Inducing Temporal Relations from Time Anchor Annotation

Fei Cheng and **Yusuke Miyao**
Research Center for Financial Smart Data
National Institute of Informatics
2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo, Japan
{fei-cheng, yusuke}@nii.ac.jp

Abstract

Recognizing temporal relations among events and time expressions has been an essential but challenging task in natural language processing. Conventional annotation of judging temporal relations puts a heavy load on annotators. In reality, the existing annotated corpora include annotations on only ”salient” event pairs, or on pairs in a fixed window of sentences. In this paper, we propose a new approach to obtain temporal relations from absolute time value (a.k.a. *time anchors*), which is suitable for texts containing rich temporal information such as news articles. We start from time anchors for events and time expressions, and temporal relation annotations are induced automatically by computing relative order of two time anchors. This proposal shows several advantages over the current methods for temporal relation annotation: it requires less annotation effort, can induce inter-sentence relations easily, and increases informativeness of temporal relations. We compare the empirical statistics and automatic recognition results with our data against a previous temporal relation corpus. We also reveal that our data contributes to a significant improvement of the downstream time anchor prediction task, demonstrating 14.1 point increase in overall accuracy.

1 Introduction

Temporal information extraction is becoming an active research field in natural language processing (NLP) due to the rapidly growing need for NLP applications such as timeline generation and question answering (Llorens et al., 2015; Meng et al., 2017). It has great potential to create many practical applications. For example, SemEval-2015 Task 4 (Minard et al., 2015) collects news articles about a target entity and the task required participants automatically ordering the events involving that entity in a timeline. The timeline representation of news can help people more easily comprehend a mass of information. This work aims to contribute to such timeline applications by extracting temporal information in specific domains like news articles.

TimeBank[1] (Pustejovsky et al., 2003) is the first widely used corpus with temporal information annotated in the NLP community. It contains 183 news articles that have been annotated with events, time expressions and temporal relations between events and time expressions. The annotation follows the TimeML[2] specification (Saurı et al., 2006). Along with the TimeBank and other temporal information corpora, a series of competitions on temporal information extraction (TempEval-1,2,3) (Verhagen et al., 2009, 2010; UzZaman et al., 2012) are attracting growing research efforts.

A majority of temporal information corpora adopt temporal links (TLINKs) to encode temporal information in documents. A TLINK denotes a temporal relation between mentions, i.e., events, time expressions and document creation time (DCT) (Setzer, 2002). However, annotating TLINKs is a painful work, because annotation candidates are quadratic to the number of mentions in a document. The original TimeBank only annotated those “salient” mention pairs judged by annotators, while the definition of “salient” is not necessarily clear. Annotators had to face a complicated task; identify “salient” mention pairs, and label temporal relations. For solving this, many dense annotation schemata are proposed to force annotators to annotate more or even complete graph pairs. However, dense annotation is time-consuming and unstable human judgments

[1] https://catalog.ldc.upenn.edu/LDC2006T08
[2] http://www.timeml.org/

Proceedings of NAACL-HLT 2018, pages 1833–1843
New Orleans, Louisiana, June 1 - 6, 2018. ©2018 Association for Computational Linguistics

on "salient" pairs are not improved at all. As a consequence, a high proportion of "vague" or "no-link" pairs appears in these dense corpora such as TimeBank-Dense (Cassidy et al., 2014).

In this work, we propose a new approach to obtain temporal relations from time anchors, i.e. absolute time value, of all mentions. We assume that a temporal relation can be induced by comparing the relative temporal order of two time anchors (e.g. *YYYY-MM-DD*) in a time axis. We use pre-defined rules (Section 3) to generate temporal order (TORDER) relations (e.g. BEFORE, AFTER, SAME_DAY, etc.) by taking two annotated time anchors as input. This proposal requires the annotation of time anchors, of which the annotation effort is linear with the number of mentions. This is the first work to obtain temporal relations shifted from the annotation of individual mentions, which is distinguished from most annotation work of manually annotating mention pairs.

This approach brings several advantages over the current temporal relation annotation. First, as long as time anchors of all mentions in a document are given, our pre-defined rules can induce the temporal relations for all the quadratic pairs. This skips the step of identifying "salient" pairs. Second, annotating the time anchors is relatively easy, as the annotation work is linear to the number of mentions. Third, the automatic generation rules can provide flexible relation types based on our definition and this increased informativeness might contribute positively to downstream tasks.

In our first evaluation (Section 4), we compare the correspondence and difference between the new TORDERs and conventional TLINKs. The comparison of empirical statistics shows the new data is label balanced, contains informative relations and reduces "vague" relations. Besides, the classification performance suggests the new data achieve reasonable accuracy, although accuracy numbers are not directly comparable.

Many text processing tasks are often requiring to know time anchors when events occurred in a timeline. In Section 5, we evaluate the data in a downstream time anchor prediction task (Reimers et al., 2016) by using the temporal relation recognizers separately trained with TORDERs or TLINKs. The main results show that the recognizer trained with our TORDERs significantly outperforms the recognizer trained with the TLINKs by 14.1 point *exact match* accuracy.

2 Background

2.1 Temporal Relation Annotation

TimeBank started a wave of data-driven temporal information extraction research in the NLP community. The original TimeBank only annotated relations judged to be *salient* by annotators and resulted in sparse annotations. Subsequent TempEval-1,2,3 competitions (Verhagen et al., 2009, 2010; UzZaman et al., 2012) mostly relied on TimeBank, but also aimed to improve coverage by annotating relations between all events and time expressions *in the same sentence*. However, most missing relations between mentions in different sentences are not considered.

In order to solve the sparsity issue, researchers started the work towards denser annotation schema. Bramsen et al. (2006) annotated multi-sentence segments of text to build directed acyclic graphs. Kolomiyets et al. (2012) annotated temporal dependency structures, though they only focused on relations between pairs of events. Do et al. (2012) produced the densest annotation and the annotator was required to annotate pairs "as many as possible". Cassidy et al. (2014) proposed a compulsory mechanism to force annotators to label every pair in a given sentence window. They performed the annotation (TimeBank-Dense) on a subset (36 documents) of TimeBank, which achieved a denser corpus with 6.3 TLINKs per event and time expression, comparing to 0.7 in the original TimeBank corpus. However, it raises the issue that hand-labeling all dense TLINKs is extremely time-consuming and the unclear definition of "salient" is not improved at all.

2.2 Temporal Relation Classification

The majority of the temporal relation classifiers focus on exploiting a variety of features to improve the performance in TimeBank. Laokulrat et al. (2013) extracted lexical and morphological features derived from WordNet synsets. Mani et al. (2006); D'Souza and Ng (2013) incorporated semantic relations between verbs from VerbOcean.

Recently, more researchers move on to diverse approaches on the TimeBank-Dense corpus. Chambers et al. (2014) proposed a multi-sieve classifier composed of several rule-based and machine learning based sieves ranked by their precision. Mirza and Tonelli (2016) started to mine the value of low-dimensional word embeddings by concatenating them with traditional sparse feature vectors to improve their classifier.

Inspired by the success of the deep learning work in the similar task: relation extraction, Cheng and Miyao (2017) proposed the shortest dependency path based Bi-directional Long short-term memory (Hochreiter and Schmidhuber, 1997) (Bi-LSTM) to achieve state-of-the-art performance in the TimeBank-Dense corpus, which is adopted for the experiments in this paper. There are two reasons to use this classifier: 1) inter-sentence temporal relations are well treated. 2) only word, part-of-speech and dependency relation embeddings are required as input.

2.3 Time Anchor Annotation

A related task: Cross-Document Event Ordering (Minard et al., 2015) aims to order the events involving a target entity in a timeline given written news in English. Compared to traditional TLINKs, annotating time anchors of events is intuitively more straightforward in such tasks.

Reimers et al. (2016) proposed an annotation scheme, which requires annotators to infer the exact time of each individual event. They distinguished events that occur at a *Single-Day* from that span over *Multi-Day* by setting the granularity as one day. For *Single-Day* events, the event time is written in the format '*YYYY-MM-DD*' when the precise event time can be determined. Otherwise, they required annotators to narrow down the possible time as precisely as possible. An imprecise *Single-Day* event can be annotated as a tuple *(after, before)*, e.g. '*(after 1998-10-02,)*', '*(, before 2000-01-31)*' or '*(after 1998-10-02, before 2000-01-31)*'. In the case of *Multi-Day*, an event is annotated as a tuple *(begin, end)*, where *begin* and *end* are represented with *Single-Day*. For instance of a sentence:

> *The economy* ***created*** *jobs at a surprisingly robust pace in January, the government* ***reported*** *on Friday, evidence that America's economic stamina has withstood any* ***disruption*** *caused so far by the financial tumult in Asia.*

The *Multi-Day* event ***created*** is annotated as *(begin=1998-01-01, end=1998-01-31)*. The *Single-Day* event ***reported*** is annotated as the same day as DCT *(1998-02-06)*. The imprecise *Multi-Day* event ***disruption*** is annotated as *(begin=(, before1998-02-06), end=(, before1998-02-06))* as the event must have occurred before the time of this news, but the precise *begin* and *end* dates cannot be inferred from the text. Time anchors have the capability of anchoring all the events from a document into the same timeline as shown in Figure 1. They annotated the time anchors of total 1,498 events from 36 documents of TimeBank-Dense.

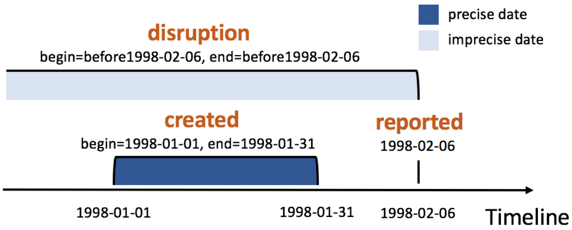


Figure 1: Anchoring events in a timeline

In temporal information retrieval, Berberich et al. (2010) proposed a four-tuple representation *('earliest begin', 'latest begin', 'earliest end', 'latest end')* for uncertain time expression (e.g. '1990s') in order to integrate such temporal information into language model. In the time anchor annotation, an event 'in 1990s' will be annotated as a *Multi-Day* event with imprecise *begin* and *end* points, i.e. *(begin=(after 1990-01-01, before1999-12-31), end=(after 1990-01-01, before1999-12-31))*, which is quite similar to their four-tuple representation.

3 Automatic generation of TORDERs

TimeML states that TLINKs present a temporal relation between event to event, event to time expression, and event to DCT. The sparse TLINK coverage in the majority of temporal information corpora is attributed to the unstable identification of "salient" pairs by human annotators. Denser annotation schemata somehow improved sparseness, but the annotation work became very time-consuming. These issues plague the development of temporal information extraction work.

Our temporal order (TORDER) proposal is designed with the goal of solving unstable recognition of "salient" pairs and reducing annotation effort. We hypothesize that a temporal relation can be automatically computed by comparing the relative temporal order between two time anchors (e.g. *YYYY-MM-DD*) in a time axis. We propose a set of pre-defined generation rules, which have the capability to rigorously induce a TORDER by taking the two annotated time anchors as input. Annotat-

TORDER	Condition
Two precise S_1 and S_2	
BEFORE	if $S_1 < S_2$
AFTER	if $S_1 > S_2$
SAME_DAY	if $S_1 = S_2$
A precise S_1 and an imprecise S_2 $(\mathit{after}_2, \mathit{before}_2)$	
BEFORE	if $S_1 \leq \mathit{after}_2$
AFTER	if $S_1 \geq \mathit{before}_2$
VAGUE	other cases
Two imprecise S_1 $(\mathit{after}_1, \mathit{before}_1)$ and S_2 $(\mathit{after}_2, \mathit{before}_2)$	
BEFORE	if $\mathit{before}_1 \leq \mathit{after}_2$
AFTER	if $\mathit{after}_1 \geq \mathit{before}_2$
PVAGUE	if $\mathit{before}_1 = \mathit{before}_2$ and $\mathit{after}_1 = \mathit{after}_2$
VAGUE	other cases

Table 1: Definition of the temporal orders between two *Single-Day* events. '<', '>', '=' denote one event is in the left of, right of and same position as the other event in a left-to-right time axis.

TORDER	Condition
A *Single-Day* S_1 and a *Multi-Day* M_2 $(\mathit{begin}_2, \mathit{end}_2)$	
BEFORE	if S_1 **BEFORE** begin_2
AFTER	if S_1 **AFTER** end_2
IS_INCLUDED	if S_1 **AFTER/SAME_DAY** begin_2 and S_1 **BEFORE/SAME_DAY** end_2
VAGUE	other case
Two *Multi-Day* M_1 $(\mathit{begin}_1, \mathit{end}_1)$ and M_2 $(\mathit{begin}_2, \mathit{end}_2)$	
BEFORE	if end_1 **BEFORE** begin_2
AFTER	if begin_1 **AFTER** end_2
SAME_SPAN	if begin_1 **SAME_DAY** begin_2 and end_1 **SAME_DAY** end_2
IS_INCLUDED	if begin_1 **AFTER/SAME_DAY** begin_2 and end_1 **BEFORE/SAME_DAY** end_2 (*)
INCLUDES	if begin_1 **BEFORE/SAME_DAY** begin_2 and end_1 **AFTER/SAME_DAY** end_2 (*)
PVAGUE	if begin_1 **PVAGUE/SAME_DAY** begin_2 and end_1 **PVAGUE/SAME_DAY** end_2 (*)
VAGUE	other cases

Table 2: Definition of the temporal orders involving *Multi-Day* events M $(begin, end)$. '*' denotes excluding the **SAME_SPAN** case in the current condition.

ing time anchors of individual mentions extremely reduces annotation effort, as it is linear with mention numbers. As long as time anchors are given, our pre-defined rules can induce the temporal relations for all the quadratic pairs, which skips the step of identifying "salient" pairs.

TimeBank contains the normalized date '*YYYY-MM-DD*' of time expressions and DCT, but does not include events' time. Our proposal of inducing a TORDER by comparing two time anchors requires the time anchor annotation of events in the same granularity as time expressions and DCT. Therefore, annotating the events with '*YYYY-MM-DD*' is a reasonable setting and one day is used as the minimal granularity of annotation. We choose the annotation (Reimers et al., 2016) of the day-level time anchors of events as the source of our automatic TORDER generator. In the case that a corpus can provide more specific time information '*YYYY-MM-DD, hh-mm-ss*' (e.g. this morning, three o'clock in the afternoon), our TORDER generator can be flexible to handle this information as long as the time anchors of all mentions are annotated in the same granularity.

For the clear demonstration of the definition of the auto-generated temporal order, we separately describe the generation of the pairs with two *Single-Day* mentions, and the pairs involving *Multi-Day* mentions. In this paper, TORDER labels are written in the upper-case bold font to be distinguished from TLINK labels written in the lower-case italic font. Table 1 introduces the definition of temporal orders between two *Single-Day* pairs S_1 and S_2. **PVAGUE** (i.e. partially vague) denotes that two imprecise time anchors are equivalent. For instance, we cannot induce a clear temporal relation between two events both occurring on *(,before1998-02-06)*, but nevertheless both events provide partially equivalent date information *'1998-02-06'*. It can possibly provide useful information for the future processes of classification or time inference. **PVAGUE** in the *Multi-Day* definition takes the same consideration.

In order to introduce the temporal orders involving *Multi-Day* events, a *Multi-Day* event M is denoted as a tuple of two *Single-Day* dates $(begin, end)$. A temporal order between a *Single-Day* S_1 and *Multi-Day* M_2 $(begin_2, end_2)$ can be derived by computing the temporal order of two *Single-Day* S_1 and $begin_2$, or S_1 and end_2 first. All the types of temporal orders involving *Multi-Day* events are defined in Table 2. One additional INCLUDES relation that *Multi-Day* event includes a *Single-Day* event can be obtained by reversing the symmetric IS_INCLUDED.

The example of automatically computing temporal orders can be demonstrated by using the events in Figure 1. Both *Multi-Day* ***created*** and ***disruption*** are clearly BEFORE the *Single-Day* ***reported***, because ***reported*** is AFTER the *end* dates of ***created*** and ***disruption***. The relation between ***created*** and ***disruption*** is induced as VAGUE, as the imprecise *begin*, *end* of ***disruption*** cannot be determined with a relation to ***created***.

In this paper, the definition adopts a similar relation set to TLINK for the purpose that we can perform fair comparison and evaluation in the next two sections. However, our inducing proposal can be very scalable to introduce more temporal relations. For instance, Allen's interval algebra (Allen, 1990) defines 'starts', 'finish' relations, which are not included in our current definition. We can easily extend our definition by detecting whether two time anchors have the equivalent *begin* or *end* points.

Our inducing proposal takes human annotated time expressions and normalized values as inputs to generate TORDER relations as the training data of the next processes (e.g. classification). In the case of processing raw texts, we can perform detection and normalization of time expressions by using existing temporal taggers, e.g. HeidelTime (Strötgen and Gertz, 2015), SUTime (Chang and Manning, 2012), etc.

4 Comparison of TORDERs and TLINKs

Fairly evaluating the TORDER's capability of encoding temporal order information compared to the existing data is difficult but necessary work. This section provides empirical statistics of TORDER and TLINK annotations, and compare the performance of automatic recognition. Additionally, we evaluate these two frameworks in a downstream task performance in Section 5.

4.1 Correspondences and Differences

Our new TORDERs are formally similar to the conventional TLINKs, as both state a temporal relation between two mentions. BEFORE and AFTER represent that one mention occurs before or after in a timeline, which is close to *before* and *after*. INCLUDES and IS_INCLUDED are more clearly conditioned as a *Single-Day* or *Multi-Day* mention occurs during the other *Multi-Day* mention, compared to *includes* and *is_included*. SAME_DAY and SAME_SPAN are designed for the one-day minimal granularity. Ideally, these two relations will include *simultaneous* and other TLINKs with two mentions occurring in the same day. VAGUE and PVAGUE state that our generation rules cannot induce the relations, similar to *vague* (i.e. annotators cannot judge the relations).

The one-day minimal granularity is the main reason causing the difference between TORDER and TLINK types. For a sentence:

> *I went to* ***sleep*** *after taking a* ***bath****.*

According to the TimeML specification, ***sleep*** is obviously *after* ***bath***. But in the one-day granularity, the relation is shifted to SAME_DAY. This brings the obstacle that we cannot measure whether the temporal information encoded in TORDERs is more informative than TLINKs by directly comparing the classification results.

Our TORDER definition shows the capability of capturing some relations which cannot be encoded by TLINK. For instance:

> Stocks ***rose***, ***pushing*** the Dow Jones industrial average up 72.24 points, to 8,189.49, ***leaving*** the index within 70 points of its record.

These TLINKs among the three events are annotated as *vague* in TimeBank-Dense, as the annotators cannot state their temporal orders. However, we can easily obtain SAME_DAY relations, since their day-level time anchors are the same.

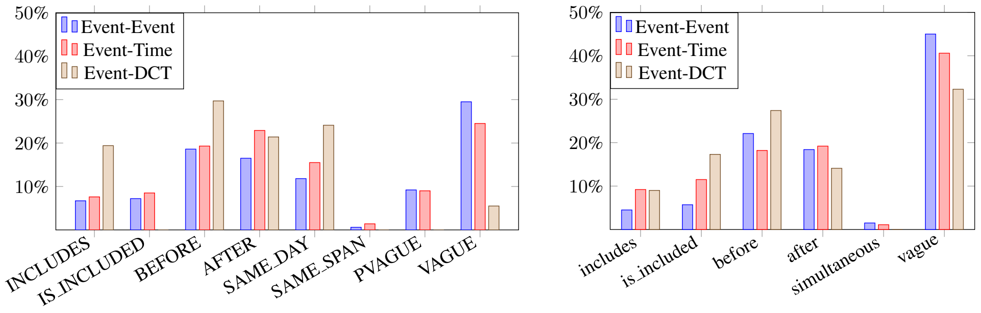
Imprecisely represented time anchors (e.g. *after YYYY-MM-DD*) are the major drawback of losing temporal order information. For instance:

> America's economic stamina has **withstood** any **disruption**...

The TLINK between ***withstood*** and ***disruption*** is annotated as *after*. While both of them were annotated as the same time anchor *(begin=before 1998-02-06, end= before 1998-02-06)*, our TORDER generator induced a PVAGUE relation and temporal order information is lost.

The hypothesis that our proposal skipping the unstable manual identification of "salient" pairs can reduce the VAGUE relations in the new data. This can be measured by comparing the numbers of the TORDER and TLINK relations on the same mention pairs. If the observation of a part of *vague* TLINKs induced as non-VAGUE TORDERs in the new data can be found, it will be the evidence.

Depending on the text domain, TLINKs or TORDERs can be advantageous in different scenarios. TLINKs can capture the temporal ordering information between events, when time expressions are often absent in the documents such as novels and narratives. But the annotation work is time consuming and a part of relations will be neglected by the unstable human identification of "salient" pairs. TORDERs have the capability of capturing more informative relations by skipping the "salient" pairs recognition and need less annotation effort. But they require that the events can be anchored in a timeline from a document (e.g. often the case of news articles) and imprecise time anchors cause some information loss.



(a) TORDER relations

(b) TLINK relations

Figure 2: The label distribution of the TORDER and TLINK relations

	b	***a***	***s***	***i***	***ii***	***v***
BEFORE	1486	24	0	22	26	542
AFTER	19	1242	5	26	66	503
SAME_DAY	155	93	83	164	343	647
SAME_SPAN	4	0	9	5	6	42
INCLUDES	104	61	2	225	25	372
IS_INCLUDED	56	71	1	25	214	333
PVAGUE	91	40	41	23	36	336
VAGUE	331	261	33	145	136	1464

Table 3: The comparison of the numbers of TORDER and TLINK annotations for the same mention pairs. ***b***:*before*, ***a***:*after*, ***s***:*simultaneous*, ***i***:*includes*, ***ii***:*is_included*, ***v***:*vague*.

4.2 Empirical Comparison

Investigating the quality of auto-generated TORDERs is important to demonstrate the value of this research. In this section, we empirically compare the statistics of the auto-generated TORDERs and human-annotated TLINKs. Theoretically, a TORDER between two mentions with any distance in a document can be automatically computed. However, it is important to make the new data in a comparable manner to the existing data. In this paper, we follow the process of TimeBank-Dense (Cassidy et al., 2014) to generate the complete graph of the 10,007 mention pairs in the same and adjacent sentences. The TORDER data used in this paper are publicly available[3] and our scalable generation method can be easily applied for inducing relations of longer distance pairs.

Table 3 shows the comparison between the numbers of the TimeBank-Dense TLINKs and the new TORDERs. One observation as we expected is that our approach captures new relations for a considerable part of the mention pairs that were judged as ***v*** (*vague*) in the human-annotated TLINKs. 542 *vague* relations are induced as AFTER in the new TORDERs, as well as other relation types. However, a part of non-*vague* TLINKs are shifted to VAGUE TORDERs. This matches our description of the imprecise time anchor issue. It is a trade-off between the part of mention pairs obtaining richer temporal information and the part of pairs losing information. That is the reason why we need a downstream task (i.e. Time Anchors Prediction in Section 5) to measure how much temporal order information is encoded in TORDERs and TLINKs. The shift of TLINK relations to SAME_DAY due to the one-day minimal granularity setting can also be clearly observed.

Figure 2 shows the label distributions of the auto-generated TORDERs and the TimeBank-Dense TLINKs. We investigate the statistics of Event-Event, Event-Time, and Event-DCT pairs. The TimeBank-Dense corpus is obviously sparser due to the high proportion of *vague* in all three types of pairs. Our TORDERs show a more balanced distribution of labels, which suggests that this method possibly encodes more informative temporal orders compared to the traditional TLINKs. In particular, TORDERs show extremely rare VAGUE labels in Event-DCT pairs. When given the precise *Single-Day* DCT of a document, our proposal to compare the temporal order between the time anchor of a event and the DCT manages to avoid the most unstable judgments made by the human annotators in the Event-DCT pairs. Although the different definition of TORDERs from TLINKs makes direct comparison difficult, the more balanced distribution of TORDERs can possibly provide more informative classification results to benefit the downstream tasks.

[3] https://github.com/racerandom/temporalorder

	Event-DCT		Event-Time		Event-Event	
	F1	N	F1	N	F1	N
AFTER	0.585	65	0.509	67	0.426	184
BEFORE	0.659	65	0.452	68	0.488	257
INCLUDES	0.400	38	0.136	27	0.158	105
IS_INCLUDED	0	0	0	20	0.077	86
SAME_DAY	0.631	82	0.485	56	0.314	131
SAME_SPAN	0	0	0	2	0	0
PVAGUE	0	0	0	1	0.149	92
VAGUE	0	18	0.417	119	0.487	335
Overall	0.557	268	0.403	360	0.374	1190
Non-VAGUE	0.597	250	0.390	240	0.351	763

(a) TORDER

	Event-DCT		Event-Time		Event-Event	
	F1	N	F1	N	F1	N
after	0.582	68	0.550	64	0.443	223
before	0.612	58	0.331	91	0.465	326
includes	0.170	22	0.290	31	0.126	45
is_included	0.559	48	0.338	42	0.099	47
simultaneous	0	0	0	6	0	19
vague	0.433	72	0.557	126	0.6116	530
Overall	0.511	268	0.441	360	0.492	1190
Non-*vague*	0.539	196	0.378	234	0.395	660

(b) TLINK

Table 4: The classification results of Event-Event, Event-DCT and Event-Time F1-measure on individual relation types and weighted overall F1. '**N**' denotes the number of the relations in the test split.

4.3 Classification Results

Although the classification results of TORDERs and TLINKs are not directly comparable, they can show some evidence whether TORDERs is functional to provide temporal order information. Table 4 shows the Bi-LSTM classification results with the data split[4](Chambers et al., 2014) (27 training/validation documents, 9 testing documents).

The classification system achieves fairly high F1 0.631 in Event-DCT and 0.485 in Event-Time on the SAME_DAY temporal orders, which are the main information source to predict the precise time of events. The performance on AFTER, BEFORE temporal orders are close to the TLINKs in number, but not meaningfully comparable. The high proportion of *vague* in the TLINKs results in biased predictions. When we use a more meaningful evaluation 'Non-*vague*' overall, the TLINKs performance drops sharply. Generally, the classification results suggest that our proposal of auto-generated TORDERs has sufficient capability to encode temporal information, which can be well classified from the textual inputs.

5 Evaluation in Time Anchor Prediction

In this section, we describe a two-step system trained with the existing TLINKs and our data to challenge a downstream time anchor prediction task. The different performance can be seen as the evidence whether our auto-generated TORDERs can capture comparable temporal information to the human-annotated TLINKs.

5.1 Task Definition

Predicting the time of events from the news articles is an attractive goal, which is a necessary step towards automatic event timeline extraction. Reimers et al. (2016) bring the task of time anchor prediction, which aims to predict the time anchor of each *Single-Day* event given a document. They use a general two-step process to determine the event anchors as shown in Figure 3. Given a set of documents with events and time expressions already annotated, the system first obtains a list of possible times for each event. Then, the most precise time is selected for each event.

A serious issue is that their baseline system still depends on the TimeBank-Dense TLINK classifier and the time anchor annotation is only used for the final evaluation. That leaves the space to consider a new method without relying on the human-annotated TLINKs. Our auto-generated TORDERs are a natural alternative to TLINKs to provide the similar temporal order information of mention pairs, but with less annotation efforts. The second-step selection rules just need a slight modification to replace the previous TLINK types with the new TORDER types.

5.2 The Two-step System in Experiments

In this work, we adopt a similar two-step architecture. The first-step temporal order classifier is designed to provide the temporal relations of the mention pairs in a document.

The second-step selects the most precise time by taking all Event-Time and Event-DCT relations of a target event as input. For instance in Figure 3, the second-step received a set of relations e.g. $(is\_included, DCT)$, $(is\_included, Friday)$ and $(vague, January)$ of ***reported***. For the system trained with the TimeBank-Dense TLINKs, we adopt the same selection algorithm as described in (Reimers et al., 2016). When the system is trained

[4] https://github.com/nchambers/caevo/blob/master/src/main/java/caevo/Evaluate.java

Event Type	Source	TORDER		Gold TORDER		TLINK		Gold TLINK	
		Exact	Partial	Exact	Partial	Exact	Partial	Exact	Partial
Precise	Event-DCT	0.586	0.866	0.739	0.866	0.387	0.570	0.525	0.545
	Event-Time	0.384	0.555	0.577	0.619	0.216	0.288	0.412	0.447
	All	**0.660**	**0.870**	0.835	0.930	0.444	0.611	0.595	0.617
Imprecise	Event-DCT	**0.351**	0.631	0.530	0.647	0.234	0.395	0.364	0.449
	Event-Time	0.074	0.217	0.119	0.184	0.051	0.133	0.200	0.227
	All	0.299	**0.642**	0.509	0.686	0.252	0.429	0.444	0.517
Overall	Event-DCT	0.482	0.762	0.619	0.769	0.319	0.493	0.454	0.503
	Event-Time	0.259	0.419	0.393	0.444	0.149	0.255	0.326	0.358
	All	**0.501**	**0.769**	0.646	0.822	0.360	0.530	0.528	0.573

Table 5: The comparison of the cross-validation performance in the time anchor prediction task. 'Exact' and 'Partial' denote the two evaluation metrics: *exact match* and *partial match* accuracy. 'Gold' denotes the oracle performance of using the gold TORDERs or gold TLINKs as the input of the second-step.

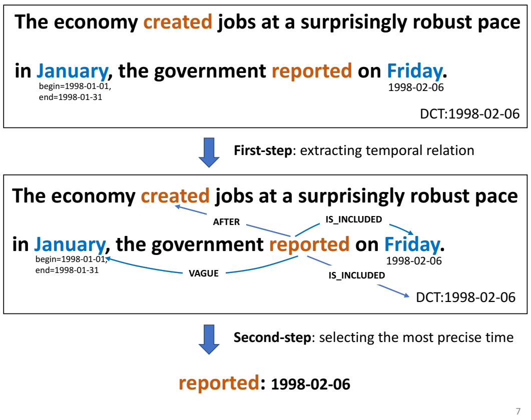

The economy created jobs at a surprisingly robust pace
in January, the government reported on Friday.
begin=1998-01-01,
end=1998-01-31
1998-02-06
DCT:1998-02-06
First-step: extracting temporal relation
The economy created jobs at a surprisingly robust pace
AFTER
IS_INCLUDED
in January, the government reported on Friday.
begin=1998-01-01,
end=1998-01-31
VAGUE
IS_INCLUDED
1998-02-06
DCT:1998-02-06
Second-step: selecting the most precise time
reported: 1998-02-06
7

Figure 3: The two-step process to determine the event anchors proposed in (Reimers et al., 2016).

with the TORDERs, we slightly modified the algorithm by replacing the TLINK relations with similar TORDER relations. SAME_DAY replaces *simultaneous* to predict precise dates, although their definition is quite different.

5.3 Experiment Settings

We perform a 6-fold cross-validation strategy to predict all the TORDERs and TLINKs of the mention pairs in the 36 documents of the TimeBank-Dense corpus. In each run, we split 30 documents for training and validation to predict the other 6 test documents.

We define two evaluation metrics, i.e. *Exact Match* accuracy and *Partial Match* accuracy to measure the performance in this task as follows:

$$\textit{exact match} = \frac{\#\textit{Number of the exact match predictions}}{\#\textit{Total number of the test samples}}$$

$$\textit{partial match} = \frac{\#\textit{Number of the partial match predictions}}{\#\textit{Total number of the test samples}}$$

We define two *partial match* cases: 1) a precise *(1998-02-06)* is *partial match* with an imprecise *(after 1998-02-06)*, if the date values are the same. 2) *(after 1998-02-06)* is *partial match* with *(after 1998-02-06, before 1998-02-21)*, if one is a part of the other.

5.4 Main Results

Table 5 summarizes the main results of the two-step time anchor prediction system trained with TORDER and TLINK data. 'Precise', 'Imprecise' and 'Overall' denote the results of predicting time anchors of precise events, imprecise events, and overall performance. 'Event-DCT' or 'Event-Time' denotes the second-step selection takes only Event-DCT or Event-Time pairs as input, which helps us to investigate how much information is provided by the different types of pairs for predicting the final time anchors. The new TORDERs show significantly superior out-performance in all three settings (i.e. only Event-DCT pairs, only Event-Time pairs, or Event-DCT + Event-Time), compared to the TLINKs. With both Event-DCT and Event-Time temporal order information, the system achieves the highest overall *exact match* and *partial match* accuracy.

The Event-DCT, Event-Time pairs are the source of temporal information for predicting *time anchors*. The system only using the Event-DCT achieves surprisingly high accuracy, particularly on the TORDER *partial match* accuracy of the

	Exact	Partial
CAEVO	0.442	0.553
Bi-LSTM TLINK	0.437	0.550
Bi-LSTM TORDER	0.586	0.811

Table 6: The comparison to the state-of-the-art dense TLINK classifier

precise events. The reason is that most events reported in news articles usually occur in precisely the same day as DCT. Therefore, the TORDER Event-DCT is benefited from the low proportion of vague relations, which sharply outperforms the TLINK Event-DCT by 16.3% overall *exact match*. However, the contribution of the Event-Time to the overall might be underestimated in this task somehow. The TORDER Event-Time still beats the TLINKs by 11% overall *exact match* and 16.4% overall partial match. Furthermore, the Event-Time encoding the temporal information within 1-sentence window in our experiments can be easily strengthen by our TORDER proposal to introduce more inter-sentence pairs.

5.5 Comparison to a state-of-the-art dense TLINK classifier

In this section, we perform an additional experiment to make a comparison to a system with the first-step replaced by a state-of-the-art dense TLINK classifier CAEVO (Chambers et al., 2014). We adopt the data split setting in Section 4.3 for three classifiers: CAEVO, Bi-LSTM classifier trained with TLINKs and Bi-LSTM classifier trained with TORDERs.

The results are summarized in Table 6. CAEVO achieves the *exact match* accuracy slightly better than the Bi-LSTM model trained with the TLINKs. The Bi-LSTM model trained with the TORDERs sharply outperforms the other two systems by approximate 14% *exact match* accuracy and approximate 26% in *partial match* accuracy.

6 Conclusion

In this paper, we propose a new approach to obtain temporal relations based on time anchors (i.e. absolute time value) of mentions in news articles. Our pre-defined generation rules can automatically induce TORDER relations by comparing the temporal order of two time anchors in a timeline. The requirement of our proposal for annotating time anchors is much easier compared to conventional methods, as the annotation effort is linear with the number of mentions. The TORDER data used in this paper are publicly available. The analysis, empirical comparison and classification results of the new TORDERs and the TimeBank-Dense TLINKs show our new data achieve the low VAGUE proportion, the informative relation types and the balanced label distribution. We perform the second evaluation of using the temporal relation classifier to complete the downstream task of time anchor prediction in news articles. The main results show our TORDERs significantly outperform the TLINKs in this task, which suggests our proposal has the capability to encode informative temporal order information with less annotation effort.

The main limitation of TORDER is that events are required to be anchored in a timeline. Strötgen and Gertz (2016) introduce the highly different characteristics of time expressions in four domains of text. It suggests that our proposal is difficult to be applied in some domains. One possible solution is to adopt a hybrid annotation method to annotate a target event towards the most relevant event (TLINK-style), when temporal information is absent in its context. Although this work is motivated for contributing to timeline applications, evaluating this proposal in the temporal question answering is also valuable. **SAME_DAY** could be harmful because this task possibly requires to know the exact order between two events occurring on the same day. It is worth conceiving a more general solution to improve the limitations of TORDER in the future work.

Acknowledgments

We would like to thank the anonymous reviewers for their valuable comments and thank Jason Bennett for useful discussions and proofreading.

References

James F Allen. 1990. Maintaining knowledge about temporal intervals. In *Readings in qualitative reasoning about physical systems*, Elsevier, pages 361–372.

Klaus Berberich, Srikanta Bedathur, Omar Alonso, and Gerhard Weikum. 2010. A language modeling approach for temporal information needs. In *European Conference on Information Retrieval*. Springer, pages 13–25.

Philip Bramsen, Pawan Deshpande, Yoong Keok Lee, and Regina Barzilay. 2006. Inducing temporal graphs. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pages 189–198.

Taylor Cassidy, Bill McDowell, Nathanael Chambers, and Steven Bethard. 2014. An annotation framework for dense event ordering. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics, Baltimore, Maryland, pages 501–506. `http://www.aclweb.org/anthology/P14-2082`.

Nathanael Chambers, Taylor Cassidy, Bill McDowell, and Steven Bethard. 2014. Dense event ordering with a multi-pass architecture. *Transactions of the Association for Computational Linguistics* 2:273–284. `http://aclweb.org/anthology/Q/Q14/Q14-1022.pdf`.

Angel X. Chang and Christopher Manning. 2012. Sutime: A library for recognizing and normalizing time expressions. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Mehmet Uğur Doğan, Bente Maegaard, Joseph Mariani, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012)*. European Language Resources Association (ELRA), Istanbul, Turkey, pages 3735–3740. ACL Anthology Identifier: L12-1122. `http://www.lrec-conf.org/proceedings/lrec2012/pdf/284_Paper.pdf`.

Fei Cheng and Yusuke Miyao. 2017. Classifying temporal relations by bidirectional lstm over dependency paths. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics, Vancouver, Canada, pages 1–6. `http://aclweb.org/anthology/P17-2001`.

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

Jennifer D'Souza and Vincent Ng. 2013. Classifying temporal relations with rich linguistic knowledge. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Atlanta, Georgia, pages 918–927. `http://www.aclweb.org/anthology/N13-1112`.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9(8):1735–1780. `https://doi.org/10.1162/neco.1997.9.8.1735`.

Oleksandr Kolomiyets, Steven Bethard, and Marie-Francine Moens. 2012. Extracting narrative timelines as temporal dependency structures. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*. Association for Computational Linguistics, pages 88–97.

Natsuda Laokulrat, Makoto Miwa, Yoshimasa Tsuruoka, and Takashi Chikayama. 2013. Uttime: Temporal relation classification using deep syntactic features. In *Second Joint Conference on Lexical and Computational Semantics (* SEM)*. volume 2, pages 88–92. `http://aclweb.org/anthology/S/S13/S13-2015.pdf`.

Hector Llorens, Nathanael Chambers, Naushad UzZaman, Nasrin Mostafazadeh, James Allen, and James Pustejovsky. 2015. Semeval-2015 task 5: Qa tempeval - evaluating temporal information understanding with question answering. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*. Association for Computational Linguistics, Denver, Colorado, pages 792–800. `http://www.aclweb.org/anthology/S15-2134`.

Inderjeet Mani, Marc Verhagen, Ben Wellner, Chong Min Lee, and James Pustejovsky. 2006. Machine learning of temporal relations. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Sydney, Australia, pages 753–760. `https://doi.org/10.3115/1220175.1220270`.

Yuanliang Meng, Anna Rumshisky, and Alexey Romanov. 2017. Temporal information extraction for question answering using syntactic dependencies in an lstm-based architecture. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Copenhagen, Denmark, pages 887–896. `https://www.aclweb.org/anthology/D17-1092`.

Anne-Lyse Minard, Manuela Speranza, Eneko Agirre, Itziar Aldabe, Marieke van Erp, Bernardo Magnini, German Rigau, and Ruben Urizar. 2015. Semeval-2015 task 4: Timeline: Cross-document event ordering. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*. Association for Computational Linguistics, Denver, Colorado, pages 778–786. `http://www.aclweb.org/anthology/S15-2132`.

Paramita Mirza and Sara Tonelli. 2016. On the contribution of word embeddings to temporal relation classification. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*. The COLING 2016 Organizing Committee, Osaka, Japan, pages 2818–2828. `http://aclweb.org/anthology/C16-1265`.

James Pustejovsky, Patrick Hanks, Roser Sauri, Andrew See, Robert Gaizauskas, Andrea Setzer, Dragomir Radev, Beth Sundheim, David Day, Lisa Ferro, et al. 2003. The timebank corpus. In *Corpus linguistics*. volume 2003, page 40. https://catalog.ldc.upenn.edu/LDC2006T08.

Nils Reimers, Nazanin Dehghani, and Iryna Gurevych. 2016. Temporal anchoring of events for the timebank corpus. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Berlin, Germany, pages 2195–2204. http://www.aclweb.org/anthology/P16-1207.

Roser Saurı, Jessica Littman, Bob Knippen, Robert Gaizauskas, Andrea Setzer, and James Pustejovsky. 2006. Timeml annotation guidelines version 1.2. 1.

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

Jannik Strötgen and Michael Gertz. 2015. A baseline temporal tagger for all languages. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Lisbon, Portugal, pages 541–547. http://aclweb.org/anthology/D15-1063.

Jannik Strötgen and Michael Gertz. 2016. Domain-sensitive temporal tagging. *Synthesis Lectures on Human Language Technologies* 9(3):1–151.

Naushad UzZaman, Hector Llorens, James Allen, Leon Derczynski, Marc Verhagen, and James Pustejovsky. 2012. Tempeval-3: Evaluating events, time expressions, and temporal relations. *arXiv preprint arXiv:1206.5333* http://aclweb.org/anthology/S/S13/S13-2001.pdf.

Marc Verhagen, Robert Gaizauskas, Frank Schilder, Mark Hepple, Jessica Moszkowicz, and James Pustejovsky. 2009. The tempeval challenge: identifying temporal relations in text. *Language Resources and Evaluation* 43(2):161–179. https://doi.org/10.1007/s10579-009-9086-z.

Marc Verhagen, Roser Sauri, Tommaso Caselli, and James Pustejovsky. 2010. Semeval-2010 task 13: Tempeval-2. In *Proceedings of the 5th international workshop on semantic evaluation*. Association for Computational Linguistics, pages 57–62. http://www.aclweb.org/anthology/S10-1010.