 1437–1447. ACL.

Artuur Leeuwenberg and Marie-Francine Moens. 2016. KULeuven-LIIR at SemEval 2016 Task 12: Detecting narrative containment in clinical records. *Proceedings of the International Workshop on Semantic Evaluation (SemEval)*, pages 1280–1285.

Artuur Leeuwenberg and Marie-Francine Moens. 2017a. KULeuven-LIIR at SemEval-2017 Task 12: Cross-domain temporal information extraction from clinical records. In *Proceedings of the International Workshop on Semantic Evaluation (SemEval)*, pages 1030–1034. ACL.

Artuur Leeuwenberg and Marie-Francine Moens. 2017b. Structured learning for temporal relation extraction from clinical records. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 1150–1158. ACL.

Artuur Leeuwenberg and Marie-Francine Moens. 2018a. Temporal information extraction by predicting relative time-lines. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1237–1246, Brussels, Belgium. ACL.

Artuur Leeuwenberg and Marie-Francine Moens. 2018b. Word-level loss extensions for neural temporal relation classification. In *Proceedings of the International Conference on Computational Linguistics (COLING)*, pages 3436–3447. ACL.

Peifeng Li, Qiaoming Zhu, Guodong Zhou, and Hongling Wang. 2016. Global inference to Chinese temporal relation extraction. In *Proceedings of the International Conference on Computational Linguistics (COLING)*, pages 1451–1460. ACL.

Gérard Ligozat. 1996. A new proof of tractability for ORD-Horn relations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 395–401.

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.

Chen Lin, Timothy Miller, Dmitriy Dligach, Hadi Amiri, Steven Bethard, and Guergana Savova. 2018. Self-training improves recurrent neural networks performance for temporal relation extraction. In *Proceedings of the Ninth International Workshop on Health Text Mining and Information Analysis*, pages 165–176.

Chen Lin, Timothy Miller, Dmitriy Dligach, Steven Bethard, and Guergana Savova. 2016b. Improving temporal relation extraction with training instance augmentation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, page 108. ACL.

Chen Lin, Timothy Miller, Dmitriy Dligach, Steven Bethard, and Guergana Savova. 2017. Representations of time expressions for temporal relation extraction with convolutional neural networks. In *Proceedings of the Workshop on Biomedical Natural Language Processing*, pages 322–327.

Chen Lin, Timothy Miller, Dmitriy Dligach, Steven Bethard, and Guergana Savova. 2019. A BERT-based universal model for both within-and cross-sentence clinical temporal relation extraction. In *Proceedings of the Clinical Natural Language Processing Workshop*, pages 65–71.

Jimmy Lin, Miles Efron, Yulu Wang, and Garrick Sherman. 2014. Overview of the TREC-2014 microblog track. Technical report, Maryland University College Park.

Yu-Kai Lin, Hsinchun Chen, and Randall A Brown. 2013. MedTime: A temporal information extraction system for clinical narratives. In *Journal of Biomedical Informatics*, volume 46, pages S20–S28. Elsevier.

Wang Ling, Chris Dyer, Alan W Black, and Isabel Trancoso. 2015. Two/too simple adaptations of Word2Vec for syntax problems. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 1299–1304. ACL.

Xiao Ling and Daniel S Weld. 2010. Temporal information extraction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 10, pages 1385–1390.

Sijia Liu, Liwei Wang, Vipin Chaudhary, and Hongfang Liu. 2019. Attention neural model for temporal relation extraction. In *Proceedings of the Clinical Natural Language Processing Workshop*, pages 134–139, Minneapolis, Minnesota, USA. ACL.

Hector Llorens, Nathanael Chambers, Naushad UzZaman, Nasrin Mostafazadeh, James Allen, and James Pustejovsky. 2015. SemEval-2015 task 5: QA TempEval

- evaluating temporal information understanding with question answering. In *Proceedings of the International Workshop on Semantic Evaluation (SemEval)*, pages 792–800. ACL.

Hector Llorens, Leon Derczynski, Robert J Gaizauskas, and Estela Saquete. 2012. TIMEN: An open temporal expression normalisation resource. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, pages 3044–3051. ELRA.

Andrew Makhorin. 2008. GLPK (GNU linear programming kit). In *http://www.gnu.org/s/glpk/glpk.html*.

Inderjeet Mani, Marc Verhagen, Ben Wellner, Chong Min Lee, and James Pustejovsky. 2006. Machine learning of temporal relations. In *Proceedings of the International Conference on Computational Linguistics and the Annual Meeting of the Association for Computational Linguistics (COLING-ACL)*, pages 753–760. ACL.

Inderjeet Mani, Ben Wellner, Marc Verhagen, and James Pustejovsky. 2007. Three approaches to learning TLINKS in TimeML. In *Technical Report CS-07–268, Computer Science Department*.

Inderjeet Mani and George Wilson. 2000. Robust temporal processing of news. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 69–76. ACL.

David McClosky and Christopher D Manning. 2012. Learning constraints for consistent timeline extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 873–882. ACL.

Bill McDowell, Nathanael Chambers, Alexander Ororbia II, and David Reitter. 2017. Event ordering with a generalized model for sieve prediction ranking. In *Proceedings of the International Joint Conference on Natural Language Processing (IJCNLP)*, volume 1, pages 843–853. ACL.

Itay Meiri. 1996. Combining qualitative and quantitative constraints in temporal reasoning. In *Artificial Intelligence*, volume 87, pages 343–385. Elsevier.

Yuanliang Meng and Anna Rumshisky. 2018. Context-aware neural model for temporal information extraction. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, volume 1, pages 527–536. ACL.

Yuanliang Meng, Anna Rumshisky, and Alexey Romanov. 2017. Temporal information extraction for question answering using syntactic dependencies in an LSTM-based architecture. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 887–896. ACL.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

SA Miller and LK Schubert. 1990. Time revisited. In *Computational Intelligence*, volume 6, pages 108–118. Blackwell.

Timothy Miller, Steven Bethard, Dmitriy Dligach, Chen Lin, and Guergana Savova. 2015. Extracting time expressions from clinical text. In *Proceedings of the Workshop on Biomedical Natural Language Processing*, pages 81–91, Beijing, China. ACL.

Anne-Lyse Minard, Manuela Speranza, Eneko Agirre, Itziar Aldabe, Marieke van Erp, Bernardo Magnini, German Rigau, and Ruben Urizar. 2015. Semeval-2015 Task 4: Timeline: Cross-document event ordering. In *Proceedings of the International Workshop on Semantic Evaluation (SemEval)*, pages 778–786. ACL.

Seyed Abolghasem Mirroshandel and Gholamreza Ghassem-Sani. 2012. Towards unsupervised learning of temporal relations between events. In *Journal of Artificial Intelligence Research*, volume 45, pages 125–163.

Paramita Mirza. 2014. Extracting temporal and causal relations between events. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics Student Research Workshop*, pages 10–17.

Paramita Mirza. 2015. Recognizing and normalizing temporal expressions in Indonesian texts. In *Conference of the Pacific Association for Computational Linguistics*, pages 135–147. Springer.

Paramita Mirza and Sara Tonelli. 2014. An analysis of causality between events and its relation to temporal information. In *Proceedings of the International Conference on Computational Linguistics (COLING)*, pages 2097–2106. ACL.

Paramita Mirza and Sara Tonelli. 2016. Catena: Causal and temporal relation extraction from natural language texts. In *Proceedings of the International Conference on Computational Linguistics (COLING)*, pages 64–75. ACL.

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 the International Conference on Language Resources and Evaluation (LREC)*, pages 4388–4395. ELRA.

Nelson Morgan and Hervé Bourlard. 1990. Generalization and parameter estimation in feedforward nets: Some experiments. In *Advances in Neural Information Processing Systems*, pages 630–637.

Nasrin Mostafazadeh, Alyson Grealish, Nathanael Chambers, James Allen, and Lucy Vanderwende. 2016. CaTeRS: Causal and temporal relation scheme for semantic

annotation of event structures. In *Proceedings of the Workshop on Events*, pages 51–61.

Amitabha Mukerjee and Gene Joe. 1990. *A Qualitative Model for Space*. Texas A and M University. Computer Science Department.

Vinod Nair and Geoffrey E Hinton. 2010. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 807–814.

Bernhard Nebel and Hans-Jürgen Bürckert. 1995. Reasoning about temporal relations: a maximal tractable subclass of Allen's interval algebra. In *Journal of the ACM*, volume 42, pages 43–66. ACM.

Matteo Negri and Luca Marseglia. 2004. Recognition and normalization of time expressions: ITC-irst at TERN 2004. In *Rapport interne, ITC-irst, Trento*.

Jun-Ping Ng, Yan Chen, Min-Yen Kan, and Zhoujun Li. 2014. Exploiting timelines to enhance multi-document summarization. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, volume 1, pages 923–933. ACL.

Thien Huu Nguyen and Ralph Grishman. 2015. Relation extraction: Perspective from convolutional neural networks. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 39–48. ACL.

Azadeh Nikfarjam, Ehsan Emadzadeh, and Graciela Gonzalez. 2013. Towards generating a patient's timeline: extracting temporal relationships from clinical notes. In *Journal of Biomedical Informatics*, volume 46, pages S40–S47. Elsevier.

Qiang Ning, Zhili Feng, and Dan Roth. 2017. A structured learning approach to temporal relation extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1027–1037. ACL.

Qiang Ning, Zhili Feng, Hao Wu, and Dan Roth. 2018a. Joint reasoning for temporal and causal relations. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 2278–2288. ACL.

Qiang Ning, Hao Wu, Haoruo Peng, and Dan Roth. 2018b. Improving temporal relation extraction with a globally acquired statistical resource. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 841–851, New Orleans, Louisiana. ACL.

Qiang Ning, Hao Wu, and Dan Roth. 2018c. A multi-axis annotation scheme for event temporal relations. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1318–1328, Melbourne, Australia. ACL.

Qiang Ning, Zhongzhi Yu, Chuchu Fan, and Dan Roth. 2018d. Exploiting partially annotated data for temporal relation extraction. In *Proceedings of Joint Conference on Lexical and Computational Semantics (*SEM)*, pages 148–153. ACL.

Qiang Ning, Ben Zhou, Zhili Feng, Haoruo Peng, and Dan Roth. 2018e. CogCompTime: A tool for understanding time in natural language. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing: System Demonstrations (EMNLP)*, pages 72–77, Brussels, Belgium. ACL.

Feng Niu, Christopher Ré, AnHai Doan, and Jude Shavlik. 2011. Tuffy: Scaling up statistical inference in Markov logic networks using an RDBMS. In *Proceedings of the VLDB Endowment*, volume 4, pages 373–384. VLDB Endowment.

Taichi Noro, Takashi Inui, Hiroya Takamura, and Manabu Okumura. 2006. Time period identification of events in text. In *Proceedings of the International Conference on Computational Linguistics and the Annual Meeting of the Association for Computational Linguistics (COLING-ACL)*, pages 1153–1160. ACL.

Amy Olex, Luke Maffey, Nicholas Morgan, and Bridget McInnes. 2018. Chrono at SemEval-2018 Task 6: A system for normalizing temporal expressions. In *Proceedings of the International Workshop on Semantic Evaluation (SemEval)*, pages 97–101. ACL.

Agnieszka Onisko, Allan Tucker, and Marek J. Druzdzel. 2015. *Prediction and Prognosis of Health and Disease*. Springer International Publishing, Cham.

Anusri Pampari, Preethi Raghavan, Jennifer Liang, and Jian Peng. 2018. emrQA: A large corpus for question answering on electronic medical records. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2357–2368. ACL.

Feng Pan, Rutu Mulkar-Mehta, and Jerry R Hobbs. 2006a. An annotated corpus of typical durations of events. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, pages 77–82. ELRA.

Feng Pan, Rutu Mulkar-Mehta, and Jerry R Hobbs. 2006b. Learning event durations from event descriptions. In *Proceedings of the International Conference on Computational Linguistics and the Annual Meeting of the Association for Computational Linguistics (COLING-ACL)*, pages 393–400. ACL.

Feng Pan, Rutu Mulkar-Mehta, and Jerry R Hobbs. 2011. Annotating and learning event durations in text. In *Computational Linguistics*, volume 37, pages 727–752. MIT Press.

Nanyun Peng and Mark Dredze. 2015. Named entity recognition for Chinese social media with jointly trained embeddings. In *Proceedings of the Conference on*

Empirical Methods in Natural Language Processing (EMNLP), pages 548–554. ACL.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543. ACL.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 2227–2237. ACL.

Slav Petrov, Dipanjan Das, and Ryan McDonald. 2012. A universal part-of-speech tagset. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, pages 2089–2096. ELRA.

Karl Pichotta and Raymond J Mooney. 2016. Learning statistical scripts with LSTM recurrent neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 2800–2806.

James Pustejovsky, José M Castano, Robert Ingria, Roser Sauri, Robert J Gaizauskas, Andrea Setzer, Graham Katz, and Dragomir R Radev. 2003a. TimeML: Robust specification of event and temporal expressions in text. In *New directions in Question Answering*, volume 3, pages 28–34.

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

James Pustejovsky, Kiyong Lee, Harry Bunt, and Laurent Romary. 2010. ISO-TimeML: An international standard for semantic annotation. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, volume 10, pages 394–397.

James Pustejovsky and Amber Stubbs. 2011. Increasing informativeness in temporal annotation. In *Proceedings of the Linguistic Annotation Workshop*, pages 152–160. ACL.

Preethi Raghavan, James L Chen, Eric Fosler-Lussier, and Albert M Lai. 2014. How essential are unstructured clinical narratives and information fusion to clinical trial recruitment? In *Proceedings of the AMIA Summits on Translational Science*, volume 2014, pages 218–223. American Medical Informatics Association.

Livy Real, Alexandre Rademaker, Fabricio Chalub, and Valeria de Paiva. 2018. Towards temporal reasoning in Portuguese. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*. ELRA.

Nils Reimers, Nazanin Dehghani, and Iryna Gurevych. 2016. Temporal anchoring of events for the TimeBank corpus. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, volume 1, pages 2195–2204.

Nils Reimers, Nazanin Dehghani, and Iryna Gurevych. 2018. Event time extraction with a decision tree of neural classifiers. In *Transactions of the Association for Computational Linguistics*, volume 6, pages 77–89. ACL.

Matthew Richardson and Pedro Domingos. 2006. Markov logic networks. In *Machine Learning*, volume 62, pages 107–136. Springer.

Sebastian Ruder. 2017. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*.

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.

Steven Schockaert and Martine De Cock. 2008. Temporal reasoning about fuzzy intervals. In *Artificial Intelligence*, volume 172, pages 1158–1193. Elsevier.

Burr Settles. 2012. Active learning. In *Synthesis Lectures on Artificial Intelligence and Machine Learning*, volume 6, pages 1–114. Morgan & Claypool Publishers.

Andrea Setzer. 2002. *Temporal information in newswire articles: an annotation scheme and corpus study.* Ph.D. thesis, University of Sheffield.

Andrea Setzer, Robert Gaizauskas, and Mark Hepple. 2003. Using semantic inferences for temporal annotation comparison. In *Proceedings of the International Workshop on Inference in Computational Semantics*.

Andrea Setzer, Robert Gaizauskas, and Mark Hepple. 2005. The role of inference in the temporal annotation and analysis of text. In *Language Resources and Evaluation*, volume 39, pages 243–265. Springer.

Yuval Shahar. 1999. Timing is everything: Temporal reasoning and temporal data maintenance in medicine. In *Joint European Conference on Artificial Intelligence in Medicine and Medical Decision Making*, pages 30–46. Springer.

Anders Søgaard and Yoav Goldberg. 2016. Deep multi-task learning with low level tasks supervised at lower layers. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, volume 2, pages 231–235. ACL.

Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958.

Jannik Strötgen and Michael Gertz. 2010. HeidelTime: High quality rule-based extraction and normalization of temporal expressions. In *Proceedings of the International Workshop on Semantic Evaluation (SemEval)*, pages 321–324. ACL.

Jannik Strötgen and Michael Gertz. 2015. A baseline temporal tagger for all languages. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 541–547. ACL.

Jannik Strötgen and Michael Gertz. 2016. Domain-sensitive temporal tagging. In *Synthesis Lectures on Human Language Technologies*, volume 9, pages 1–151. Morgan & Claypool Publishers.

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. In *Transactions of the Association for Computational Linguistics*, volume 2, pages 143–154. MIT Press.

Weiyi Sun. 2014. Time will tell: Temporal reasoning in clinical narratives and beyond. State University of New York at Albany.

Weiyi Sun, Anna Rumshisky, and Ozlem Uzuner. 2013a. Annotating temporal information in clinical narratives. In *Journal of Biomedical Informatics*, volume 46, pages S5–S12. Elsevier.

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

Weiyi Sun, Anna Rumshisky, and Ozlem Uzuner. 2015. Normalization of relative and incomplete temporal expressions in clinical narratives. In *Journal of the American Medical Informatics Association*, volume 22, pages 1001–1008. Oxford University Press.

Yawei Sun, Gong Cheng, and Yuzhong Qu. 2018. Reading comprehension with graph-based temporal-casual reasoning. In *Proceedings of the International Conference on Computational Linguistics (COLING)*, pages 806–817. ACL.

Simon Suster and Walter Daelemans. 2018. CliCR: a dataset of clinical case reports for machine reading comprehension. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 1551–1563. ACL.

Xavier Tannier and Philippe Muller. 2011. Evaluating temporal graphs built from texts via transitive reduction. In *Journal of Artificial Intelligence Research*, volume 40, pages 375–413.

Xavier Tannier, Philippe Muller, et al. 2008. Evaluation metrics for automatic temporal annotation of texts. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, pages 150–155. ELRA.

Marta Tatu and Munirathnam Srikanth. 2008. Experiments with reasoning for temporal relations between events. In *Proceedings of the International Conference on Computational Linguistics (COLING)*, pages 857–864. ACL.

Hegler Tissot, Marcos Didonet Del Fabro, Leon Derczynski, and Angus Roberts. 2016. Normalisation of imprecise temporal expressions extracted from text. In *Knowledge and Information Systems*, pages 1–34. Springer.

Hegler Tissot, Angus Roberts, Leon Derczynski, Genevieve Gorrell, and Marcus Didonet Del Fabro. 2015. Analysis of temporal expressions annotated in clinical notes. In *Proceedings of the Workshop on Interoperable Semantic Annotation*. ACL.

Julien Tourille, Olivier Ferret, Aurelie Neveol, and Xavier Tannier. 2017a. Neural architecture for temporal relation extraction: A Bi-LSTM approach for detecting narrative containers. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 224–230, Vancouver, Canada. ACL, ACL.

Julien Tourille, Olivier Ferret, Aurelie Neveol, and Xavier Tannier. 2017b. Neural architecture for temporal relation extraction: A Bi-LSTM approach for detecting narrative containers. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, volume 2, pages 224–230.

Kristina Toutanova, Dan Klein, Christopher D Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 173–180. ACL.

Ozlem Uzuner, Andreea Bodnari, Shuying Shen, Tyler Forbush, John Pestian, and Brett R South. 2012. Evaluating the state of the art in coreference resolution for electronic medical records. *Journal of the American Medical Informatics Association*, 19(5):786–791.

Naushad UzZaman and James F. Allen. 2011. Temporal evaluation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, HLT '11, pages 351–356, Stroudsburg, PA, USA. ACL.

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 Joint Conference on Lexical and Computational Semantics and the International Workshop on Semantic Evaluation (*SEM-SemEval)*, volume 2, pages 1–9. ACL.

Siddharth Vashishtha, Benjamin Van Durme, and Aaron Steven White. 2019. Fine-grained temporal relation extraction. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems (NIPS)*, pages 5998–6008.

Alakananda Vempala, Eduardo Blanco, and Alexis Palmer. 2018. Determining event durations: Models and error analysis. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 164–168. ACL.

Marc Verhagen. 2004. Times between the lines. *Brandeis University, Massachusetts*.

Marc Verhagen. 2005. Temporal closure in an annotation environment. In *Language Resources and Evaluation*, volume 39, pages 211–241. Springer.

Marc Verhagen, Robert Gaizauskas, Frank Schilder, Mark Hepple, Graham Katz, and James Pustejovsky. 2007. SemEval-2007 Task 15: TempEval temporal relation identification. In *Proceedings of the International Workshop on Semantic Evaluation (SemEval)*, pages 75–80. ACL.

Marc Verhagen, Robert Knippen, Inderjeet Mani, and James Pustejovsky. 2006. Annotation of temporal relations with Tango. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, pages 2249–2252. ELRA.

Marc Verhagen and James Pustejovsky. 2008. Temporal processing with the TARSQI toolkit. In *Proceedings of the International Conference on Computational Linguistics (COLING)*, pages 189–192. ACL.

Marc Verhagen, Roser Sauri, Tommaso Caselli, and James Pustejovsky. 2010. SemEval-2010 Task 13: TempEval-2. In *Proceedings of the International Workshop on Semantic Evaluation (SemEval)*, pages 57–62. ACL.

Marc Vilain, Henry Kautz, and Peter Van Beek. 1990. Constraint propagation algorithms for temporal reasoning: A revised report. In *Readings in Qualitative Reasoning about Physical Systems*, pages 373–381. Elsevier.

Kenneth F Wallis. 2014. The two-piece normal, binormal, or double Gaussian distribution: its origin and rediscoveries. *Statistical Science*, pages 106–112.

Vincent Walsh. 2003. A theory of magnitude: Common cortical metrics of time, space and quantity. In *Trends in Cognitive Sciences*, volume 7, pages 483–488. Elsevier.

Zhenhua Wang, Lidan Shou, Ke Chen, Gang Chen, and Sharad Mehrotra. 2015. On summarization and timeline generation for evolutionary tweet streams. In *IEEE Transactions on Knowledge and Data Engineering*, volume 27, pages 1301–1315. IEEE.

David Weiss, Chris Alberti, Michael Collins, and Slav Petrov. 2015. Structured training for neural network transition-based parsing. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics and the International Joint Conference on Natural Language Processing (ACL-IJCNLP)*, volume 1, pages 323–333. ACL.

Jennifer Williams and Graham Katz. 2012. Extracting and modeling durations for habits and events from Twitter. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 223–227. ACL.

Dongfang Xu, Egoitz Laparra, and Steven Bethard. 2019. Pre-trained contextualized character embeddings lead to major improvements in time normalization: a detailed analysis. In *Proceedings of the Joint Conference on Lexical and Computational Semantics (*SEM)*, pages 68–74.

Rui Yan, Liang Kong, Congrui Huang, Xiaojun Wan, Xiaoming Li, and Yan Zhang. 2011. Timeline generation through evolutionary trans-temporal summarization. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 433–443. ACL.

Katsumasa Yoshikawa, Sebastian Riedel, Masayuki Asahara, and Yuji Matsumoto. 2009. Jointly identifying temporal relations with Markov logic. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics and the International Joint Conference on Natural Language Processing (ACL-IJCNLP)*, pages 405–413. ACL.

Jianfei Yu and Jing Jiang. 2016. Learning sentence embeddings with auxiliary tasks for cross-domain sentiment classification. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 236–246. ACL.

Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, Jun Zhao, et al. 2014. Relation classification via convolutional deep neural network. In *Proceedings of the International Conference on Computational Linguistics (COLING)*, pages 2335–2344. ACL.

Dongxu Zhang and Dong Wang. 2015. Relation classification via recurrent neural network. *arXiv preprint arXiv:1508.01006*.

Shu Zhang, Dequan Zheng, Xinchen Hu, and Ming Yang. 2015. Bidirectional long short-term memory networks for relation classification. In *Proceedings of the Pacific Asia Conference on Language, Information and Computation*.

Xin Wayne Zhao, Yanwei Guo, Rui Yan, Yulan He, and Xiaoming Li. 2013. Timeline generation with social attention. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1061–1064. ACM.

Li Zhou and George Hripcsak. 2007. Temporal reasoning with medical data—a review with emphasis on medical natural language processing. In *Journal of Biomedical Informatics*, volume 40, pages 183–202. Elsevier.

Peng Zhou, Wei Shi, Jun Tian, Zhenyu Qi, Bingchen Li, Hongwei Hao, and Bo Xu. 2016. Attention-based bidirectional long short-term memory networks for relation classification. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Henghui Zhu, Ioannis Ch Paschalidis, and Amir Tahmasebi. 2018. Clinical concept extraction with contextual word embedding. *arXiv preprint arXiv:1810.10566*.

List of Publications

Journal Articles

Artuur Leeuwenberg and Marie-Francine Moens. 2019. Extracting Bounded Calendar Timelines from English Clinical Reports. *IEEE/ACM Transactions on Audio, Speech, and Language Processing (TASLP)*. IEEE Press. (**submitted**).

Artuur Leeuwenberg and Marie-Francine Moens. 2019. A Survey on Temporal Reasoning for Temporal Information Extraction from Text. In *The Journal of Artificial Intelligence Research (JAIR)*, pp. 341-380. AI Access.

Peer-Reviewed International Conference Articles

Artuur Leeuwenberg and Marie-Francine Moens. 2018. Temporal Information Extraction by Predicting Relative Time-lines. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1237-1246, Brussels, Belgium. ACL.

Artuur Leeuwenberg and Marie-Francine Moens. 2018. Word-Level Loss Extensions for Neural Temporal Relation Classification. In *Proceedings of the International Conference on Computational Linguistics (COLING)*, pp. 3436-3447, Santa Fe, New-Mexico, USA. ACL.

Quynh Thi Ngoc Do, **Artuur Leeuwenberg**, Geert Heyman, and Marie-Francine Moens. 2018. A Flexible and Easy-to-Use Semantic Role Labeling Framework for Different Languages. In *Proceedings of International Conference on Computational Linguistics (COLING): System Demonstrations*, pp. 161–165, Santa Fe, New Mexico, USA.

Artuur Leeuwenberg and Marie-Francine Moens. 2017. Structured Learning for Temporal Relation Extraction from Clinical Records. In *Proceedings of the European Chapter of the Association for Computational Linguistics (EACL)*, pp. 1150-1158, Valencia, Spain. ACL.

Workshop Proceedings and Meeting Abstracts

Mariya Hendriksen, **Artuur Leeuwenberg**, and Marie-Francine Moens. 2019. LSTM for dialogue breakdown detection: Exploration of different model types and word embeddings. In *Proceedings of the International Workshop on Spoken Dialogue Systems Technology (IWSDS)*, Siracusa, Italy. Springer.

Quynh Ngoc Thi Do, **Artuur Leeuwenberg**, Geert Heyman, and Marie-Francine. 2018. How to use DAMESRL: A framework for deep multilingual semantic role labeling. In *CLARIN Annual Conference 2018*, Pisa, Italy.

Artuur Leeuwenberg and Marie-Francine Moens. 2017. KULeuven-LIIR at SemEval 2017 Task 12: Cross-Domain Temporal Information Extraction from Clinical Records. In *Proceedings of the International Workshop on Semantic Evaluation (SemEval)*, Vancouver, Canada. ACL.

Artuur Leeuwenberg and Marie-Francine Moens. 2016. KULeuven-LIIR at SemEval 2016 Task 12: Detecting Narrative Containment in Clinical Records. In *Proceedings of the International Workshop on Semantic Evaluation (SemEval)*, San Diego, California. ACL.

FACULTY OF ENGINEERING SCIENCE
DEPARTMENT OF COMPUTER SCIENCE
LANGUAGE INTELLIGENCE AND INFORMATION RETRIEVAL LAB
Celestijnenlaan 200A
B-3001 Leuven
tuur.leeuwenberg@cs.kuleuven.be
www.liir.cs.kuleuven.be