Relative and Incomplete Time Expression Anchoring for Clinical Text

Louise Dupuis
IoPPN, King's College London
Université Paris-Saclay, CentraleSupélec
louise.dupuis@kcl.ac.uk

Nicol Bergou
IoPPN, King's College London
nicol.2.bergou@kcl.ac.uk

Hegler Tissot
IHI, University College London
h.tissot@ucl.ac.uk

Sumithra Velupillai
IoPPN, King's College London
sumithra.velupillai@kcl.ac.uk

Abstract

Extracting and modeling temporal information in clinical text is an important element for developing timelines and disease trajectories. Time information in written text varies in preciseness and explicitness, posing challenges for NLP approaches that aim to accurately anchor temporal information on a timeline. Relative and incomplete time expressions (RI-Timexes) are expressions that require additional information for their temporal anchor to be resolved, but few studies have addressed this challenge specifically. In this study, we aimed to reproduce and verify a classification approach for identifying anchor dates and relations in clinical text, and propose a novel relation classification approach for this task.

1 Introduction

Temporal information is a crucial aspect of the analysis of clinical texts in electronic health records in order to improve understanding of disease trajectories. Being able to extract and model time information, such as dates and durations of events, leads to knowledge about the temporal context of clinically important information like symptoms or treatments, and can be used e.g. to reconstruct a patient's timeline. With such timelines, a wide range of applications can be developed, such as population-based observational retrospective studies on temporal patterns in diseases and treatments, or individualised patient summaries.

Several solutions have been proposed and developed to extract and normalize temporal information from text both in the general and clinical NLP domains (Leeuwenberg and Moens, 2019; Derczynski, 2017; Tissot et al., 2019). The most widely used model for annotating temporal information and cues for NLP applications is the TimeML model (Pustejovsky et al., 2010), where *time expressions* (timexes) are a core element. These are typically annotated into types (e.g. dates, durations) and normalized to a temporal value that can be used for further temporal reasoning.

However, accurate normalization of relative and incomplete temporal expressions is still an understudied area. Relative and incomplete time expressions (RI-Timexes), as defined in (Sun et al., 2015) are time expressions that require another timex for their value to be resolved. For example, in the following sentences *"He arrived on 09/18/2002. Three days later, he was transferred to the Medical Intensive Care Unit."*, the normalized temporal value of the date timex *"09/18/2002"* does not depend on any context. Such expressions are called *absolute timexes*. *RI-Timexes*, on the other hand, require additional contextual information. For example, to assign and compute the normalized temporal value of the RI-Timex *"Three days later"*, we need information about what this expression refers to in the narrative – in this case, the previous date timex *"09/18/2002"*. The temporal expression that the relative timex refers to is called the *anchor*. An anchor relation, which specifies the link between the two expressions, can also be defined. With these two pieces of information, it becomes feasible to compute a normalized value for the RI-Timex (09/21/2002 in this case).

In the clinical domain, two of the most widely used temporal extraction and normalization tools are the java-based libraries HeidelTime (Strötgen and Gertz, 2010) and SUTime (Chang and Manning, 2012). Their approach to normalize relative time expressions is to define one main anchor date for the whole document (Document Creation Time,

Proceedings of the 3rd Clinical Natural Language Processing Workshop, pages 117–129
November 19, 2020. ©2020 Association for Computational Linguistics

DCT), and all timexes in the document are resolved relatively to this. This method might work well on e.g. short texts that refer to a single event, but is not necessarily appropriate for longer narrative notes, for example clinical assessments relating to a patient's history.

Adaptations and variations of these systems were used by several teams in the 2012 i2b2 NLP Challenge on Temporal Relations in Clinical Data (Sun et al., 2013a), and the best performing system on timex normalization yielded value accuracy of 0.73 (Sun et al., 2013b). In the analysis of these results, it was found that relative time expressions were a major source of submitted system's errors. In the proposed solutions, the main approaches relied on either defining a single anchor date (the DCT) for the whole text, or creating a set of rules that anchors expressions based on specific signal words, such as *"operation"*, or *"birth"* (Sun et al., 2015). According to Leeuwenberg and Moens (2019), almost all current state-of-the-art NLP systems use handcrafted rules based on lexical patterns to solve timex normalization. However, such rules have limitations, and not many deal with anchoring RI-Timexes in clinical notes.

One study specifically addresses the problem of anchoring RI-Timexes (Sun et al., 2015). They propose two simplification hypotheses, for identifying and classifying the anchor date and anchor relation respectively: they restrict the anchor date possibilities to four different temporal expressions (admission date, discharge date, previous timex, and previous absolute timex), and the anchor relation to three possibilities (before, after, and equal/during). This allows them to approach the problem as a multi-class classification problem. They manually annotated three corpora following these hypotheses, and proposed a supervised machine-learning approach, along with a rule-based approach for the final relative value normalization.

To our knowledge, there have been no further studies on alternative approaches for identifying and classifying anchor dates and relations for RI-Timexes in clinical text. Our long-term goal is to develop approaches for modelling time information that can be used for clinical timeline reconstruction, for which novel approaches for identifying, anchoring and normalizing RI-Timexes are needed. Our contribution in this study is a) we aimed to reproduce the findings published in Sun et al. (2015), allowing to verify the viability of their hypothesis, and to define a baseline against which we could compare new approaches; b) we propose an alternative annotation model for anchoring RI-Timexes, and developed and applied a new, adapted, annotation model; and c) we propose a new computational approach and model the problem as a relation classification problem, using a BERT transformer model (Devlin et al., 2019) trained on clinical data (Alsentzer et al., 2019).[1]

2 Materials and Methods

2.1 Data

We used the 2012 i2b2 NLP temporal challenge dataset (Sun et al., 2013a). This data is a subset of the MIMIC III database (Johnson et al., 2016), which contains de-identified electronic health record data associated with over 40K patients admitted to the Beth Israel Deaconess Medical Center in Boston, Massachusetts between 2001 and 2012, available under a data use agreement.

2.1.1 2012 TIMEX i2b2

The 2012 i2b2 data set contains 310 discharge summaries, annotated with time expressions, events, and temporal links in the TimeML format (Pustejovsky et al., 2003). They contain an 'Admission' section, which usually presents the clinical history of the patient and the reason for their hospitalisation, and a 'Discharge' section, which summarizes the course of the hospital stay (annotated as SECTIME). The annotation guidelines are presented in (Sun et al., 2013a). For our study, we only used the timex annotations, of which there are 4185 in total, out of which 2,992 are dates and times. The dataset is divided into a training set of 190 documents and a test set of 120 documents.

2.1.2 2015 RI-TIMEX i2b2 subset

We also had access to RI-Timex annotations from Sun et al. (2015), for a subset of the 2012 i2b2 data set (henceforth called 2015 RI-Timex i2b2 subset). These annotations specify anchor dates and anchor relations for 484 relative and incomplete temporal expressions, for 104 documents from the 2012 i2b2 data set (all of which are part of the 2012 i2b2 test set). The data was annotated based on the following assumptions:

1. The anchor date for a RI-Timex is either one of the section times (i.e. the 'Admission Date'

[1] Annotation guidelines and code are available at https://github.com/KCL-Health-NLP/NeuralTime

or the ‘Discharge Date’), the previous timex or the previous absolute timex. Here, “previous” is to be understood as “when going backward in the text” – the “previous timex” is the temporal expression that comes directly before the RI-Timex in the text, the previous absolute timex is the first of the previous expressions to be an absolute timex. Note that these four possibilities are not mutually exclusive, as the previous timex can be the previous absolute timex as well.

2. Anchor relations are restricted to three possibilities: ‘Before’, ‘Equal’ or ‘After’ the anchor date.

The annotations were generated through the following process: a) to isolate the RI-Timexes, they applied a pattern-based filter on all timexes annotated with the types ‘date’ and ‘time’ to identify absolute timexes; b) the remaining timexes were manually reviewed to identify RI-Timexes; c) each identified RI-Timex was then assigned an anchor date that could be ‘Admission’, ‘Dicharge’, ‘Previous Timex’, or ‘Previous Absolute Timex’, and an anchor relation ‘Before’, ‘Equal’ or ‘After’; d) an ‘Other’ category exists for cases where none of the four possibilities works. In particular, they chose the anchor date and relation using a limited context window containing the neighboring sentences. Tables 1 and 2 show the distribution of the anchor date types and anchor relation categories. For our study, we randomly divided these 484 annotations into a training set and a test set, respectively covering 411 and 73 examples. Note that sometimes, the discharge, previous timex and previous absolute timex could refer to the same actual timex, which is why the numbers of anchor relations in the table add up to more than the total of RI-Timexes.

	A	**D**	**PT**	**PAT**	**Other**	$\sum$
Training	246	81	143	118	6	594
Test	43	14	27	21	1	106
Total	289	95	170	139	7	700

Table 1: Anchor date type distribution: 2015 RI-Timex i2b2 data. A: Admission Date: D: Discharge Date: PT: Previous Timex; PAT: Previous Absolute Timex

2.2 2020 RI-TIMEX i2b2: Corpus development

To generate a new gold standard with RI-Timex anchor date type and relation annotations on the entire 2012 i2b2 data set, we defined a new annotation model to represent these concepts, which allowed us to not limit ourselves to only the four anchor date possibilities defined by Sun et al. (2015).

	Before	**Equal**	**After**	**None**	$\sum$
Training	51	169	185	6	411
Test	8	28	36	1	73
Total	59	197	221	7	484

Table 2: Anchor relation annotation distribution: 2015 RI-Timex i2b2 data.

2.2.1 Absolute timex filtering

To identify potential RI-Timexes, we started by reproducing the method of filtering out the most common absolute timexes. Following Sun et al. (2015)’s methodology, we applied this filtering only to the timexes of type ‘Date’ and ‘Time’. The total number of ‘Date’ and ‘Time’ timexes in the 2012 i2b2 dataset is 2,992, and 586 SECTIMEs (3.578 in total). The format we defined as representative of absolute timexes to filter out are, with “x” as a digit includes:

a) xx/xx/xx

b) xx/xx/xxxx

c) xx/xx

d) x/xx

e) the four previous format with ‘-‘ instead of ‘/’

f) all other expressions similar to x:xx

After this first filtering step, we obtained 1668 absolute timexes filtered and 1324 relative timexes.

2.2.2 Annotation guidelines

The goal of this annotation task is to differentiate absolute from RI-Timexes, and to link the latter to anchor dates. In our annotation guidelines, we define the following concepts:

Absolute time expression: an expression which contains all the information needed to normalize it to a standard date, e.g. “12/05/2020”;

Relative time expression: an expression whose temporal meaning is stated as a relative value against another time expression, e.g. “two days” in “two days before the admission”;

Incomplete time expression: an expression which holds only partial information : the context is needed to determine the calendar date, e.g. “in December”;

Anchor date: the reference point which can be used to infer the normalized value of a relative or incomplete temporal expression;

Anchor relation: the nature of the temporal link between a relative or incomplete expression and their anchor date.

We kept the anchor relation restricted to the three possibilities: 'Before', 'Equal' or 'After'. The main difference between our annotation model and the one from Sun et al. (2015) is that we did not restrict the options for the anchor date, which could be any of the date and time timexes in the text.

We used the annotation tool that was developed for the i2b2 challenge (MAE). We generated annotation tags for RI-Timexes, and for the absolute timexes. Examples of relative and absolute time expressions as XML tags are shown in Figure 1. The annotators were given the following instructions:

a) for every absolute timex, decide whether it is truly an absolute timex or a RI-Timex that was mislabelled in the filtering, which is done by modifying the "absolute" attribute;

b) for every RI-Timex instance, decide whether it is truly a RI-Timex;

c) for every true RI-Timex, chose an anchor, i.e. another date – this is done by creating an ANCHORLINK, which is a link entity between a RI-Timex and another time expression, the anchor date; the anchorlink has a 'relation' attribute which the annotator needs to complete with either 'before', 'equal' or 'after'.

Three annotators worked on the annotations: two computer scientists, and researcher in life sciences. We divided the annotation process into three phases. Phase 1: we had three annotators, and each pair of annotators double-annotated a set of ten documents, for which inter-annotator agreement (IAA) was calculated, and we analysed disagreements to refine the guidelines. Phase 2: two annotators double-annotated a new set independently using the updated guidelines, after which IAA was again calculated. In the final phase, the remaining set was split in two and annotated separately.

2.3 Experimental setup

2.3.1 Baseline: Binary classification

We reproduced the methodology of Sun et al. (2015). To predict the anchor date for a given RI-Timex, four binary classifiers were trained, to discern if the RI-Timex is anchored to one of the four possible anchor dates. Similarly, for anchor relations, three binary classifiers were trained.

Sun et al. (2015) used SVM classifiers from the LibSVM implementation. These types of classifiers are especially adapted to text classification, as they can handle high dimensional inputs such as those created by one-hot encoded word vectors. We used the SVM algorithms of the sklearn library. As the hyperparameters were not specified in Sun et al. (2015), we performed hyperparameter optimization using 10-fold cross validation on the training set of both data sets. The optimized hyperparameters are included in the appendix.

The following set of features are used in Sun et al. (2015):

* The bag-of-word representation of a window of 8 tokens before and after the timex, as well as the timex itself. All numbers are normalized to a uniform token

* The bag-of-word of the previous timex

* The TimeML type of the previous timex (Date, Duration, Frequency, or Time)

We developed our binary-classification models with a mostly equivalent but slightly modified set of features:

* The numbers written in all letters were not normalized

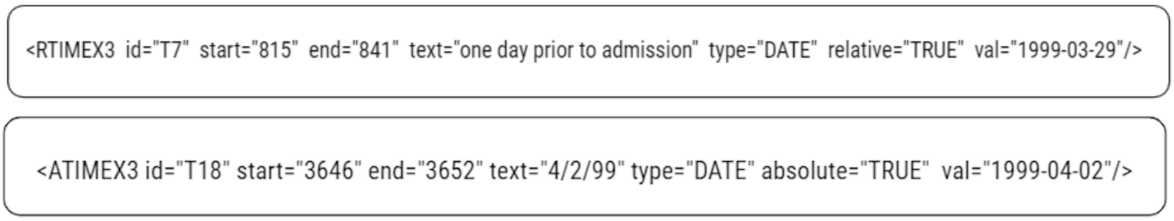
* We did not include the type of the previous timex, as we only considered Date and Time types

In the original paper, the expression "previous timex" was ambiguous as it was not clear whether or not it included the previous absolute time expression. We chose to use bag of word representation of both the previous timex and the previous absolute timex.

2.3.2 Relation classification approach

A common way to model the resolution of temporal relations in text is to classify pairs of temporal entities. We define our problem as a relation classification problem where, given two temporal expressions r and p, with r being a RI-Timex and p a potential anchor date, the task is to decide whether or not r is anchored to p, and the nature of the anchor relation: 'Before', 'Equal' or 'After'.

Model Choice: recent literature has shown some attempts to use neural models to classify temporal relations in text (Lin et al., 2019). We propose to use the BERT transformer model, to solve our anchor date relation problem. Transformer models such as BERT are trained on large corpora to generate a contextual language model, and can be fine-tuned to specific NLP tasks. This enables transfer learning and allows state-of-the-art performances



```
<RTIMEX3 id="T7" start="815" end="841" text="one day prior to admission" type="DATE" relative="TRUE" val="1999-03-29"/>
```

```
<ATIMEX3 id="T18" start="3646" end="3652" text="4/2/99" type="DATE" absolute="TRUE" val="1999-04-02"/>
```

Figure 1: Examples: RI-Timex and Absolute Time Expression annotations in XML tag format.

using relatively small task-specific data sets, without having to retrain the model from scratch. We use a version of BERT that was pre-trained on the whole MIMIC corpus (Alsentzer et al., 2019), making it especially adapted to the i2b2 dataset.

Input Definition: While not constrained within a sentence by previous models (Lin et al., 2019), BERT was not designed to solve problems of long-distance relations within a text. There is a limitation on the size of the input text sequences it can accept (512 tokens). As our problem might require longer text sequences, we defined an adapted input representation. For example, the method used by Lin et al. (2019) was to pass as input to BERT a single sequence of tokens with the relevant timexes tagged. For this method to work, both of the expressions from the candidate relation have to be part of a 512 token window in the text. We performed data analysis to quantify how many of our annotated relations were long-distance relations, and in particular, how many of the linked expressions were more than 512 tokens apart. We observed that the percentage of timexes where the number of tokens between the RI-Timex and its anchor date is greater that 512 is 33%.

We solved this problem by using the sequence pair classification feature of BERT: we transformed the inputs into a window of about 200 tokens around the two expressions. 7.6% of the anchor dates are located *after* the RI-TIMEX in the text, which means that the window of tokens had to cover both sides of each expression to be able to capture the relationship between them.

Data Augmentation: To create our candidate relation pairs, we generated all *(RI- timex, potential anchor)* pairs, where the potential anchors were all timexes from the Date or Time annotations. There are 17 786 examples of such pairs in total. 93% (16 638) pairs are not an anchor pair; 6.5% (1148) pairs are. To improve class representation, we used two techniques:

- Oversampling, which means augmenting the number of cases for the underrepresented class. There was a natural way of increasing the number of anchor dates, using the normalized value of the absolute timexes. In particular, for each RI-timex, we used every absolute timex that had the same normalized value as the original anchor date, as additional anchors. This method doubled the number of training examples that were actually anchor/timex pairs.

- Undersampling, which is the process of reducing the number of examples from the dominant class. We report results obtained when keeping only 50% of the training examples that were not an anchor/timex pair.

After oversampling and undersampling, we obtained 5316 training examples, out of which 1304 (24%) are anchor dates. 20% of the train set was used as a validation set to assess the performance of the model during training. We did not apply either oversampling nor undersampling on the blind test set. Table 3 reports the label distribution. We used the implementation of ClinicalBert from the huggingface transformers library (Alsentzer et al., 2019). We fine-tuned the model using an NVIDIA GPU. Technical details and hyper-parameters are reported in the appendix.

	Is ⚓	**B**	**E**	**A**	$\sum$
Training	1086	138	459	452	4222
Validation	255	36	111	108	1094
Test	501	76	199	226	8474

Table 3: Label distribution on the 2020 RI-Timex data for the BERT inputs

Output definition: Our goal is for the model to predict if the first expression is anchored to the second one and, if it is, what is the nature of the relation between them (Before, Equal or After). We cast this as a multi-label classification problem, where the model outputs a vector of four probabilities: the probability that the relation is an anchor ("Is ⚓") relation, and the probabilities that this relation is of the type Equal, Before or After. This way, if the model is sure that the first timex is anchored to the second, but unsure about the nature of their relation, it has the possibility to output different levels of probability for these two elements.

2.4 Evaluation approach and metrics

2.4.1 Inter-Annotator Agreement Evaluation

We evaluate the anchor date annotations as either strict or relaxed. The relaxed version takes into account that there are often several valid options as anchor dates: two links are considered equivalent if their anchor dates have the same normalized value.

2.4.2 Classification Evaluation

An important part of our work is to compare our results with those obtained by Sun et al. (2015). Direct comparison is impossible as we did not have access to their full annotated data. However, to the best of our knowledge, we did the maximum to reproduce their precise methodology. The authors only report results on 10-fold cross validation on the training set. Furthermore, they only report accuracy. We report precision, recall and f-score on both the 10-fold CV and the test sets, as well as accuracy on the test set.

3 Results

We report results for our annotation process: the inter-annotator agreements, and a comparison between our annotations and the annotations from Sun et al. (2015). We also report results for our classification experiments: the two attempts to reproduce Sun et al. (2015)'s methodology with distinct datasets and our anchor date predictions with a fine-tuned BERT model.

3.1 2020 RI-TIMEX i2b2 Annotation

Our annotation guidelines and resulting annotations are similar to the model used by Sun et al. (2015). The main difference is that we allow the anchor date to be *any* timex within the document, whereas they restrict the possibilities to four cases: one of the section times (Admission and Discharge date), the previous timex or the previous absolute timex. Results on the inter-annotator agreement on a subset of the corpus are presented in Table 4.

	Phase 1			**Phase 2**
	B1 P1	**B2 P2**	**B3 P3**	**B4 P1**
A: Strict	78	83	43	60
A: Relaxed	80	100	49	76

Table 4: Annotation agreement results on a subset 10 docs in each batch (B), and annotator pair (P) in two phases for guideline refinement. Adjudication was done by one of the annotators after consensus discussions with all annotators. A: Anchoring annotations of each identified RI-Timex.

Tables 5 and 6 show the resulting distributions of anchor date types and anchor relations, respectively. Note that as the anchor date categories are not mutually exclusive, the percentages do not add up to 100%. The 'Other' category for the anchor date types represents anchors that did not fall into the four categories used in Sun et al. (2015). They represent 7% of cases, thus indicating a substantial number of cases that were not naturally anchored to the four previously used categories.

	A	**D**	**PAT**	**PRT**	**O**	**N**	$\sum$
n	523	191	281	177	83	18	1273
%	45.0	16.4	24.2	15.2	7.1	1.5	-

Table 5: Distribution of annotation labels on the 2020 RI-Timex corpus: anchor date types. A: Admission; D: Discharge; PAT: Previous Absolute Timex; PRT: Previous Timex; O: Other; N: None

	Before	**Equal**	**After**	**None**	**Total**
n	161	476	512	18	1167
%	13.7	40.8	43.8	1.5	100

Table 6: Distribution of annotation labels on the 2020 RI-Timex corpus: anchor relations.

3.2 Classification

3.2.1 Baseline: Binary classification

We reproduced Sun et al. (2015)'s methodology on the 2015 RI-Timex subset and the 2020 RI-Timex data: bag-of-word representation of the time expression, the previous timex, and previous absolute timex with an SVM model.

Results on the 2015 RI-Timex subset and the 2020 RI-Timex data are presented in Table 7(a) and (b), respectively. For comparison, Table 8 shows the classification results reported by Sun et al. (2015) on the feature set that we used. Note that they only reported accuracy for this task.

	Anchor dates								Anchor relations					
Scores	A		D		PT		PAT		Before		Equal		After	
Phase →	CV	T	CV	T	CV	T	CV	T	CV	T	CV	T	CV	T
Precision	76.7	80.4	82.2	63.6	71.5	75.0	69.5	74.3	90.5	100	78.1	82.8	85.5	86.5
Recall	82.6	86.0	60.6	50.0	70.8	55.3	70.5	78.8	66.7	50	72.6	87.7	83.7	88.9
F-score	79.4	83.1	68.8	56.0	70.8	63.6	69.4	76.5	76.0	66.7	74.3	84.2	84.1	87.7
Accuracy	-	79.5	-	84.9	-	67.1	-	78.1	-	94.5	-	87.7	-	87.7

(a) 2015 RI-Timex data

Precision	77.7	81.1	75.3	81.6	93.8	84.2	82.7	77.2	87.9	86.0	82.4	74.7	84.3	84.7
Recall	73.7	79.2	64.8	44.4	80.0	82.5	76.6	83.6	63.6	56.5	78.4	80.4	84.1	78.4
F-score	74.5	80.1	68.2	57.5	86.2	83.3	78.7	80.3	72.3	69.2	80.0	77.4	84.0	81.4
Accuracy	-	80.4	-	88.4	-	90.0	-	85.3	-	92.1	-	81.8	-	84.1

(b) 2020 RI-Timex data

Table 7: Results on the 2015 RI-Timex data (a) and 2020 RI-Timex data (b), anchor dates and relations. A: Admission; D: Discharge; PT: Previous Timex; PAT: Previous Absolute Timex. Each row presents results for the two evaluation phases: 10-fold cross validation (CV) and test set (T).

A	D	PT	PAT	Before	Equal	After
77.56	92.47	68.91	75.16	93.4	81.4	92.1

Table 8: Accuracy for the 10-fold Cross validation on the training set reported by (Sun et al., 2015). A: Admission; D: Discharge; PT: Previous Timex; PAT: Previous Absolute Timex.

		Is ⚓	B	E	A	Avg
Valid.	P	85.2	85.0	86.5	83.0	85.5
	R	88.2	94.4	81.0	86.1	86.7
	F	86.7	89.4	83.7	84.5	85.8
Test	P	34.0	35.4	29.4	29.4	32.2
	R	76.2	60.5	57.2	72.1	70.3
	F	47.0	44.6	39.2	41.8	44.1

Table 9: Results of the anchor relation classification by BERT on the validation and test sets. Is ⚓: Is an anchor: B: Before; E: Equal; A: After. P: Precision; R: Recall; F: F-score.

3.2.2 Relation classification

Table 9 shows the results on the validation set after 15 epochs of fine-tuning the Clinical BERT model on our relation classification task, and on the final model applied on the test set. The results on the validation set range between 81-91% precision and 85-91% recall, yielding an average overall F-score of 87.6%. Results drop on the blind test sets, with an average of 70% recall and 32% precision. For comparison purposes, we also computed results on the test set where we performed oversampling. When the number of positive anchor relations went from 501 to 1086 (for 8474 total testing examples), the precision rose to 62-70%, and the average f-score is 67.2. Detailed results on this oversampled test set are presented in the appendix.

4 Discussion and Conclusion

We present a study on identifying and classifying anchor dates and relations specifically for RI-Timexes in clinical text. We attempted to reproduce the findings by Sun et al. (2015), in order to produce a baseline against which we could compare new approaches. Because the full dataset used in that study was not available, we developed new guidelines and produced a new reference standard of annotations with anchor date types and relations (2020 RI-TIMEX i2b2 data). We applied the methodology presented in Sun et al. (2015) on the 2015 RI-Timex subset, and our 2020 RI-Timex i2b2 data, reaching comparable results. We also propose to approach this problem as a relation classification task. To this end, we re-train the Clinical BERT model on our data. Results on the validation set were promising, but dropped on the test set.

4.1 Annotations and Corpus development

The additional guidelines that were defined after the first phase to solve ambiguous cases are presented in the appendix. One example concerns expressions relating to post-operation timelines – it was decided that all post-operative expressions should be anchored to the operation date, the exception to that rule being if there was another time expression which can serve as anchor for the post-operative expression with the relation 'EQUAL'.

An analysis of the annotation disagreements on the second round revealed that the main disagreements were due to longer, more complicated cases. For instance, in one case, there were two operation dates in the same document, in another there were two admission dates. Other examples included cases where events were unclear. *"Day of transfer"*, for example, could refer to a transfer between services or to the admission date.

Our annotation model allowed us to select more possibilities for anchor date types, while still being able to map our data to the 2015 RI-Timex subset. We observed that 7% of the RI-Timexes were anchored to dates located later in the text, indicating that the four categories proposed by Sun et al. (2015) had limitations.

Multi-anchor dates are still a challenge. For every RI-Timex, there are often more than one timex that can be an appropriate anchor date. This can be problematic if a machine learning model tries to mimic manual annotation labels. Not only does it need to learn what constitutes an acceptable anchor date, but also how to discriminate between potential alternative anchor dates. A solution for this problem could be to modify the guidelines and annotate as many anchor relations as possible.

4.2 Classification approaches

Sun et al. (2015) only report classification accuracy, and comparison with our results show that this distorts results to be more advantageous. For instance, the "Before" category and the "Discharge date" category are under-represented in the two datasets that we used. We can see that there is a sharp difference between the accuracy and f-score on these categories. In both cases, the accuracy is high but does not represent the actual performance of the model. It is notable that the results we have using our annotations are better than the one we obtained using their data subset. This is probably linked to the total number of samples. Another explanation would be that our annotations better capture the natural anchoring of RI-Timexes and are thus easier to predict for the model.

There could be several explanations to the difference in results between the validation set (around 90% f-score on average) and the test set (45%) on our BERT-based relation classification approach. One is that the model's hyperparameters were chosen to maximise results on the validation set, thus leading to a form of overfitting on the validation set, even though the model was never trained on this data. However, we repeated the experiment with the same hyperparameters and a different random validation set, and the results were similar. Furthermore, this would not explain the difference between the precision (about 30%) and recall (70%).

The likelier explanation lies in the oversampling process that we applied to the validation set but not to the test set. We have seen that to ensure inter-annotator agreement on which anchor date to pick, we had to define very precise, unambiguous guidelines which favored some solutions over others. These guidelines might be very difficult for the model to capture. Furthermore, the nature of the input means that the model only has access to the RI-Timex and the potential anchor date, but does not have any information about a competing anchor date that could have been preferred by a human annotator. By generating more instances of coherent anchor date/timex tuples, we decrease the complexity of the problem and allow it to generalise better. The fact that when we oversample the test set the results change dramatically supports this hypothesis. The results would probably improve even more if we could generate all anchoring relations accurately.

Another issue is the way input is processed in these types of transformer models. In our work, the model only had access to a small part of the context (200 token window of text on each side of the expression). To be able to reach the performance of a human, the model would need to be able to access and analyse the whole text, just as annotators did. One very interesting solution is the use of a context-aware neural network, as presented in Meng and Rumshisky (2018). The neural network reads the text linearly while using an external memory to store relations, and can then use the global context to classify them. Other alternatives could be to still leverage the power of pre-trained transformer models, by using solutions to pass entire, long texts to the model instead (Pappagari et al., 2019). A deep learning approach is potentially not the most appropriate to represent complex temporal relations, for example Li et al. (2020) recently reported good results using an ontology.

4.3 Conclusion

Our results on reproducing previous findings were promising. Our newly developed corpus results in comparable results using the same classification approach, but highlights limitations in the previous approach. Casting the problem as a relation classification task shows promise, but might require further considerations.

References

Emily Alsentzer, John R Murphy, Willie Boag, Wei-Hung Weng, Di Jin, Tristan Naumann, and Matthew McDermott. 2019. Publicly available clinical bert embeddings. *arXiv preprint arXiv:1904.03323*.

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

Leon R. A. Derczynski. 2017. *Automatically Ordering Events and Times in Text*. Studies in Computational Intelligence. Springer International Publishing.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Alistair EW Johnson, Tom J Pollard, Lu Shen, H Lehman Li-Wei, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. 2016. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3(1):1–9.

Artuur Leeuwenberg and Marie-Francine Moens. 2019. A survey on temporal reasoning for temporal information extraction from text. *Journal of Artificial Intelligence Research*, 66:341–380.

Fang Li, Jingcheng Du, Yongqun He, Hsing-Yi Song, Mohcine Madkour, Guozheng Rao, Yang Xiang, Yi Luo, Henry W Chen, Sijia Liu, et al. 2020. Time event ontology (teo): to support semantic representation and reasoning of complex temporal relations of clinical events. *Journal of the American Medical Informatics Association*, 27(7):1046–1056.

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 2nd Clinical Natural Language Processing Workshop*, pages 65–71.

Yuanliang Meng and Anna Rumshisky. 2018. Context-aware neural model for temporal information extraction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 527–536.

Raghavendra Pappagari, Piotr Zelasko, Jesús Villalba, Yishay Carmiel, and Najim Dehak. 2019. Hierarchical transformers for long document classification. In *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 838–844. IEEE.

James Pustejovsky, José M Castano, Robert Ingria, Roser Sauri, Robert J Gaizauskas, Andrea Setzer, Graham Katz, and Dragomir R Radev. 2003. Timeml: Robust specification of event and temporal expressions in text. *New directions in question answering*, 3:28–34.

James Pustejovsky, Kiyong Lee, Harry Bunt, and Laurent Romary. 2010. Iso-timeml: An international standard for semantic annotation. In *LREC*.

Jannik Strötgen and Michael Gertz. 2010. Heideltime: High quality rule-based extraction and normalization of temporal expressions. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, SemEval '10, page 321–324, USA. Association for Computational Linguistics.

Weiyi Sun, Anna Rumshisky, and Ozlem Uzuner. 2013a. Annotating temporal information in clinical narratives. *Journal of biomedical informatics*, 46:S5–S12.

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. *Journal of the American Medical Informatics Association*, 22(5):1001–1008.

Hegler Tissot, Marcos Didonet Del Fabro, Leon Derczynski, and Angus Roberts. 2019. Normalisation of imprecise temporal expressions extracted from text. *Knowledge and Information Systems*, 61(3):1361–1394.

Appendix

Annotation Instructions and Guidelines

The goal of this annotation task is to differentiate absolute from relative or incomplete time expressions, and to link the latter to anchor dates.

- Absolute time expression : an expression which contains all the information needed to normalize it to a standard date. Eg : "12/05/2020"
- Relative time expression : an expression whose temporal meanings is stated as a relative value against another time expression. Eg " two days before the admission"
- Incomplete time expression : an expression which holds only partial information : the context is needed to determine the calendar date. Eg : "in December"
- Anchor date : The reference point which can be used to infer the normalized value of a relative or incomplete temporal expression.

Instructions

1. Load the data in the annotation tool

   Once loaded, the following tabs should appear :

   - - RTIMEX3 : these are the relative time expressions, the main focus of the annotation process
   - - ATIMEX3 : the absolute time expressions, here to provide anchorage for the RTIMEX3
   - - SECTIME : These are special annotations for the admission and discharge date
   - - ANCHORLINK : These links will be created by the annotator to join a RTIMEX3 to an anchor date (an ATIMEX3 or SECTIME).

2. The text should appear along with the annotations outlined in different colors. The RTIMEX3 are in blue, the ATIMEX3 in red, the discharge and admission date are usually double annotated as both absolute time expressions and SECTIME so they are underlinded. The ANCHORLINKs are empty as they will be created during the annotation process 6. The process is as follows : The focus should be on each RTIMEX3 annotations until they are all annotated

   - First, the "relative" column must be filled : with "TRUE" if the expression is indeed a relative time expression, "FALSE" if it is an absolute timex3 which was not correctly filtered. Common example would be : "May 1997", "On Christmas of 2002", "April 2nd 2015"
   - If "relative" is TRUE, an ANCHORLINK has to be created This is done by holding down the ctrl key (or the command key, if you are on a Mac) and left click each of the entities that will be included in the link, with the RTIMEX3 first and the Anchor Date second. For precise instructions on how to select the appropriate anchor date, see the "Guidelines" section. A link window will pop up and ask you to confirm the two dates and the link type. Special case - if the anchor date is the admission or discharge date : because these are double annotated as ATIMEX3 and SECTIME, the program will let you choose between the two instances. They have the same value so it does not matter too much but the SECTIME should be preferred.
   - Once the ANCHORLINK is created, the "relation" attribute has to be filled with either BEFORE, EQUAL or AFTER

3. Check the ATIMEXEs : sometimes, an expression marked as an absolute time expression is in fact a relative one. For this, the "absolute" attribute of the A-TIMEXes as to be filled with True or False. If the expression is in fact a relative one, it has to be anchored.

4. Output the file Once all the RI-TIMEXES are filtered and anchored, choose the "Export as XML" option in the File menu, and save the file with its original name in a separate folder.

5. Upload the data

Guidelines for Ambiguous Cases :

- SELECTING THE ANCHOR DATE When selecting the anchor date, the first potential anchor dates to study are : the previous absolute timex, the previous timex, the admission date and the discharge date. One should prioritize absolute anchor dates over relative ones, and if there is still an ambiguity, “EQUAL” relations over “BEFORE” and “AFTER”. These four possibilities are to be prioritised, but other anchor dates are valid as well.
- POSTOPERATIVE DAYS As a general rule, expressions relating to the “post-operation” concept should be anchored to the day of the operation. The exception to that rule is if there is another time expression which can serve as anchor for the POD with the relation “EQUAL”
- AGE RELATED EXPRESSIONS : Some time expressions annotated as dates age in fact age expressions. If this case arises, one has to change the type of the expression to “AGE_RELATED”.
- INCOMPLETE EXPRESSIONS Some expressions are not relative but rather incomplete: their normalized value depends on one or more missing information, such as the year. ex “Labor Day” In this case, they should still be annotated as RI-TIMEXs, and if they cannot be anchored, it is possible to change the “mod” attribute of the expressions to “EXT”, to signify that there is a need for external information.
- NON-ANNOTATED EXPRESSIONS Sometimes, expression which should be annotated as either RI-TIMEXs or A-TIMEX are not annotated at all : this is likely an error coming from i2b2’s gold standard, and we should let them as is. No annotation should be added to the documents.
- INCOMPLETE TIMES Eg “2.30 pm” They are to be anchored to the day they belong to. Usually they are wrongly annotated as absolute time expressions.
- SECTION TIMES Usually, imprecise expressions found at the beginning of a document relate to the Admission date, and those found at the end to the Discharge date.

SVM Classification : Optimized Hyperparameters

Here we report the optimized hyperparameters for each binary classification category:

- Admission Date : ’C’: 1000, ’gamma’: 0.0001, ’kernel’: ’rbf’
- Discharge Date : ’C’: 100, ’gamma’: 0.001, ’kernel’: ’rbf’
- Previous Timex : ’C’: 100, ’gamma’: 0.001, ’kernel’: ’rbf’
- Previous Absolute Timex = ’C’: 100, ’gamma’: 0.001, ’kernel’: ’rbf’
- Before = ’C’: 1000, ’gamma’: 0.0001, ’kernel’: ’rbf’
- Equal = ’C’: 100, ’gamma’: 0.001, ’kernel’: ’rbf’
- After = ’C’: 1000, ’gamma’: 0.0001, ’kernel’: ’rbf’

BERT Relation Classification : Model Specification

The BERT model was trained for 15 epochs on a NVIDIA GPU with the following characteristics :

NVIDIA-SMI 415.18 — Driver Version: 415.18 — CUDA Version: 10.0

The hyperparameters were :
Length of input : 512

Learning rate : 2e-5
Number of training epochs : 15
Gradient accumulation steps : 0.9
Batch size : 5
fp16 : False

Additional Results

Detailed results and distribution of BERT Relation classification

		Is ⚓	**B**	**E**	**A**	**Avg**
Valid.	P	85.2	85.0	86.5	83.0	85.5
	R	88.2	94.4	81.0	86.1	86.7
	F	86.7	89.4	83.7	84.5	85.8
Test	P	34.0	35.4	29.4	29.4	32.2
	R	76.2	60.5	57.2	72.1	70.3
	F	47.0	44.6	39.2	41.8	44.1
O. Test	P	70.6	70.0	62.1	62.3	67.0
	R	73.0	49.1	56.8	71.5	67.5
	F	71.7	57.7	59.4	66.6	67.2

Table 10: Results of the anchor relation classification by BERT on the validation and test sets.O.Test : Oversampled Test Set; Is ⚓: Is an anchor: B: Before; E: Equal; A: After. P: Precision; R: Recall; F: F-score.

	Is ⚓	**B**	**E**	**A**	**Total**
Train.	1049	138	459	452	4222
Valid.	255	36	111	108	1094
Test	501	76	199	226	8474
O. Test	1086	185	419	482	8474

Table 11: Distributions of examples in the anchor relation classification dataset .O.Test : Oversampled Test Set; Is ⚓: Is an anchor: B: Before; E: Equal; A: After. P: Precision; R: Recall; F: F-score.

Package Versions

Package versions on the local system

Python version : 3.7.6
gensim==3.8.1
h5py==2.10.0
matplotlib==3.2.1
nltk==3.4.5
numpy==1.18.1
pandas==1.0.3
scikit-learn==0.22.2.post1
scipy==1.4.1
sklearn==0.0
spacy==2.2.3
torch==1.5.0+cpu
torchvision==0.6.0+cpu
tqdm==4.43.0
transformers==2.11.0

Package Versions on the GPU server (BERT model training)

Python version : 3.7.6

```
torch==1.5.1+cu92
torchvision==0.6.1+cu92
```