Towards Extracting Absolute Event Timelines From English Clinical Reports

Artuur Leeuwenberg and Marie-Francine Moens

***Abstract*—Temporal information extraction is a challenging but important area of automatic natural language understanding. Existing approaches annotate and extract various parts of the temporal information conveyed in language like relative event order, temporal expressions, or event durations. Most schemes focus primarily on annotation of temporally certain (often explicit) information, resulting in partial annotation, and under-representation of implicit information. In this article, we propose an approach towards extraction of more complete (implicit and explicit) temporal information for *all* events, and obtain probabilistic absolute event timelines by modeling temporal uncertainty with information bounds. As a case study, we use our scheme to annotate a set of English clinical reports, and propose and evaluate a multi-regression model for predicting probabilistic absolute timelines, obtaining promising results.**

***Index Terms*—Clinical records, implicit information, temporal information extraction, temporal uncertainty.**

I. Introduction

In THIS article, 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 [1]–[3], we focus here on the clinical domain, for which having precise temporal information is vital. High quality temporal extraction from text could be an important enrichment of the structured electronic health record, with much potential for applications [4], [5]. 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 [6]–[9], event durations [10], [11], and explicit temporal cues like temporal expressions [7], [12]–[14].

However, for a majority of events, existing schemes provide only partial event time information, leaving many event times unbounded. With a bounded event time, we mean a closed interval on the calendar timeline during which 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 work, 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.

An example of our proposed scheme is shown in Fig. 1. The bounds in our scheme model temporal uncertainty. They indicate how precisely the temporal information can be determined based on the text. Modeling temporal uncertainty is very important to deal with implicit information, which is often underrepresented in current schemes, and in timeline evaluation. For example, in Fig. 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 IV), 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 (or even the most probable time period between two events). The ability to perform such queries could be useful in practical applications and for timeline visualization.

This work makes the following contributions:

- We propose a novel annotation scheme, to annotate bounded absolute timelines, while 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.

First, we will discuss how the current work relates to existing research on temporal annotation and timeline extraction. Second, 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.

Manuscript received September 17, 2019; revised June 13, 2020 and August 28, 2020; accepted September 8, 2020. Date of publication September 28, 2020; date of current version October 8, 2020. This work was supported by the European Research Council Advanced Grant CALCULUS H2020-ERC-2017-ADG 788506, and by the IWT-SBO project ACCUMU LATE 150056. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Dilek Hakkani-Tur. *(Corresponding author: Artuur Leeuwenberg.)*

Artuur Leeuwenberg is with the Julius Center for Health Sciences, and Primary Care, University Medical Center Utrecht, Utrecht University, 3508 GA, Utrecht, The Netherlands (e-mail: aleeuw15@umcutrecht.nl).

Marie-Francine Moens is with the Department of Computer Science, KU Leuven, 3001 Leuven, Belgium (e-mail: sien.moens@cs.kuleuven.be).

Digital Object Identifier 10.1109/TASLP.2020.3027201

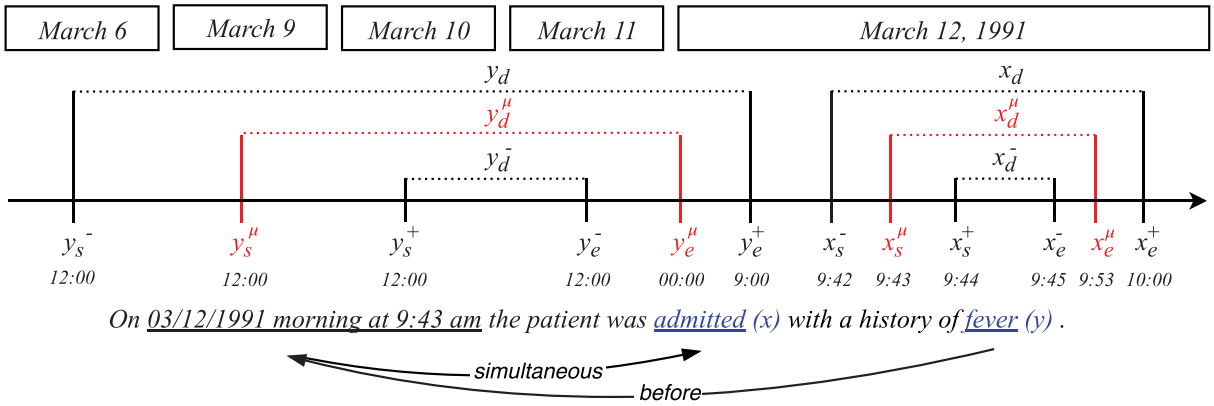

Fig. 1. An example sentence annotated with our scheme: containing events x (admitted), and y (fever), with most likely start x_s^μ, duration x_d^μ (dotted line), and end x_e^μ (all in red), and their corresponding lower and upper bounds ($^-$, and $^+$ in black). And similarly for event y. Below the sentence the existing temporal links of TimeML are shown.

II. Related Work

A. Event Position

Currently, the most widely used annotation scheme is TimeML [6], [7], in which events (e.g., *a meeting*), and temporal expressions (e.g., *yesterday*, or *02/02/2001*) are temporally linked by basic Allen interval relations,[1] [15]. Adaptations of this scheme were also annotated in several clinical corpora [8], [16], from which we use the i2b2 temporal corpus as a starting point of our work [16].[2] To extract TimeML style temporal graphs, multiple shared tasks have been organized, resulting in many systems [21]–[26]. Current state-of-the-art systems are mostly neural-network-based models [27]–[31]. [32] 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 [9], [33], 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 [34]. However, most events cannot be directly linked to such expressions, giving them no absolute anchors to the timeline. [35] address this issue and reannotate the 36 news articles from TimeBank Dense [36] with a new scheme and propose a corresponding system [37], based on a neural decision tree. Their annotations provide calendar dates for all within-day events. By 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.[3] 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 empirically compare with their work.

B. 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 of duration is present.[4] [10], [11] add explicit and implicit duration annotations to all events of the 58-document TimeBank corpus [38]. 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 Section V-A), we extend [10] 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. Various methods have been proposed to predict coarse-grained event durations in the TimeBank corpus [10], [11], [39], [40], for which the state of the art is a Long Short-Term Memory (LSTM) network ensemble [41], which we retrain on our data and adopt as a baseline.

III. The Annotation Scheme and Dataset

Our scheme annotates on the event level. For each event mention annotators have annotated two types of information: (1) the most likely (or *mode*) 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.

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

[2] These documents are a subset of MIMIC III [17] and besides temporal TimeML annotations, also carry relation annotations [18], co-reference [19], and question answering information [20] (including temporal questions), increasing the potential of this dataset for future research.

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

[4] 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.

A. Mode Event Time Components

The **timeline** is interpreted as the calendar timeline, discretized on minute level. We define the mode event time for an event x as an interval $[x_s^\mu, x_e^\mu]$ on the timeline, ranging from its most likely starting point x_s^μ, to its mode end point x_e^μ (with $x_s^\mu < x_e^\mu$).[5] The duration x_d^μ of the event is the difference between its mode starting point and the mode end point:

$$x_d^\mu = x_e^\mu - x_s^\mu \tag{1}$$

So, each event's most likely time can be fully specified by the modes of any pairwise combination of event **components** x_s, x_d, or x_e (the third component's mode can always be inferred using Eq. 1). 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.[6]

B. 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 for that component.[7] The bounds have the following properties:

$$x_s^- \leq x_s^\mu \leq x_s^+ \quad \text{(start bounds)} \tag{2}$$

$$x_e^- \leq x_e^\mu \leq x_e^+ \quad \text{(end bounds)} \tag{3}$$

$$d_{min} \leq x_d^- \leq x_d^\mu \leq x_d^+ \quad \text{(duration bounds)} \tag{4}$$

$$x_s^\mu \leq x_e^\mu \quad \text{(start before end)} \tag{5}$$

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 again infer the bounds for the third. Hence, annotators only need to annotate two components to fully specify the mode event time and all bounds.

C. 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, which after inference results in the 9 values, shown in Fig. 1.

[5]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).

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

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

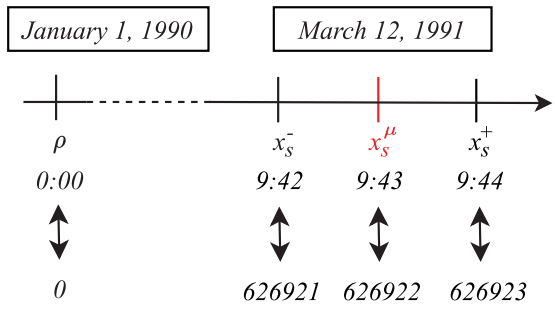

Fig. 2. Calendar times have a one-to-one mapping to regression values, which represent the number of minutes since a reference point ρ, lying in the past.

D. 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 after a fixed reference point ρ in the past. In Fig. 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 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.

IV. 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.

A. 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 [42]. As an example, two TPN distributions are shown in Fig. 4. A TPN distribution is a combination of two half normal distributions, joint at the mode. 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:

$$pdf(t) = \begin{cases} A \exp\left[-(t-\mu)^2/2\sigma_l^2\right], & t \leq \mu \\ A \exp\left[-(t-\mu)^2/2\sigma_r^2\right], & t \geq \mu \end{cases} \tag{6}$$

with scaling factor:

$$A = \left(\sqrt{2\pi}\left(\sigma_l + \sigma_r\right)/2\right)^{-1} \tag{7}$$

B. Annotations as Distributions

Because TPN distributions are asymmetric distributions that can be parameterized by exactly three values: σ_l, σ_r, and μ, they align well with our asymmetric bound annotations.[8] For each component c, consisting of mode x_c^μ, lower bound x_c^- and

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

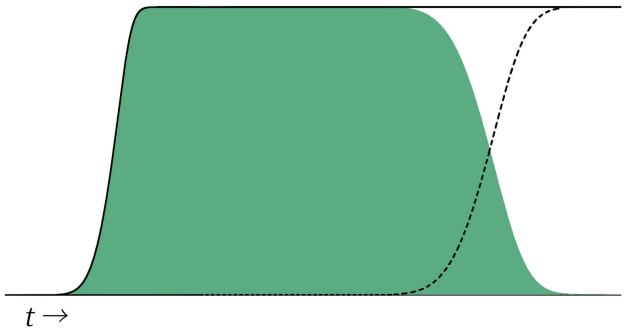

Fig. 3. Probability that some event: (1) has started before time t ($cdf_s(t)$: solid black line); (2) that it has ended before time t ($cdf_e(t)$: dashed line); and (3) is happening at time t ($cdf_s(t) - cdf_e(t)$, in green).

upper bound x_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 VI, predict mode component values, and their deviations. Hence, we can construct the corresponding TPN distributions for predicted event components in the same way.

C. 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 8, with *erf*$(\cdot)$ as the Gaussian error function.

$$cdf(t) = \begin{cases} \frac{\left(1+\text{erf}\left[\frac{t-\mu}{\sqrt{2}\sigma_l}\right]\right)\sigma_l}{\sigma_l+\sigma_r} & t \leq \mu \\ \frac{\sigma_l+\text{erf}\left[\frac{t-\mu}{\sqrt{2}\sigma_r}\right]\sigma_r}{\sigma_l+\sigma_r} & t \geq \mu \end{cases} \quad (8)$$

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 Fig. 3.

V. The Annotated Clinical Dataset

Three annotators with > 3 years of study in Biomedicine annotated in total 169[9] English clinical reports from the i2b2 temporal shared task [16]. Dataset statistics are given in Table I. The documents are already annotated with TimeML from which we adopt event span annotations, on top of which we annotate our scheme.

[9]Existing benchmark temporal corpus sizes range between 37 documents with 1,729 events [36] to 500 documents with 15,769 events [8].

TABLE I
Statistics on Full Dataset $A_1 \cup A_2 \cup A_3$, and the Subset Annotated by at Least Two Annotators A^2

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

TABLE II
For All 9 Annotated Components, We Show the Distribution (in %) of the Number of Events that is Annotated With a Value of a Certain Order of Magnitude

	Start			End			Duration		
	μ	σ_l	σ_r	μ	σ_l	σ_r	μ	σ_l	σ_r
hours	0	67	0	0	66	0	63	66	43
days	0	16	70	0	16	68	17	19	36
weeks	0	6	13	0	6	14	7	3	7
months	0	6	6	0	6	6	7	6	5
years	0	3	6	0	3	5	4	3	4
decades	100	3	5	100	4	6	3	3	5

We have built a new annotation tool (a screenshot can be found at the end of this article in Fig. 7). Besides temporal inference, the tool provides insight to the annotators about their own annotations by visualization of the mode timeline. Additionally, it 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 document of the TimeML annotations [16]. The annotators regularly discussed difficulties "in person" with the adjudicator, and used a shared document to establish agreement on difficult cases.

A. Dataset Analysis

We analyzed the annotated values regarding to order of magnitude. This is shown in Table II. 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 interesting observation 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 in the clinical narrative events are often chronologically ordered. This can result in the fact that while reading, annotators have more knowledge about past events, 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

TABLE III
AGREEMENT PERCENTAGES FOR THE DIFFERENT 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). THE SCORES FOR THE 17% SUBSET OF EVENTS THAT WERE ALREADY COARSE-BOUNDED BY THE EXISTING TIMEML-ANNOTATIONS ARE GIVEN IN-BETWEEN BRACKETS

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

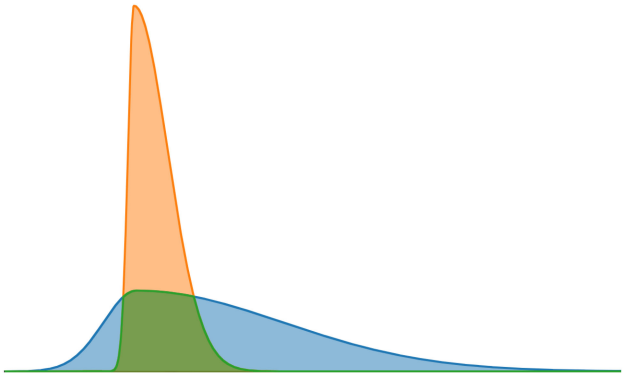

Fig. 4. Two two-piece normal distributions for the same event's start time: overlap $P_s^{\cap} = 0.38$ (in green).

high uncertainty the left deviation approaches 0, while the right deviation can grow to, in principle, forever. 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 III. We will now discuss the used metrics one-by-one.

B. Overlap Agreement ($P^{\cap}$)

Our first metric to calculate IAA between two annotators is obtained as the proportion of overlap between the TPN-distributions for each component (intersection over union).[10] A visualization of this metric is shown in Fig. 4. This metric takes into account all components of the annotations in a single score (left bound, mode, and right bound), and is therefore 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 taken 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; around 10^8 minutes). We discuss this further in Section V-F.

[10][34] use a similar metric, based on fuzzy sets instead of TPN distributions, to study imprecise timexes.

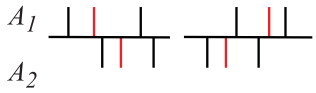

Fig. 5. An example of inclusion $P^{\in}$: on the left, the mode value (red) of A_2 is included in the bounds of A_1 (agreement); on the right, neither mode value is included within the bounds of the other (disagreement).

C. 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 (illustrated in Fig. 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 Fig. 5.

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

To analyze IAA with regard to relative event order, per annotator we consider all pairs of events in each document, 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 [9] 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.

E. Agreement With TimeML (P^{tl})

To be able to better compare our timeline annotations with the existing TimeML annotations, we follow the strategy of [32] to evaluate relative timelines using TimeML. Based on the timeline, We assign each TimeML-annotated event pair a temporal link (TLink), and calculate accuracy with the originally annotated TLinks. For this, we use the merged TLinks present in the data (*before*, *after*, and *overlap*) [16]. Following the annotation guidelines of [16] as close as possible, we use the following classification function to assign TLink types to event-event and

event-timex pairs:

$$R(x,y) = \begin{cases} before & \text{iff } x_s^\mu < y_s^\mu \\ after & \text{iff } x_e^\mu > y_e^\mu \\ overlap & \text{iff } x_s^\mu \geq y_s^\mu \wedge x_e^\mu \leq y_e^\mu \end{cases} \quad (9)$$

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.

F. Changing Bound Granularity

As can be seen from Table III, 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 the 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 III, we observe much higher agreement, especially for inclusion agreement $P^\in$. This indicates that on a more coarse grained level the annotators agree well on event position, and duration. We use these coarse bounded timelines, with high agreement, as our final data for model construction.

VI. Absolute Timeline Model (ATLM)

For each event, our model predicts the mode start time, the mode duration, and their corresponding bounds (from which the end time and bounds automatically follow).[11] 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 Fig. 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.

A. Word Representations

We experiment with two types of word representations: (1) 300-dimensional GloVe embeddings [43] trained on 300 M words from the clinical MIMIC III dataset [17], and (2) ELMo embeddings [44], in particular the embeddings by [45], which are trained on the same clinical dataset. We use ELMo for its ability to capture character-level information, important for encoding temporal expressions [46].

B. 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),

[11]Coarse start times and durations yielded higher IAA than end times.

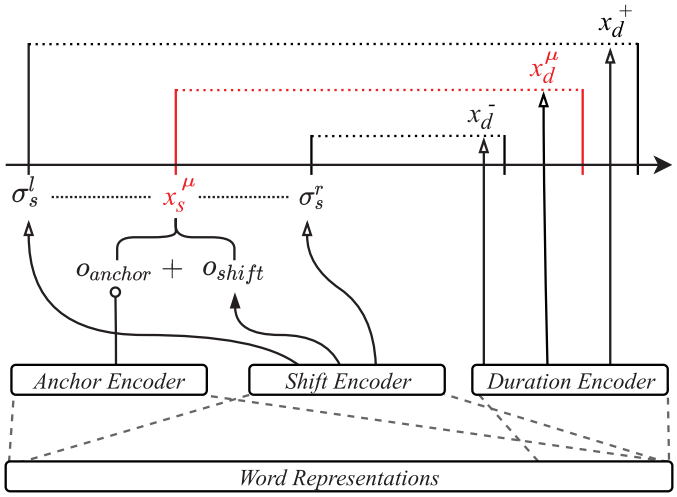

Fig. 6. A schematic overview of our model, which predicts the start and duration modes of each event, and the corresponding bounds from the input sentence.

as this has shown to be effective features for estimating event duration [41]. We encode the event and its context using either an LSTM [47] or CNN[12] [48]. 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 VI-D).

C. 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.

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 IV-B.

D. Regression Layers

In this section, we explain the meaning of the arrows in Fig. 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

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

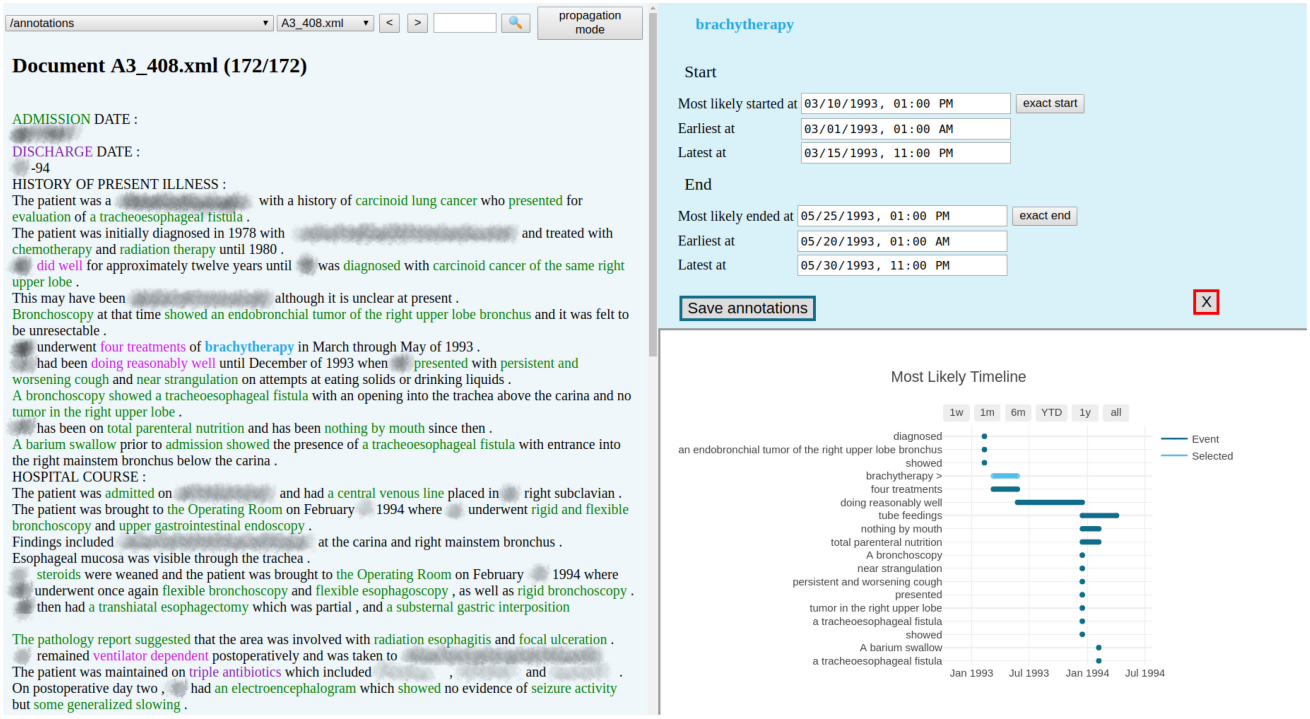

Fig. 7. A screenshot of the annotation tool. On the left the text is displayed, with colored event expressions (depending on their annotation status). On the top right, the annotated values can be entered, and on the bottom right the mode timeline is visualized.

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 date-time value of the predicted anchor.

E. 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 use the L_1 loss as given by Equation 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.[13]

$$L_1 = \frac{1}{N}\sum_{i=1}^{N} l(x_i) \tag{10}$$

$$l(x) = \sum_{c}^{C} |\hat{x}_c^{\mu} - x_c^{\mu}| + |\hat{x}_c^{\sigma_l} - x_c^{\sigma_l}| + |\hat{x}_c^{\sigma_r} - x_c^{\sigma_r}| \tag{11}$$

For minimization we use Adam [49] 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 validation set for early stopping [50], with a patience value of 20.

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

VII. 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.

A. Evaluation

Our annotated corpus is split into a 132-document training set (10,431 events) and 37-document test set (2,451 events). The test set consists of all documents that have been annotated by more than one annotator. This way, the agreement scores 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 IV-B. For evaluation, we calculate measures proposed in Section V-A. Hyper-parameters are tuned on a small 15-document development set (from the training data).

B. Baselines

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

1) Event Duration Baseline (D-LSTM): As this is the first clinical corpus to annotate full event durations, as a baseline

TABLE IV
RESULTS OF THE DIFFERENT ABSOLUTE TIMELINE MODELS ON THE TEST SET. THE SCORES FOR THE 17% SUBSET OF EVENTS THAT WERE ALREADY BOUNDED BY THE EXISTING TIMEML-ANNOTATIONS ARE GIVEN IN-BETWEEN BRACKETS. 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

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

used to predict event duration we implement the current state-of-the-art model for news texts by [41]. Their model is an LSTM-ensemble built on top of GloVe embeddings. To adapt their model to the clinical domain, we retrain the GloVe embeddings on 100 M words of clinical reports from MIMIC III [17]. As [41] 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, explained in Section VI-D, to predict the duration mode, and its left and right deviations.

2) 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 retrain a neural state-of-the-art clinical TLink extraction model [51] that is publicly available 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 [32]: (1) Each event is assigned an interval with start variable x^{μ}_s and end variable x^{μ}_e. (2) The variables are set such that the predicted TLinks between the events are satisfied on the timeline.[14] 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 9, modeling pointwise order ($a < b$) as a margin-based hinge loss: $max(a + m - b, 0)$, with a margin m of 1 minute. Equality ($=$) is modeled with an L1 loss: $|a - b|$. As events are also linked to TIMEXES, we assign two fixed constants x^{μ}_s, and x^{μ}_e 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 [49].

VIII. RESULTS AND ANALYSIS

From Table IV, 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 $p < 0.0001$[15]). This can be expected, as TL2ATL uses 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 $p < 0.01$[15]). 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 ($p < 0.0001$[15]), which suggests that ELMo embeddings seem better at capturing duration information. Because for both start and duration, the best model uses 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 II). It indicates that the models have more difficulty with events with longer durations,

[14] In our experiments the predicted TLinks were satisfied for 95% due to inconsistencies in the predicted graphs.

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

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 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 model development.

IX. Conclusion

In this article, 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 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 benchmark for further development in this area of research.[16]

We strongly believe future research into model development for this task is required, and extension of this work to other domains is a valuable avenue of investigation.

Acknowledgment

The authors would like to thank the reviewers for their constructive comments to improve the paper. Also, we thank Krupa Shah, Eva Verdonck, and Marie Lamberigts for their careful annotation work.

References

[1] R. Campos, G. Dias, A. M. Jorge, and A. Jatowt, "Survey of temporal information retrieval and related applications," in *Proc. ACM Comput. Surv.*, vol. 47, no. 2. ACM, 2015, p. 15.

[2] K. Höffner, S. Walter, E. Marx, R. Usbeck, J. Lehmann, and A.-C. Ngonga Ngomo, "Survey on challenges of question answering in the Semantic Web," in *Proc. Semantic Web*, vol. 8, no. 6. IOS Press, 2017, pp. 895–920.

[3] J.-P. Ng, Y. Chen, M.-Y. Kan, and Z. Li, "Exploiting timelines to enhance multi-document summarization," in *Proc. Annu. Meet. Assoc. Comput. Linguist. (ACL)*, vol. 1. ACL, 2014, pp. 923–933.

[16]The model code, dataset, annotation tool, guidelines, and evaluation scripts are available at: [Online]. Available: https://liir.cs.kuleuven.be

[4] Y. Shahar, "Timing is everything: Temporal reasoning and temporal data maintenance in medicine," in *Proc. Joint Eur. Conf. Artif. Intell. Med. Med. Decision Making*. Springer, 1999, pp. 30–46.

[5] H. Jung, J. Allen, N. Blaylock, W. De Beaumont, L. Galescu, and M. Swift, "Building timelines from narrative clinical records: Initial results based-on deep natural language understanding," in *Proc. Workshop Biomed. Natural Lang. Process.* ACL, 2011, pp. 146–154.

[6] J. Pustejovsky *et al.*, "TimeML: Robust specification of event and temporal expressions in text," in *Proc. New Directions Question Answering*, vol. 3, 2003, pp. 28–34.

[7] J. Pustejovsky, K. Lee, H. Bunt, and L. Romary, "ISO-TimeML: An international standard for semantic annotation," in *Proc. Int. Conf. Lang. Res. Evaluation*, vol. 10, 2010, pp. 394–397.

[8] W. F. Styler IV *et al.*, "Temporal annotation in the clinical domain," in *Trans. Assoc. Comput. Linguist.*, vol. 2. MIT Press, 2014, p. 143.

[9] Q. Ning, H. Wu, and D. Roth, "A multi-axis annotation scheme for event temporal relations," in *Proc. Annu. Meet. Assoc. for Comput. Linguist. (ACL)*. Melbourne, Australia: ACL, 7 2018, pp. 1318–1328.

[10] F. Pan, R. Mulkar-Mehta, and J. R. Hobbs, "An annotated corpus of typical durations of events," in *Proc. Int. Conf. Lang. Res. Evaluation*. ELRA, 2006, pp. 77–82.

[11] F. Pan, R. Mulkar-Mehta, and J. R. Hobbs, "Annotating and learning event durations in text," in *Comput. Linguist.*, vol. 37, no. 4. MIT Press, 2011, pp. 727–752.

[12] A. Setzer, "Temporal information in newswire articles: An annotation scheme and corpus study," Ph.D. dissertation, University of Sheffield, 2002.

[13] L. Ferro, L. Gerber, I. Mani, B. Sundheim, and G. Wilson, "Tides 2005 standard for the annotation of temporal expressions," MITRE, Tech. Rep., 2005.

[14] S. Bethard and J. Parker, "A semantically compositional annotation scheme for time normalization," in *Proc. Int. Conf. Lang. Res. Evaluation*. Paris, France: ELRA, 5 2016.

[15] J. F. Allen, "Maintaining knowledge about temporal intervals," in *Commun. ACM*. ACM, 1983, pp. 832–843.

[16] W. Sun, A. Rumshisky, and O. Uzuner, "Annotating temporal information in clinical narratives," in *J. Biomed. Inf.*, vol. 46. Elsevier, 2013, pp. S5–S12.

[17] A. E. Johnson *et al.*, "MIMIC-III, a freely accessible critical care database," in *Proc. Sci. Data*, vol. 3. Nature Publishing Group, 2016, p. 160035.

[18] W. Sun, A. Rumshisky, and O. Uzuner, "Evaluating temporal relations in clinical text: 2012 i2b2 challenge," *J. Amer. Med. Inf. Assoc.*, vol. 20, no. 5, pp. 806–813, 2013.

[19] O. Uzuner, A. Bodnari, S. Shen, T. Forbush, J. Pestian, and B. R. South, "Evaluating the state of the art in coreference resolution for electronic medical records," *J. Amer. Med. Informat. Assoc.*, vol. 19, no. 5, pp. 786–791, 2012.

[20] A. Pampari, P. Raghavan, J. Liang, and J. Peng, "emrQA: A large corpus for question answering on electronic medical records," in *Proc. Conf. Empirical Methods Natural Language Process.* ACL, 2018, pp. 2357–2368.

[21] M. Verhagen, R. Gaizauskas, F. Schilder, M. Hepple, G. Katz, and J. Pustejovsky, "SemEval-2007 Task 15: TempEval temporal relation identification," in *Proc. Int. Workshop Semantic Evaluation (SemEval)*. ACL, 2007, pp. 75–80.

[22] M. Verhagen, R. Sauri, T. Caselli, and J. Pustejovsky, "SemEval-2010 Task 13: TempEval-2," in *Proc. Int. Workshop Semantic Evaluation (SemEval)*. ACL, 2010, pp. 57–62.

[23] N. UzZaman, H. Llorens, L. Derczynski, J. Allen, M. Verhagen, and J. Pustejovsky, "Semeval-2013 Task 1: TempEval-3: Evaluating time expressions, events, and temporal relations," in *Proc. Joint Conf. Lexical Comput. Semantics Int. Workshop Semantic Evaluation (*SEM-SemEval)*, vol. 2. ACL, 2013, pp. 1–9.

[24] S. Bethard, L. Derczynski, G. Savova, J. Pustejovsky, and M. Verhagen, "Semeval-2015 Task 6: Clinical TempEval," in *Proc. Int. Workshop Semantic Evaluation (SemEval)*. ACL, 2015, pp. 806–814.

[25] S. Bethard, G. Savova, W.-T. Chen, L. Derczynski, J. Pustejovsky, and M. Verhagen, "Semeval-2016 Task 12: Clinical TempEval," in *Proc. Int. Workshop Semantic Evaluation (SemEval)*. ACL, 2016, pp. 1052–1062.

[26] S. Bethard, G. Savova, M. Palmer, and J. Pustejovsky, "SemEval-2017 Task 12: Clinical TempEval," in *Proc. Int. Workshop Semantic Evaluation (SemEval)*. Vancouver, Canada: ACL, 8 2017, pp. 565–572.

[27] J. Tourille, O. Ferret, A. Neveol, and X. Tannier, "Neural architecture for temporal relation extraction: A Bi-LSTM approach for detecting narrative containers," in *Proc. Annu. Meet. Assoc. for Comput. Linguist. (ACL)*, ACL. Vancouver, Canada: ACL, Jul. 2017, pp. 224–230.

[28] Q. Ning, Z. Feng, and D. Roth, "A structured learning approach to temporal relation extraction," in *Proc. Conf. Empirical Methods Natural Language Process*. ACL, 2017, pp. 1027–1037.
[29] Y. Meng and A. Rumshisky, "Context-aware neural model for temporal information extraction," in *Proc. Annu. Meet. Assoc. for Comput. Linguist.*, vol. 1. ACL, 2018, pp. 527–536.
[30] S. Liu, L. Wang, V. Chaudhary, and H. Liu, "Attention neural model for temporal relation extraction," in *Proc. Clinical Natural Language Process. Workshop*. Minneapolis, Minnesota, USA: ACL, Jun. 2019, pp. 134–139.
[31] G. Alfattni, N. Peek, and G. Nenadic, "Extraction of temporal relations from clinical free text: A systematic review of current approaches," *J. Biomed. Informat.*, p. 103488, 2020.
[32] A. Leeuwenberg and M.-F. Moens, "Temporal information extraction by predicting relative time-lines," in *Proc. Conf. Empirical Methods Natural Language Process*. Brussels, Belgium: ACL, Oct. 2018, pp. 1237–1246.
[33] S. Vashishtha, B. Van Durme, and A. S. White, "Fine-grained temporal relation extraction," in *Proc. Annu. Meet. Assoc. for Comput. Linguist. (ACL)*, 2019.
[34] H. Tissot, M. D. Del Fabro, L. Derczynski, and A. Roberts, "Normalisation of imprecise temporal expressions extracted from text," in *Proc. Knowledge Inform. Syst.* Springer, 2016, pp. 1–34.
[35] N. Reimers, N. Dehghani, and I. Gurevych, "Temporal anchoring of events for the TimeBank corpus," in *Proc. Annu. Meet. Assoc. for Comput. Linguist. (ACL)*, vol. 1, 2016, pp. 2195–2204.
[36] T. Cassidy, B. McDowell, N. Chambers, and S. Bethard, "An annotation framework for dense event ordering," in *Proc. Annu. Meet. Assoc. Comput. Linguist. (ACL)*, vol. 2. ACL, 2014, pp. 501–506.
[37] N. Reimers, N. Dehghani, and I. Gurevych, "Event time extraction with a decision tree of neural classifiers," in *Trans. Assoc. Comput. Linguist.*, vol. 6. ACL, 2018, pp. 77–89.
[38] J. Pustejovsky *et al.*, "The TimeBank corpus," in *Proc. Corpus Linguist.*, vol. 2003. Lancaster, UK, 2003, p. 40.
[39] A. Gusev, N. Chambers, P. Khaitan, D. Khilnani, S. Bethard, and D. Jurafsky, "Using query patterns to learn the duration of events," in *Proc. Ninth Int. Conf. Comput. Semantics*. ACL, 2011, pp. 145–154.
[40] J. Williams and G. Katz, "Extracting and modeling durations for habits and events from Twitter," in *Proc. Annu. Meet. Assoc. for Comput. Linguist. (ACL)*. ACL, 2012, pp. 223–227.
[41] A. Vempala, E. Blanco, and A. Palmer, "Determining event durations: Models and error analysis," in *Proc. Annu. Conf. North Amer. Chapter Assoc. for Comput. Linguist. (NAACL)*. ACL, 2018, pp. 164–168.
[42] K. F. Wallis, "The two-piece normal, binormal, or double Gaussian distribution: Its origin and rediscoveries," *Statist. Sci.*, pp. 106–112, 2014.
[43] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proc. Conf. Empirical Methods Natural Language Process*. ACL, 2014, pp. 1532–1543.
[44] M. Peters *et al.*, "Deep contextualized word representations," in *Proc. Annu. Conf. North Amer. Chapter Assoc. for Comput. Linguist.*. ACL, 2018, pp. 2227–2237.
[45] H. Zhu, I. C. Paschalidis, and A. Tahmasebi, "Clinical concept extraction with contextual word embedding," in *Proc. NeurIPS Mach. Learn. Health Workshop*, 2018.
[46] D. Xu, E. Laparra, and S. Bethard, "Pre-trained contextualized character embeddings lead to major improvements in time normalization: A detailed analysis," in *Proc. Joint Conf. Lexical and Comput. Semantics (*SEM)*, 2019, pp. 68–74.
[47] S. Hochreiter and J. Schmidhuber, "Long short-term memory," in *Neural Comput.*, vol. 9, no. 8. MIT Press, 1997, pp. 1735–1780.
[48] K. Fukushima, "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," *Bio. Cybern.*, vol. 36, no. 4, pp. 193–202, 1980.
[49] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learning Representations*, 2014.
[50] N. Morgan and H. Bourlard, "Generalization and parameter estimation in feedforward nets: Some experiments," in *Proc. Adv. Neural Inf. Process. Syst.*, 1990, pp. 630–637.
[51] A. Leeuwenberg and M.-F. Moens, "Word-level loss extensions for neural temporal relation classification," in *Proc. Int. Conf. Comput. Linguist. (COLING)*. ACL, 8 2018, pp. 3436–3447.

Artuur (Tuur) Leeuwenberg received the B.Sc. degree in artificial intelligence from Utrecht University, Utrecht, Netherlands and the M.Sc. degree in language & communication technologies from the University of Lorraine, Nancy, France and Saarland University, Saarbrcken, Germany. He received the Ph.D. degree in computer science from KU Leuven, Leuven, Belgium, in 2019. He is an Assistant Professor with the University Medical Center Utrecht, Utrecht, The Netherlands. His main research focus is on developing new methods for clinical risk modeling and natural language processing using machine learning.

Marie-Francine (Sien) Moens received the Ph.D. degree in computer science from KU Leuven, Leuven, Belgium, in 1999. She is a Professor with the Department of Computer Science, KU Leuven. She is the Director of the Language Intelligence and Information Retrieval (LIIR) research lab, Leuven, Belgium, a Member of the Human Computer Interaction group, and Head of the Informatics section. Her main direction of research is the development of novel methods for automated content recognition in text and multimedia using statistical machine learning.