Anchoring Events to the Time Axis toward Storyline Construction

Tomohiro Sakaguchi

March 2019

Abstract

To understand the present situations and predict the future, it is essential to look at the past. Since social situations and people's sense of values are constantly changing, most facts are not true eternally. In order to overview a transition of interpretations about events and things, it is necessary to read texts in a chronological order and organize information along the time axis as a *storyline*. However, there is a large amount of text on the Web, and since it is no longer impossible to read all the related text manually, a new tool is required.

Natural language processing (NLP) is a technology to analyze text using computers and is used in various applications such as information retrieval, opinion analysis, summarization, question answering, and machine translation. A text includes event information in the past, present, and future, so there have been many studies analyzing temporal information in a document. However, they mainly focus on relative local relations between events and between event and time, and there are few studies anchoring events to the time axis with the consideration of global information.

In this thesis, we present (i) temporal expressions analysis, (ii) temporal corpus construction, and (iii) temporal information analysis and timeline generation, toward storyline generation. First, we propose a model which recognizes and normalizes temporal expressions in text. Temporal expressions are fundamental for textual temporal analysis and many models have been proposed. However, analyzing loose structures of temporal expressions is a remained problem. To overcome this problem, we propose a neural network model that recognizes and normalizes loose structured temporal expressions via multi-task learning.

Then, we propose a new annotation scheme for anchoring expressions in text to the time axis comprehensively. The points of our annotation scheme are two-fold: annotat-

ing various expressions that can have temporality, and annotating various types of time information.

Finally, we analyze the relations between time and events and propose a model to construct timelines. We analyze event-event temporal and subevent relations and design three multi-class classification tasks for understanding temporal information of events. We then propose a timeline generation model which uses a wide context and external knowledge.

Acknowledgments

京都に来てからの 6 年間、数多くの方々に支えられ、助けられ、励まされ、ときには叱られ、私の「ストーリーライン」は豊かに紡がれました。

なかでも最も深い感謝の意を表したいのは、偉大なる恩師・黒橋禎夫教授です。黒橋先生は、思いついたことや興味をもったことにすぐに飛びついてしまう私を、ときにはなだめ、ときには励まし、ときには見守り、私を導いてくださりました。どんなときでも、たとえ深夜や早朝であっても、時間を割いて熱心にご指導いただき、また、常に最高の環境を用意していただき、感謝の念に堪えません。黒橋先生の瞬く間に本質を見抜く力は、私の憧れです。また先生には、状況に応じた話し方や心の持ち方、組織や社会での振る舞い方など、研究以外のことも多く教えていただきました。先生に少しでも近づけるよう、これからも精進していきたいと思います。

西田豊明教授と楠見孝教授には、学位論文を審査していただき、多くの貴重な助言をいただきました。厚く御礼申し上げます。先生方には修士課程のときから研究を見守っていただき、私自身が気づいていなかった研究の魅力や可能性を教えていただきました。

研究室の方々にも数多くの支援をいただきました。河原大輔准教授と柴田知秀講師には、研究を進めるにあたって基本的なことから実践的なことまであらゆる相談に乗っていただき、根気強く指導していただきました。森田一研究員、林部祐太研究員、村脇有吾助教、田中リベカ研究員には、いつでも快く議論に付き合っていただき、多くの建設的な助言をいただきました。

同期の栗田修平くん、Raj Dabre くんは、日頃から様々なことについて一緒に話し考えてくれました。優秀な同期に恵まれ、多くの刺激を得て、ここまでやってくることができました。多くの後輩にもお世話になりました。特に、岸本裕大くん、Arseny Tolmachev くん、澤田晋之介くんには、研究に限らず色々なことに付き合っていただきました。Reid Pryzant さん、Tariq Alkhaldi さんには、何度も英語を見て

いただきました。秘書の芦原裕子さん、吉利菜帆さんには、物品購入や出張手続きなどを一手に引き受けていただき、研究生活を支えていただきました。

研究室の卒業生の方々にも多くの機会でお世話になりました。中澤敏明さん、新里圭司さん、泉朋子さんには、研究内容や研究姿勢についてアドバイスをいただきました。萩行正嗣さんには、良いときも悪いときも親身になって相談に乗っていただきました。自然言語処理技術を使って新しい分野を開拓していく萩行さんの姿は私の目標です。修士課程の同期である小浜翔太郎くん、町田雄一郎くんには、修了後もずっと様々な形で助けていただきました。深く感謝いたします。

石川真奈見さん、二階堂奈月さん、堀内マリ香さんにはコーパス作成を行っていただきました。皆さまとの議論はアイディアのタネに富むものであり、研究の大きな原動力となりました。国立国語研究所の浅原正幸准教授には、日本語の時間情報表現について議論していただき、有益な助言をいただきました。

デザイン学大学院連携プログラムには、修士・博士課程を通して研究を支援していただき、また、自らの専門技術を社会でどのように活かすかを考える機会をいただきました。他分野の学生や研究者と議論する機会も多く、それぞれの専門分野で私の研究内容がどのように位置づけられるかを考えることができたのは非常にありがたかったです。なかでも、中小路久美代教授と北雄介講師に教えていただいた「ライン」という考え方は、この論文の根底を流れるものとなっています。

プログラムのメンバーとの活動や議論も刺激的なものでした。特に、同期の久富望さん、佐藤那央さん、小椋恵麻さん、古田幸三くんとは、多くの活動を共にし幾度となく助けてもらいました。また、阿部将和さん、小東茂夫さんには、自分の視野を広げる数多くの機会、知見、助言をいただき、充実した大学院生活を送ることができました。

自分が好きな研究を好きなだけできる日々は楽しいものでしたが、思うようにいかないことの連続でもありました。そんなとき、古川恵三さん、田中智彦さんには、親身になって考えていただき多くのアドバイスをいただきました。尾本久美子さんは、常に状況をポジティブ捉え、励ましてくれました。この他にも、数多くの方々から援助や機会をいただくことができ、今日に至ることができました。心から御礼を申し上げます。

最後に、今まであたたかく応援してくれた両親と弟に感謝します。

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

"The farther back you can look, the farther forward you are likely to see," Winston Churchill is reputed to have said. Knowing what happened in the past and how things and interpretations have been changed is essential to think about the present and the future. The importance has increased in this turbulent era.

Since people's sense of values and social situations are constantly changing, the truth of facts and interpretations change gradually. For example, there are many events that had significant impacts at that time, but are commonplaces now. There are also some events that were not focused on but are now famous. Thus, looking over the sequence of events along the time axis is essential.

The Web is a treasure trove of written text that has been accumulating for twenty years. This information space is starting to include not only the latest information but also events and knowledge in the past. However, it is no longer impossible to comprehensively read all the related text on the Web manually. *Storyline* is a structured chronology, which integrates and organizes the massive information related to a certain topic along the time axis and provides a reader-friendly knowledge. A storyline consists of various types of information such as events, actors (e.g., person, location and time), emotions, value judgement, opinions, and their relations. Figure 1.1 shows an example of storyline about a shogi player Yoshiharu Habu. Through various events in his life (e.g., shogi games), his storyline interacts with storylines of other players. It is not only persons that construct

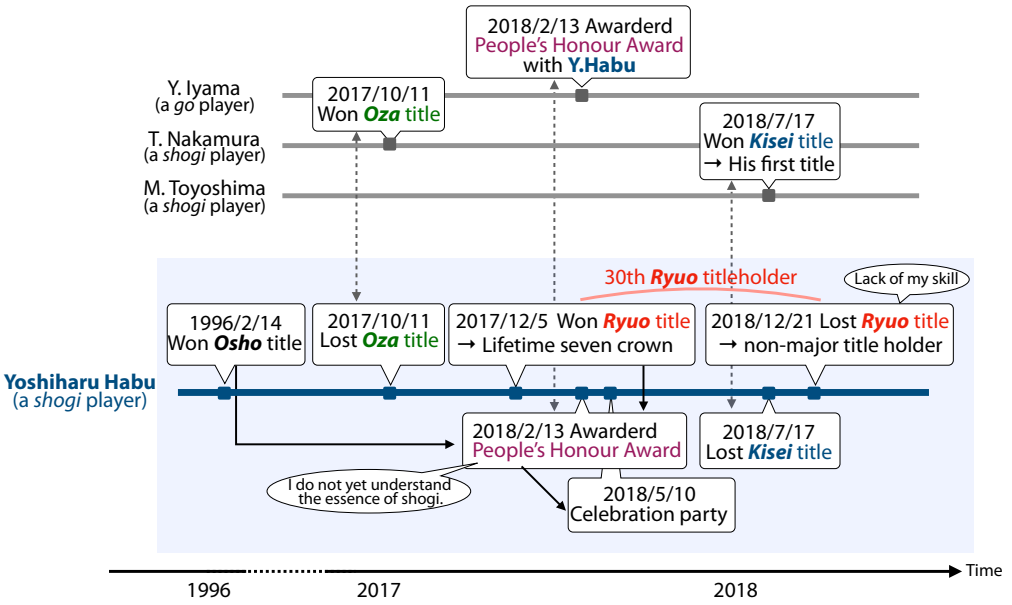

Figure 1.1: A storyline of Yoshiharu Habu, a *shogi* player (central dot line). His storyline interacts with storylines of other *shogi* and *go* players. Squares indicate events and circles indicate remarks.

storylines. Things, locations, products, and shogi titles also have storylines. Storylines can express the changes in the world.

The core skeleton is anchoring events to the time axis, namely, *timeline*. A timeline is just a chronological ordered events and it does not include complicated relations between events. However, timeline construction technique is not only useful for storyline construction but for various applications such as question answering, information retrieval and summarization. For example, it is helpful to answer a question "What is the most significant sales product for Sony until 1970?" In this thesis, we focus on timeline construction.

1.2 Previous Studies on Storyline and Timeline in NLP

Natural language processing (NLP) is a technology to analyze text using computers and is used in various applications such as information retrieval, opinion analysis, summa-

rization, question answering, and machine translation. With the evolution of the Web, storyline and timeline generation has attracted attention. They have been studied in the areas of topic detection and tracking, information retrieval, and multi-document summarization. For example, traditional information retrieval provides articles ranked by query relevance, but when users are interested in a transition of events, individual but correlated articles should be provided. Storyline and timeline generation have been studied focusing on the following three factors: extracting salient events, anchoring events to time, and incorporating various semantic relations.

1.2.1 Salient Sentences Extraction

To construct reader-friendly timelines, many studies focused on extracting salient information, which is included in timelines. Most of the studies are sentence based.

Early works developed unsupervised approaches. Swan and Allan [79] generated timelines by extracting clusters of noun phrases and named entities. They modeled the arrival of the terms as a random process with an unknown binomial distribution. Using time tagged corpus, they first extracted unusual terms in text by χ^2 measure and then grouped them into isolated topics. Allan et al. [1] proposed a temporal news summarization method. They summarized a news stream on a topic by extracting useful and novel sentences using language model techniques. The usefulness captures the relatedness to topics, and the novelty captures the redundancy of information. Chieu and Lee [21] introduced two measures to rank sentences: burstiness and interest. They expanded the idea of Allan et al. [1] for their burstiness. They defined interesting events as events that are a subject of interest in many sentences.

Some studies used ranking and graph-based methods. Yan et al. [96] formulated the task as a balanced optimization problem via iterative sentence substitution. The objective function is defined by four attributes: query relevance, information coverage, coherence, and cross-date diversity. Yan et al. [95] extended the model by considering inter-date and intra-date dependencies between sentences. Zhao et al. [99] took into consideration social attention, which improved both the informativeness and interestingness of timelines.

There are some studies which rank events by focusing on temporal information. Hu et al. [34] detected breakpoints of a theme using the Hidden Markov Model and selected key sentences from each breakpoint. Kessler et al. [39] designed a new task, namely date

selection, which extracts a list of salient dates with respect to a given topic. Each date is presented with relevant sentences. Given a query, their system retrieves the relevant documents and dates are extracted from these documents. Then a classifier extracts the salient dates by considering temporal and linguistic information. While the model scores each date independently of other dates, Tran et al. [83] overcame the problem by using a graphical model.

1.2.2 Anchoring Events to Time

As will be mentioned in Section 1.3, there are temporal corpora which annotated event-time and event-event temporal relations in the same sentence, and there are many studies to determine those relations [50, 98, 29, 15, 23, 18]. However, to construct a timeline which represents comprehensive information, techniques to anchor every event in a document to time are required.

Do et al. [27] associated each event with a specific absolute time interval by using a global inference model. Their two local pairwise classifiers associate (i) event and time interval and (ii) event and event, and a joint inference model enforces global coherency constraints on them by using Integer Linear Programming (ILP). A time interval is represented as a pair of time endpoints, and it enables to perform more concise ILP formulation. They also incorporated event coreference knowledge, which performed a significant improvement. Moschitti et al. [57] associated each event with a temporal expression by considering the structural representation of sentences. They developed a bag-of-words tree representation capturing the context of the target temporal expression and event expression and encoded it as structural features in Support Vector Machines using tree kernels.

In 2015, a shared task *TimeLine: Cross-Document Event Ordering* was held, and a timeline generation task was performed. In the task, participants are required to extract events related to a query from multiple articles and anchor them to time. Moulahi et al. [58], Navarro and Saquete [59], Navarro and Saquete [60] and Laparra et al. [43] proposed unsupervised approaches using clustering techniques. Cornegruta and Vlachos [25] introduced a machine-learning approach, which anchors events to temporal expressions. The details of these models will be described in Section 4.3.

1.2.3 Incorporating Various Semantic Relations

Hu et al. [35] defined the storyline task as a chain of events that characterize a certain aspect of the given topic and involve the same set of actors and places and proposed a model generating storylines of a topic. Storylines interact through *informative events*, which have strong correlations such as cause and effect. The model first calculates the coherence between a pair of news articles, and build a coherence graph. Then salient informative events are identified from the graph and organized as storylines of a topic.

Recently, several workshops and competitions focusing on the general and basic characteristic of storyline and timeline are held. *Computing News Storylines* workshops are held in 2015[1] and 2016[2]. In these workshops, a stream of daily news reports is regarded as "story," and storyline representation, detection, evaluation, event detection, and corpus analysis are discussed. For example, Vossen et al. [91] defined the term "story," using the narratology framework of Bal [8], as a particular way or style in which something is told. The story involves the following three elements: exposition, predicament, and extrication. They then proposed a storyline model, which connects events and represents the internal components of a story such as rising actions, climax, falling actions, and resolution. The model first extracts events which are anchored to time, and salient events are selected. Then, events which are related to the salient events are extracted, and they construct a storyline.

1.3 Notion of Relations between Event and Time

In this thesis, we focus on techniques to anchor events to time. Here, we describe how temporal information has been interpreted in NLP. There are mainly three viewpoints of interpreting relations between event and time (Figure 1.2). The first focuses on relative temporal relations between event-event and event-time. TLINK (temporal link) defined in TimeML (Time Markup Language) is a typical example. The second is a narrative container, which expanded the preceding approach to able to consider a document level information. The third anchors events to the time axis directly, which is an intuitive and direct representation for timeline construction. While the preceding relative approaches

[1] https://sites.google.com/site/computingnewsstorylines2015

[2] https://sites.google.com/site/newsstorylines2016

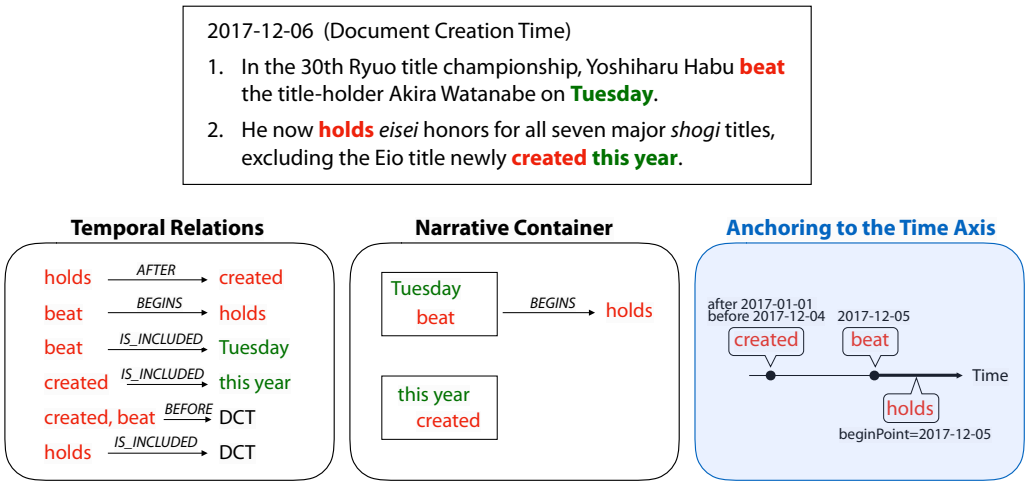

Figure 1.2: Three viewpoints of interpreting time.

focus on the local relations, this approach represents a wider context and implicit information. Thus, we mainly focus on the third approach in this thesis.

1.3.1 Relative Temporal Relations

From January to July in 2002, a workshop called TERQAS[3] was held to address the problem of how to answer temporally-based questions about the events and entities in news articles, such as "Is Gates currently CEO of Microsoft?" The workshop was the first attempt to focus on the relationship between events and time. There were two deliverables of the workshop:

1. To design a common standard for events and their temporal anchoring in text (to be called Time Markup Language, TimeML).
2. To create a human-annotated corpus marked up for temporal expressions, events, and temporal relations, based on the TimeML specification (TimeBank).

Through some revisions [64, 65], TimeML Annotation Guidelines Version 1.2.1 [72] and TimeBank 1.2 [67] which contains 183 news articles that have been annotated according to TimeML specification were constructed in 2006. The TimeML consists of

[3]The TERQAS (Time and Event Recognition for Question Answering Systems) workshop was held at MITRE Bedford and Brandeis University.

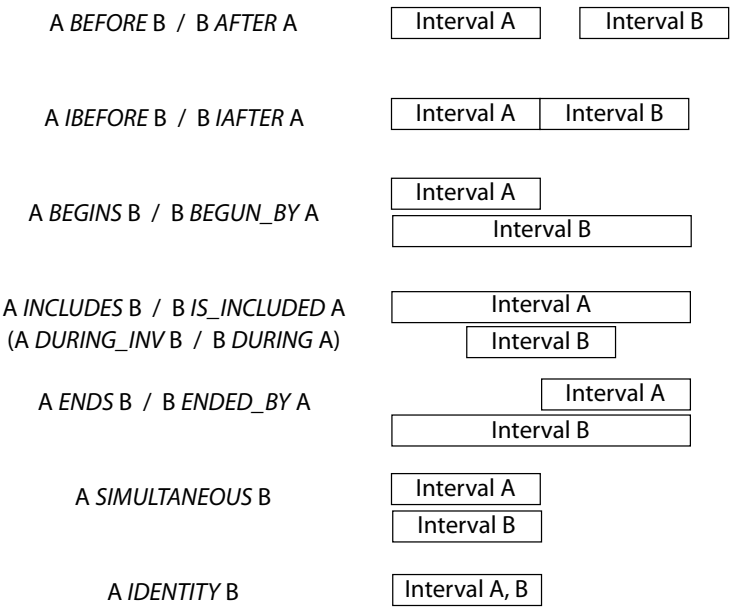

Figure 1.3: Temporal relation types in TLINK. Based on Allen's interval algebra, 14 relative temporal relations between two temporal intervals are defined.

four primary tag types: *TIMEX3* for temporal expressions, *EVENT* for temporal events, *SIGNAL* for temporal signals, and *LINK* for representing relationships:

1. TIMEX3 tag represents temporal information of temporal expressions, and it includes some attributes such as type, value, premodifier, quantifier, frequency, and temporal anchors. The type is one of *DATE*, *TIME*, *DURATION*, and *SET*. The value is represented using the ISO-8601 standard. For example, "today" in a document whose creation time (DCT) is "January 16, 2019" is annotated with *DATE* in type and `2019-01-16` in value.

2. EVENT tag is annotated with events, which represent situations that *happen* or *occur*. It includes some attributes such as occurrence, state, aspectual and reporting. TimeML distinguishes between event *tokens* and event *instances*, and in addition to the EVENT tag, events are also annotated with one or more MAKEINSTANCE tags which represent the actual realization of events such as part of speech, tense, aspect, modality, and polarity. For example, in the following sentence, the verb

"taught" represents two events, one has a positive polarity, and one has negative.

(1) John ***taught*** on Monday but not on Tuesday.

Each instance has a temporal relation with "Monday" and "Tuesday."

3. SIGNAL tag is annotated with temporal function words which represent temporal relations in event-time (e.g., event expression-temporal expression), event-event, time-time, such as "before," "while," "when," and "to."
4. LINK tag consists of three tags, TLINK, SLINK, and ALINK. They associate the three tag types above with each other. TLINK represents the temporal order between event-event and event-time. There are 14 relations such as `BEFORE` and `IS_INCLUDED`, which are based on Allen's interval algebra [2] (Figure 1.3). The difference between `INCLUDED` and `DURING_INV` is that `DURING_INV` is used when an event persists throughout a duration. `INCLUDES`, `IS_INCLUDED`, and `IDENTITY` are able to represent a set/subset relation between events. SLINK represents subordination relations between two events. Six relations, modal, factive, counter-factive, evidential, negative evidential, and conditional are defined. ALINK represents aspectual relations between an aspectual event and its argument event. Five relations, initiation, culmination, termination, continuation and reinitiation are defined.

Based on the annotation scheme and corpus, many shared tasks about temporal information analysis are designed in SemEval (the International Workshop on Semantic Evaluations) competitions. Through the competitions, temporal analysis tasks, data, and techniques are developed.

In *TempEval Temporal Relation Identification* task in SemEval-2007 (TempEval-1) [87, 86], there were three subtasks: determine the relation between (a) event-time in the same sentence, (b) event-DCT and (c) the main events of two consecutive sentences. For the task, the Timebank corpus with a simplified version of TimeML was provided. In the corpus, TIMEX3, EVENT and TLINK tags are annotated. The temporal relation tasks are not easy for humans, and the inter-annotator agreement was 0.72 in event-time task (task a,b), and 0.65 in event-event task (task c).

In *Evaluating Events, Time Expressions, and Temporal Relations* task in SemEval-2010 (TempEval-2) [88], there were six subtasks: (a) temporal expressions extraction and

normalization, (b) event extraction and classification, (c) determine the temporal relation between event-time in the same sentence, (d) event-event in the consecutive sentences, (e) event-DCT, (f) two events that one subordinates another. Tasks (a), (b) and (f) were new. Data on five languages were provided: English, Italian, Chinese, Spanish, and Korean.

In *Evaluating Time Expressions, Events, and Temporal Relations* task in SemEval-2013 (TempEval-3) [85], there were three basic subtasks in English and Spanish: (a) temporal expressions extraction and normalization, (b) event extraction and classification, (c) determine the temporal relations between (i) main events of consecutive sentences, (ii) event-event in the same sentence, (iii) event-time in the same sentence and (iv) event-DCT. Additionally, end-to-end temporal relation task, i.e., first extract events and temporal expressions, and then determine temporal relations, is performed.

The full set of temporal relations in TimeML are used, while the reduced set is used in previous competitions. The size of the dataset was also expanded. In the task, 100K word gold data and 600K word silver data were used. The existing corpora (TimeBank and AQUAINT[4]) were corrected, and new corpora were released.

Based on the basic temporal tasks in TempEval competitions, some application tasks were performed. *TimeLine: Cross-Document Event Ordering* task in SemEval-2015 [55] aimed to generate timelines from multiple English news articles. Given a set of articles and target entities, events related to the time entities in the articles are extracted and ordered along the time axis. The target entity is one of person (e.g., Steve Jobs), organization (e.g., Apple Inc.), product (e.g., Airbus A380) and financial entity (e.g., NIKKEI). The task consists of two tracks: Track A and Track B. In Track A, raw texts are given as input, and in Track B, texts with gold event mentions are given. The dataset is composed of articles from Wikinews[5].

1.3.2 Narrative Container Relations

Since TLINK in TimeML focuses on intra-sentence relations, there are many unanchored events. To overcome the issue, Pustejovsky and Stubbs [66] introduced a document level information structure, called *narrative container*. A narrative container is a bucket con-

[4] http://universal.elra.info/product_info.php?cPath=42_43&products_id=2333

[5] https://en.wikinews.org

taining events which are discussed in the text, and it is anchored to temporal expressions or other events across sentences. Black boxes in Figure 1.2 represent narrative containers containing events. For example, events "held" and "beat" are grouped into a time "Tuesday."

Styler IV et al. [78] modified the properties of TIMEX3 and EVENT for the clinical domain and linked them with temporal relations including narrative container relations. There are five temporal links, `BEFORE`, `OVERLAP`, `BEGINS-ON`, `ENDS-ON` and `CONTAINS`. `CONTAINS` represents the narrative container relations while the others are the derivation of TLINK. Using the new annotation scheme, they constructed THYME corpus (Temporal Histories of Your Medical Events), which consists of 1,254 clinical notes related to two oncology: brain cancer and colon cancer.

Using the THYME corpus, Clinical TempEval competitions [10, 11, 12] were performed. In the competitions consist of nine subtasks which are grouped into three categories: (1) time expressions recognition and classification, (2) event expressions recognition and classification, (3) identifying temporal relations between event-DCT and narrative container relations.

1.3.3 Anchoring Events to the Time Axis

Another viewpoint of time is anchoring events directly to the time axis.

In 2016, Reimers et al. [70] proposed an annotation scheme which anchors events to the time axis. They divided events into two types: single day events and multiple day events. The former is annotated with the date on which the event occurred, and the latter is annotated with the start dates (*beginPoint*) and end dates (*endPoint*) of the event. For example, ***sent*** in the following sentence, an event which ends in one day, is annotated with *1980-05-26*, and ***spent***, an event spanning multiple days, is annotated with *beginPoint=1980-05-26 endPoint=1980-06-01*.

(2) He was ***sent*** into space on May 26, 1980. He ***spent*** six days aboard the Salyut 6 spacecraft.

In the case that the exact event date is not mentioned, notations *before* and *after* being used. In the following sentence, ***appointed*** is annotated with *after 1996-01-01 before*

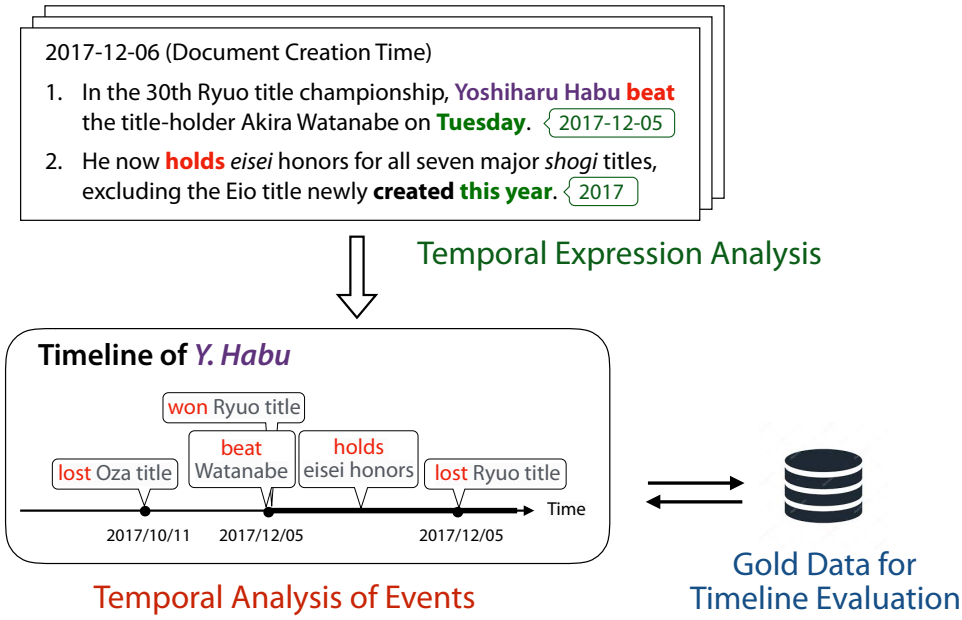

Figure 1.4: Three works in this study: Temporal expression analysis, Temporal corpus construction and Timeline construction.

1996-12-31, and ***part*** is annotated with *beginPoint=after 1984-10-01 before 1984-10-31 and endPoint=after 1984-10-01 before 1984-10-31*.

(3) In 1996 he was ***appointed*** military attache at the Hungarian embassy in Washington. [...] McBride was ***part*** of a seven-member crew aboard the Orbiter Challenger in October 1984.

The annotation scheme anchors events more accurate than the previous approaches while it is difficult to anchor events to the time axis in texts with few temporal expressions, such as novels.

1.4 Contribution of this Thesis

In this thesis, we present textual temporal analysis techniques and temporal corpus construction toward storyline construction. The contributions of this study are three-fold (Figure 1.4).

1. Analysis of loose structured temporal expressions
   Temporal expressions are fundamental for textual temporal analysis, and many models have been proposed. However, analyzing loose structures of temporal expressions is a remained problem. To overcome this problem, we proposed a neural network model that recognizes and normalizes loose structured temporal expressions via multi-task learning.
2. Temporal corpus construction
   The previous time axis viewpoint corpus represents time information by *beginPoint*, *endPoint*, *before* and *after*, and cannot represent complex time information. We proposed a new annotation scheme for anchoring expressions in text to the time axis comprehensively. The points of our annotation scheme are two-fold: annotating various expressions that can have temporality, and annotating various types of time information.
3. Timeline construction considering global context
   Since the number of temporal expressions is small, interpreting the temporal nature of events is essential. We first studied temporal information of events. We analyzed event-event temporal and subevent relations and designed three multi-class classification tasks for anchoring events to the time axis. We then proposed a timeline generation model which uses a wide context and external knowledge.

1.4.1 Temporal Expressions Analysis

Temporal expressions represent temporal information explicitly so that they are the fundamental information in the textual temporal analysis. The main tasks in temporal expressions analysis are the recognition and normalization of temporal expressions, which are performed in TempEval competitions. The recognition task is extracting temporal expressions from raw texts. Although most of the words consisting temporal expressions are specific (e.g., “January,” “Monday,” “month”), some words are not (e.g., “fall,” “period,” “last”). Numerical words (e.g., “31,” “2019”) and modifiers (e.g., “about,” “the”) also consist of temporal expressions. To recognize these expressions, it is essential to consider a context. Furthermore, temporal expressions form a loose structure such as “January 2019,” “January, 2019” and “2019 January.” The normalization task is convert-

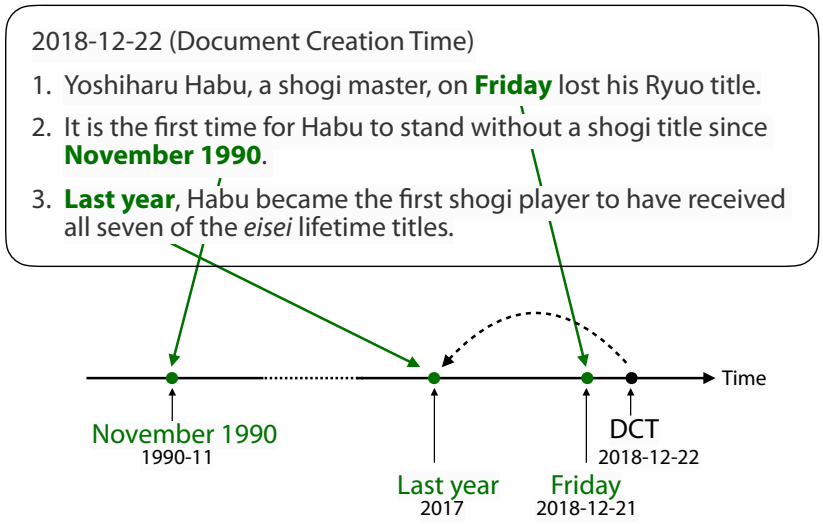

Figure 1.5: An example of temporal expressions recognition and normalization.

ing the recognized temporal expressions to a normalized format considering a context. Figure 1.5 shows an example. While "November 1990" can be normalized only from the lexical information of the temporal expression, "Friday" and "Last year" requires contexts to be normalized. Since "Friday" comes once a week, systems have to determine the appropriate date from the context. To normalize "Last year," it is essential to determine the reference date from the context. If the system wrongly assumes "November 1990" to be the reference date, "Last year" would be normalized to *1989*.

The tasks have been tackled with rule-based, machine learning based and semantic parsing approaches. Through the previous studies, it became possible to recognize ambiguous temporal expressions correctly and to normalize in consideration of contexts. However, normalizing loose structured temporal expressions has not been fully coped with. In this thesis, we propose a neural network model that recognizes and normalizes loose structured temporal expressions via multi-task learning. We only prepare a set of basic temporal interpretation rules for basic temporal expressions, and we simply rely on the neural model to robustly compose them to absorb the rich diversity of temporal expressions. Experimental results showed that the model achieved a new state-of-the-art.

1.4.2 Temporal Corpus Construction for Storyline

As described in the preceding section, previous corpora have been focused on intra-document relative temporal relations. However, the approach is not always accurate in inter-document event anchoring to the time axis. Recently, annotation schemes and corpora which directly anchor events to the time axis have been constructed. The approach has an advantage in anchoring events to the time axis accurately compared with the relational approach so that it is more appropriate for training and evaluating timelines. Furthermore, the number of annotation scales linearly while the relational approaches require a quadratic number of links.

In this thesis, we designed a new annotation scheme that expanded the existing annotation scheme in the following three points and constructed a temporal corpus using the scheme.

1. In the previous annotation schemes, complex temporal expressions such as “once in two days” and “three days in next month” cannot be represented. We introduced new notations that can represent various temporal information.
2. Many previous studies “events” defined in TimeML, which express situations that happen or occur. However, the temporal information of expressions other than “event” also can be a clue to text understanding. We annotated wider expressions: all the expressions that can have temporality.
3. Most of the previous studies performed in English. In Japanese, BCCWJ-TimeBank, which is annotated in a relation approach, is the sole corpus. We constructed a new corpus based on Japanese newspaper.

The constructed temporal corpus consists of 113 documents with 4,534 expressions. Newly proposed tags account for approximately 25% of all tags. Since the corpus has already been annotated predicate-argument structures and coreference relations, it can be utilized for integrated information analysis of events, entities and time.

1.4.3 Temporal Information Analysis of Events and Timeline Construction

To construct a timeline, it is necessary to anchor each event to the time axis. Since the number of temporal expressions included in an article is small, not only interpreting the

relations between events and temporal expressions but also interpreting the temporal nature of events and temporal relations between events is essential. Although there have been many studies analyzing these tasks, the accuracy is not high enough for the small amount of training data. We focused on utilizing linguistic contexts and external knowledge to make up for the small amount of data.

In this thesis, we first analyzed event-event temporal relation and subevent relation. We proposed a neural network model which uses external knowledge and considers the intra-sentential context. The experimental results ranked first in TAC2017 (Text Analysis Conference) event sequencing task. However, the model only focuses on event information, and it does not adequately consider temporal information.

To study temporal information of events, we designed three multi-class classification tasks: (1) judge whether an event has temporality (two classes), (2) the temporal span of events (four classes), and (3) the occurrence time of events (five classes). We proposed three neural network models to solve each task, and they are evaluated using our temporal corpus. Although F-score of the first event temporality task was about 90 points, that of second and third tasks were about 50 to 60 points.

Finally, we constructed timelines using events which have strong temporality. We proposed a model which anchors events to time using wider context and external knowledge and constructed timelines from multiple documents. Our experimental results showed that our model surpasses the state-of-the-art system by 3.5 F-score points in the TimeLine task of SemEval 2015.

1.5 Outline of the Thesis

The rest of this thesis is structured as follows. In Chapter 2 we propose a method for recognizing and normalizing temporal expressions. We examine previously unexplored problems and point out the importance of the normalization of loose structured temporal expressions. We propose a neural network model which robustly composes basic time expressions and absorbs their rich diversity of combination.

In Chapter 3 we construct a temporal corpus which anchors various types of temporal information in text to the time axis. We present our annotation scheme, which defines the judgment of temporality and time tags representing temporal information. Finally, we

report statistics of the constructed corpus on Kyoto University Text Corpus and discuss properties of the corpus.

In Chapter 4 we describe our work on temporal information analysis of events. We first present a model to determine temporal and subevent relations between events. We then design three tasks to anchor events to the time axis and present models for each task. Finally, we propose a method of anchoring events to temporal expressions and construct timelines.

In Chapter 5 we summarize this thesis and describe areas for future work.

Chapter 2

Temporal Expressions Analysis

Temporal expressions are fundamental to temporal textual analysis. While the previous studies achieve high accuracy on the types of temporal expressions that match their compositional structure knowledge, they cannot cope with the rich diversity of temporal expressions in the wild. In this chapter, we present a neural network model that overcomes this issue, recognizing and normalizing temporal expressions via multi-task learning.

The rest of this chapter is organized as follows. In Section 2.1, we sort issues about temporal expressions analysis. In Section 2.2, we present related works. In Section 2.3, we present the time representation in the model. In Section 2.4, we present the neural network model. In Section 2.5, we show the experimental results and analyze the errors. In Section 2.6, we present the conclusion of this chapter.

2.1 Introduction

Temporal information is important for many natural language processing applications such as text summarization, information retrieval, and question answering. Temporal expression analysis has been actively studied through TempEval competitions [87, 88, 85].

Temporal expression analysis consists of two tasks: recognition and normalization. In the recognition task, temporal expressions in text are detected. In the normalization task, the meanings of temporal expressions are mapped to a grounded representation using the document creation time and context. For example, the temporal expression

"last Friday" is recognized from an input sentence "I wrote this paper last Friday," and it is normalized to 2018-12-07 by referring to the document creation time (DCT), 2018-12-10.

One of the characteristics of temporal expression is that the vocabulary is very limited. Many previous studies utilized temporal expression vocabularies and related combination rules. However, there are many loose structured temporal expressions in the wild. For example, "6th November" and "November 6" are sometimes written as "6, November." Furthermore, coordinate structures such as "November 6, 7" can be combined into a large number of syntactically varied yet semantically identical expressions. Previous approaches require various combination rules to cope with this problem.

We propose a neural network model that overcomes this issue. We only prepare a set of basic temporal interpretation rules for basic temporal expressions, called *time entities*, and we merely rely on the neural model to robustly compose these time entities to absorb the rich diversity of temporal expressions. Our experiments on the TempEval-3 dataset show that the proposed method achieves a new state-of-the-art in the temporal expression resolution task.

2.2 Related Work

Rule-based approaches have been widely used for both recognition and normalization tasks. Although they achieved high accuracy, it is difficult for them to consider a wider context. SUTime [17] recognized temporal expressions by applying three types of rules in order: (1) text regex rules which map simple regular expressions over characters or tokens to temporal representations, (2) compositional rules which are iteratively applied and compose temporal representations, (3) filtering rules which remove improbable representations. Then they normalized them using heuristic rules. HeidelTime [77] used regular expression patterns for recognition and temporal value resources for normalization. They strictly separated between the algorithmic part and the resources, so that the model can be easily adapted to a new language. Their system performed the best in TempEval-3 competition, both in English and Spanish. SynTime [102] focused on findings that the words included in temporal information are limited. They manually defined few hundreds of temporal candidate words, which were categorized into three groups:

time token, modifier, and numeral. They first recognized time tokens from text, and then found their surrounding words for modifiers and numerals and determined the boundaries of temporal expressions.

Machine learning approaches for recognition have been proposed. In the studies, morphosyntactic and lexicosemantic information features are designed. ManTIME [31] designed morphological features such as lemma, character, pattern, number and tense information, and applied a CRF classifier. ClearTK [9] used word, stem, POS, character and temporal type features, and applied SVM and logistic regression. TOMN [101] proposed a new tagging scheme, instead of the traditional BIO tagging scheme. They used four labels T,O,M,N: Time token, Modifier, Numeral, and the words Outside time expression. They achieved the state-of-the-art in recognition, but they did not tackle normalization. Our recognition model is inspired by their tagging scheme.

Recently, semantic parsing approaches have been proposed. Angeli et al. [3] introduced a latent parser for normalization. They defined a compositional grammar of temporal expressions which hardly rely on any specific language, and used EM-style bootstrapping approach to learn latent parses. They adapted a CRF model for recognition. Angeli and Uszkoreit [4] expanded the grammar and adapted it to multiple languages. UWTime [45] used a Combinatory Categorial Grammar (CCG) [75, 76] for both recognition and normalization and achieved the state-of-the-art performance. They designed a time representation inspired by Angeli et al. [3]. For example, a temporal expression "2nd Friday of July" is mapped to a meaning representation *intersect(nth(2,friday),july)*. Their CCG is defined by temporal tokens which are assigned to categories and combinators, and the grammar parses temporal expressions to trees. They first extract all the possible expressions which match the grammar, and then filter them by a classifier considering lexical, POS and context features. The recognized expressions are parsed by the grammar. Since there are some different possible derivations (e.g., There are many possible dates for "Friday"), and they select the best one using a learned model considering parse tree features and linguistic context features.

Recently, neural network approaches have achieved remarkable performance in many tasks, and many studies applied it to semantic parsing. Dong and Lapata [28] proposed sequence-to-sequence and sequence-to-tree models for semantic parsing converting natural language to logical form. Cheng et al. [20] proposed a transition-based approach to

generate tree structured logical forms. Rabinovich et al. [68] proposed abstract syntax networks for code generation and semantic parsing, and similarly, Yin and Neubig [97] proposed a model which uses syntax grammar as prior knowledge. Zhong et al. [100] proposed a neural network model to convert natural language questions to corresponding SQL queries by using policy based reinforcement learning. Wang et al. [92] introduced a recurrent neural network model to math word problems, and translated problem text to math equations.

2.3 Time Representation

We tackled the temporal expression resolution task based on TIMEX3 of TimeML standard [72]. As we mentioned in Chapter 1, temporal expressions have some attributes. Here, we detect temporal expressions and resolve their types and values. The possible types are *Date*, *Time*, *Duration*, and *Set*. *Date* expressions describe a calendar time (e.g., "January, 3", "2018"), *Time* expressions describe a time of the day (e.g., "evening", "twelve o'clock"), *Duration* expressions describe a time span (e.g., "four months"), and *Set* expressions describe a set of times (e.g., "twice a week", "every October"). The value is the normalized format of time based on the ISO-8601 standard, such as `1984-01-03T12:00` for "twelve o'clock January 3, 1984", and `P4M` for "four months."

We normalize temporal expressions based on the procedure of pointer network [90]. We define atomic temporal units named *time entities* and manually associate the basic temporal expressions with them. Even though a temporal expression is loose structured, our model generates the corresponding time entity sequence.

2.3.1 Time Entities

We define two types of time entities: *Time token* and *Function* (Table 2.1). Time tokens consist of three subclasses and Functions consist of six subclasses. Time entities are expressed with a typewriter font.

Time token Time tokens represent time units and their values. Time units represent granularity of time, and we defined the following 12 units: vague, century, year, quarter, season, month, week, day, weekday, hour, minute, and second. Vague unit

Table 2.1: List of Time Entities

Time Entity	Description
Time token	
DateTime (DT)	Date and time information, which is represented by a pair of time unit and value. e.g. "May" → {`DT[Month:5]`}
Period (P)	Duration information, which is represented by a pair of time unit and value. e.g. "two years" → {`P[Year:2]`}
Unit (U)	Temporal unit of time. e.g. "year" → {`U[Year]`}
Function	
Minus/Plus	Add/subtract time values. e.g. "two years ago" → {`Minus, P[Year:2]`}
FindEarlier/FindLater	Indicate the specific date before/after a reference date. e.g. "last May" → {`FindEarlier,DT[Month:5]`}
Cast	Change the granularity of time value to the succeeding time unit. e.g. "this year" → {`Cast, U[Year]`}
Frequency	Change the type of time to *Set*. e.g. "every May" → {`Frequency, DT[Month:5]`}

a special unit which represents vague granularity of past/present/future, and the possible values are `PAST_REF`, `PRESENT_REF`, `FUTURE_REF` which are defined in TIMEX3.

There are three subclass in the time tokens: DateTime, Period, and Unit.

DateTime (DT) DateTime represents the information of *Date* and *Time* types in TIMEX3. For example, "May," the fifth month, is represented as {`DT[Month:5]`}. A year "2018" is represented as {`DT[Year:2018]`}.

Period (P) Period represents the information of *Duration* type in TIMEX3. It also has a time unit and the corresponding value. For example, "two years" is represented as {`P[Year:2]`}.

Unit (U) Unit represents the unit of time, and it only has a time unit. For example, "year" is represented as {`U[Year]`}.

Function Functions operates on the time tokens. There are six subclasses in Function: Minus, Plus, FindEarlier, FindLater, Cast, and Frequency.

Minus/Plus Minus and Plus operate on Period by subtracting/adding time values. For example, "ago" is represented as {`Minus`}. "Two years ago" is represented as {`Minus, P[Year:2]`} using two time entities. It represents the year two years before the reference date. Similarly, "two years later" is represented as {`Plus, P[Year:2]`}.

FindEarlier/FindLater FindEarlier and FindLater operate on DateTime and Unit. It indicates the specific date before/after the reference. For example, "last" is represented as {`FindEarlier`} and "last May" as {`FindEarlier, DT[Month:5]`}. It represents May before the reference date. Similarly, "next May" is represented as {`FindLater, DT[Month:5]`}. Using a Unit, "last month" is represented as {`FindEarlier, U[Month]`}. It indicates the month just before the reference date.

Cast Cast changes the granularity of time values. It operates on DateTime and Unit. For example, "this" is represented as {`Cast`}. "This May" is represented as {`Cast, DT[Month:5]`} and it indicates the fifth month of the reference year. Similarly, "this month" is represented as {`Cast, U[Month]`}.

Frequency Frequency converts the time values of DateTime and Period into a set of times. It also converts the type of TIMEX3 to *Set*. For example, "every" and "each" are represented as {`Frequency`}. "Every May" is represented as {`Frequency, DT[Month:5]`} and "every two years" as {`Frequency, P[Year:2]`}.

We manually associate 200 basic temporal words/phrases with corresponding time entity sequences, as shown in Table 2.2. Some of the expressions are represented using

Table 2.2: Temporal words/phrases associated with time entity sequences.

Expression	Time entities
january	{DT[Month:1]}
friday	{DT[WeekDay:5]}
now	{DT[Vague:PRESENT_REF]}
([0-9]+) years?	{P[Year:#1]}
month	{U[Month]}
yesterday	{Minus, P[Day:1]}
ago	{Minus}
last	{FindEarlier}
each	{Frequency}
the	{Cast}

regular expressions. For example, an expression "([0-9]+) years?" in the table includes regular expressions, and "#1" in the corresponding time entities represents the first matching value of the regular expression. Each word/phrase is associated with one or more time entities. In addition, numeral words which are not matched the knowledge (e.g., "2018", "12") are assigned to following six time entities when a character # represents the numeral value: DT[Year:#], DT[Month:#], DT[Day:#], P[Year:#], P[Month:#], P[Day:#]. To reduce improbable candidates, we restrict the possible values of month and day: 1 to 12 for month and 1 to 31 for day.

2.3.2 Composition of Time Entities

In our model, temporal information is represented as a time entity sequence and an LSTM generates the sequence by selecting time entities (See section 2.4.2). The order of time entities has a meaning. The time entity sequence is combined and converted to a time value using the following two rules (Table 2.3).

1. Successive time tokens are combined. When adjacent temporal units have a descending order in terms of granularity, they are combined. For example, a sequence {DT[Year:2018], DT[Month:11]} is combined and converted to 2018-11.

Table 2.3: Composition of time entity sequences. Values with X requires the reference resolution.

Temporal expression	Corresponding time entity sequences	Converted value
“2019”	{DT[Year:2019]}	→ 2019
“4th March, 2019”	{DT[Year:2019], DT[Month:3], DT[Day:4]}	→ 2019-03-04
“4th March”	{DT[Month:3], DT[Day:4]}	→ XXXX-03-04
“4th and 5th March”	{DT[Month:3], DT[Day:4], DT[Day:5]}	→ XXXX-03-04, XXXX-03-05
“two years ago”	{Minus, P[Year:2]}	→ XXXX
“this year”	{Cast, U[Year]}	→ XXXX

Table 2.4: Examples of the reference resolution when 2018-12-10 is selected as the reference date.

Temporal expression	Time entity sequence		Converted value		Normalized value		
Reference Date Selection							
“two years ago”	{Minus, P[Year:2]}	→	XXXX	→	2016		
“last November”	{FindEarlier, DT[Month:11]}	→	XXXX-11	→	2018-11		
“this year”	{Cast, U[Year]}	→	XXXX	→	2018		
Reference Date + Relation Selection					Before	After	Nearest
“November”	{DT[Month:11]}	→	XXXX-11	→	2018-11	2019-11	2018-11

{DT[Month:11], DT[Day:6]} is converted to XXXX-11-06. When adjacent temporal units do not have a descending order, they make a coordinate structure. For example, {DT[Year:2018], DT[Month:11], DT[Month:12]} is converted to {2018-11, 2018-12}.

2. A function of time entities placed at the beginning of a sequence acts on the succeeding time entities. For example, {FindEarlier, DT[Month:5]} is converted to the value of May just before the reference date. When the reference date is unselected, the value is XXXX-05. When the reference date is 2018-12-10, it is normalized as 2018-05.

Note that some of the converted time values (i.e., values with X) require the reference resolution (See Section 2.4.2).

2.4 Neural Model

We propose a neural network model which recognizes and normalizes via multi-task learning. First, time entities are assigned to all the possible words/phrases by using our temporal phrase knowledge. Then, the model reads a sequence of words for each sentence with the bi-directional Long Short-Term Memory (BiLSTM). The recognition and normalization models share word embeddings and the hidden states of BiLSTM. The recognition model detects temporal expressions using a sequence labeling technique. The normalization model first selects the assigned time entities in an appropriate order, and then resolves the reference as necessary. Figure 2.1 shows the architecture of the model.

Words are represented as a concatenation of the following five vectors:

1. Word embedding. Since numeral words are sparse, they are converted into one of four values focusing on their values: numeral words whose digit is one or two, three, four, and the others.
2. POS (part-of-speech) embedding. Stanford CoreNLP [51] is used to obtain POS tag of each word.
3. The final hidden state of numeric character GRU (Gated Recurrent Unit) [24], which reads the numeric characters in a numeric expression.

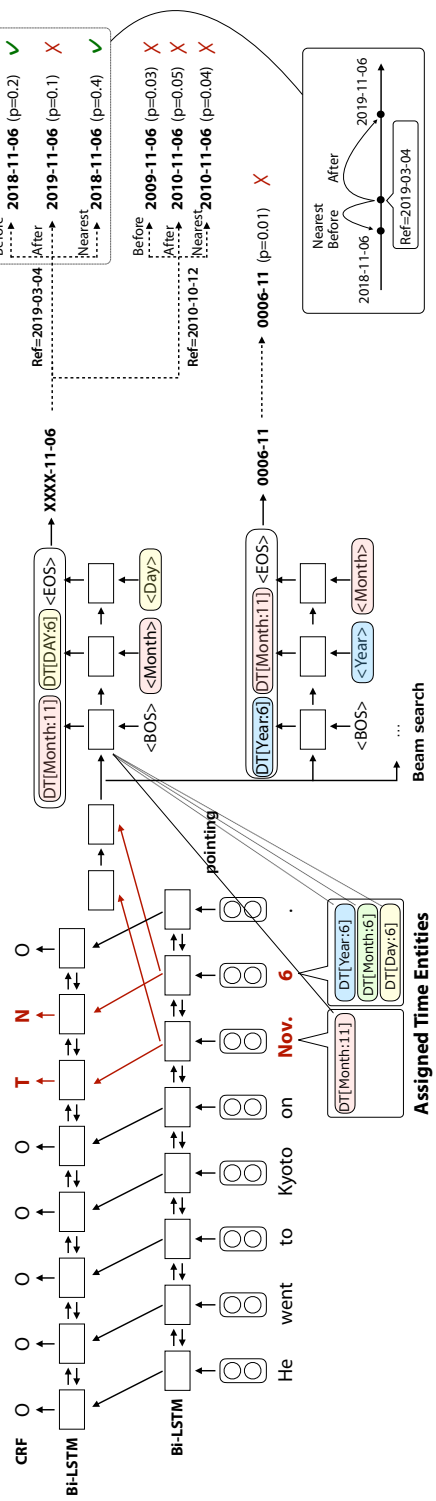

Figure 2.1: The architecture of recognition and normalization models.

4. Binary vector representing the digit and size of numeral: digit of numeral, whether it is ordinal or not, whether the size is less than 12, 30 and 2100.
5. Binary vector representing how it matches temporal phrase knowledge.

2.4.1 Recognition Model

The recognition task is modeled as a sequence labeling task using BiLSTM-CRF. We use a constituent-based tag scheme similar to TOMN [101]. Based on our temporal words/phrases knowledge, four labels N,T,F,O are used: *N*umeral words, words whose corresponding time entity is *Time token*, words whose corresponding time entity is *Function*, and words *O*utside time expression.

In order to recognize expressions with coordinate structures like "Nov. 6 and 7" as a single structure, specific linkage words (e.g., "and", "-") between temporal expressions are labeled with F.

2.4.2 Normalization Model

In the normalization task, the model first focuses only on the recognized temporal expression (i.e., tokens with non-O labels in recognition) and selects time entities in an appropriate order. If a reference is required for the composition, it is identified by using a context information.

The training data includes the resolved time values without time entity representations. Our model generates multiple time value candidates using a beam search, and the training is performed to increase the probabilities of candidates that match the correct time values. Let x denote the temporal expression, its time value y, and a time entity sequence candidate s, we maximize the probability $p(y|x)$.

$$p(y|x) = \sum_{s} p(s|x) \cdot p(y|s) \tag{2.1}$$

If a reference is not required, $p(y|s) = 1$.

Time Entity Selection

Time entities are selected by pointing to the assigned time entities, similar to pointer network [90]. The encoder reads the hidden states of recognized temporal expression,

and the decoder points a time entity. The model is expected to point the time entity in an appropriate order, while covering the temporal expression as much as possible. The model generates a list of candidate sequences via beam search under the following two restrictions: not select more than one time entity from one word, and not leave out words which assigned time tokens.

We use an attention vector as the pointing component [7]. At each time step t, the model selects the time entity k with the following probability:

$$p_k^t = \text{softmax}(v_c^T \tanh(\mathbf{W}_s s_t + \mathbf{W}_u u_i + \mathbf{W}_c c_k^t)) \tag{2.2}$$

where s_t denotes the hidden state of decoder, u_i denotes the input of decoder (i.e., a time granularity[1]), and c_k^t denotes the representation of time entity k. v_c, $\mathbf{W}_s$, $\mathbf{W}_u$ and $\mathbf{W}_c$ are trainable parameters. c_k^t, the representation of time entity k, is represented as the concatenation of corresponding word embedding, the time granularity embedding and attention score of the word:

$$c_k^t = \left[h_j, u_k, a_j^t\right] \tag{2.3}$$

where h_j denotes the representation of the j-th word in the input sentence which the time entity k is assigned to, u_k denotes the time granularity of time entity k, and a_j^t denotes the attention score of the j-th word. When the temporal expression consists of multiple words, h_j is represented as the sum of corresponding word representations. Using the idea of coverage mechanism proposed by Tu et al. [84], the attention score a_j^t is designed to remember the words previously attended and cover the temporal expression as much as possible:

$$a_j^t = \text{softmax}(v_h^T \tanh(\mathbf{W}_h h_j + \mathbf{W}_t s_t + \mathbf{W}_m m_j^{t-1})) \tag{2.4}$$

$$m_j^t = \sum_{t'=0}^{t} a_j^{t'} \tag{2.5}$$

where m_j^t denotes a coverage vector which is the sum of attention scores over all previous timesteps, and v_h, $\mathbf{W}_h$, $\mathbf{W}_t$ and $\mathbf{W}_m$ are trainable parameters.

[1] Since the specific value of time entity is not important to generate time entity sequence, time granularity is used for the input of decoder, which got better results than using the representation of time entity.

Reference Resolution

Time entity sequences which are converted with X require reference resolution. There are two types of the resolution. One requires only a reference date. The other requires a reference date and a relative relation, which is one of *before*, *after*, and *nearest* (Table 2.4).

1. The former is a case of sequences starting with Functions excluding Frequency entities. For example, a sequence {`Minus, P[Year:2]`} is converted to XXXX and normalized as 2017 when 2019-03-04 is selected as the reference date.

2. The latter is a case of sequences with omitted time unit. Here we consider a sequence {`DT[Month:11]`}, whose year information is omitted. It is first converted to XXXX-11, but even when 2018-12-10 is selected as the reference date, there are two possibilities of 2018-11 and 2019-11. Thus we consider three possibilities: *before* (2018-11), *after* (2019-11), and *nearest* (2018-11), and score each of them.

Three dates are used for the reference candidates: the document creation time (DCT) and the preceding two date expressions whose TIMEX3 types are *Date* or *Time*. Using the reference dates and the three relative relations, multiple value candidates are generated from a time entity sequence. A two-layer perceptron scores them using the following four features: reference information, value information, time entity sequence information, and context information.

1. Reference information:

   (a) Sentence distance from the reference expression. The distance from DCT is defined as zero.

   (b) Whether the reference is DCT or not.

   (c) The minimum granularity time unit of the reference. A binary vector corresponding to time unit is used. The corresponding value of minimum granularity time unit is one, and the other values are zero. For example, when the reference expression is “August,” the value corresponding to month is set one.

   (d) The final hidden state of LSTM which reads reference expression.

2. Value information of the recognized temporal expression:

(a) The minimum granularity time unit of the recognized temporal expression.

(b) Whether the minimum granularity time unit of temporal expression is same as that of the reference expression.

(c) TIMEX3 type of the recognized temporal expression.

(d) Number of days from the reference date to the normalized date of the recognized temporal expression.

(e) Whether the recognized temporal expression is same as the reference expression.

3. Time entity sequence information:

   (a) The final hidden state of LSTM in time entity selection.

4. Context information:

   (a) Self-attentive vector [46] for the context before and after the recognized temporal expression. The score of j-th word in the input sentence, note it as e_j, is computed as follows:

$$e_j = \text{softmax}(v_e^T \tanh(\mathbf{W}_m h_j)) \tag{2.6}$$

   where h_j denotes the representation of the j-th word, and v_e and $\mathbf{W}_m$ are trainable parameters. It is expected to weight on tense and conjunction, such as "will" and "after."

2.5 Experiments

2.5.1 Settings

We used the official training and testing dataset of TempEval-3. Since there are many mistakes in annotation, we used the corrected training data constructed by Lee et al. [45]. There are 256 and 20 documents in training and testing data, which include 1,822 and 138 temporal expressions respectively. 10% of training data was used for our development dataset.

Table 2.5: Experimental results in TempEval-3 dataset. The values of proposed method are the average of ten runs. The numbers in the parentheses represent the standard deviation.

	Recognition						Normalization				
	Strict Match			Relaxed Match			Type	Value			
	Pre.	*Rec.*	*F1*	*Pre.*	*Rec.*	*F1*	*F1*	*Acc.*	*Pre.*	*Rec.*	*F1*
TOMN [101]	**92.6**	**90.6**	**91.6**	95.6	**93.5**	**94.5**	-	-	-	-	
HeidelTime [77]	83.9	79.0	81.3	93.1	87.7	90.3	82.1	86.0	80.1	75.4	77.7
UWTime [45]	86.1	80.4	83.1	94.6	88.4	91.4	85.4	**90.2**	85.3	79.7	82.4
Full (3Refs+3Rels)	87.2	81.8	84.4 (2.9)	**96.2**	90.3	93.2 (0.8)	87.7 (1.3)	89.6	**86.2**	**80.9**	**83.4** (1.8)
DCT+3Rels	86.8	81.7	84.2 (2.8)	95.9	90.1	92.9 (0.9)	87.7 (1.4)	89.4	85.7	80.6	83.0 (1.4)
DCT+Nearest	88.1	83.2	85.6 (2.0)	**96.2**	90.9	93.5 (0.7)	**88.2** (1.0)	86.8	83.5	78.9	81.1 (1.0)
NoReferenceResolution	87.7	83.3	85.4 (2.5)	95.6	90.9	93.2 (1.0)	84.5 (0.8)	54.1	51.8	49.2	50.4 (0.6)

Word embeddings were initialized using pre-trained word embeddings[2] [62], whose dimension was 50. POS embeddings and numeral character embeddings were randomly initialized, and both of the dimensions were 10. The dimension of GRU hidden layer was 10.

The beam size in the time entity selection was 30. The dimensions of two-layer perceptron in the reference resolution were 300 and 100. Adam [40] was adopted as the optimizer. We conducted experiments with ten different initial parameters, and used the macro average.

For evaluation, We used the official evaluation tools in TempEval-3. The recognition results are evaluated in two metrics: strict match and relaxed match. Strict match metric only allows the exactly matching, while relaxed match allows the partial matching. When there is a relaxed match, its types and values are evaluated.

2.5.2 Results

Table 2.5 shows the experimental results. In the recognition task, the proposed model achieved 84.4 F1-score points in the strict metric, and 93.2 points in the relaxed metric. The scores outperformed that of HeidelTime, a rule-based model, and UWTime, a semantic parsing model. However, it did not reach the score of TOMN, which focused only on recognition, by 7.2 points in the strict metric and 1.3 points in the relaxed metric. In the normalization task, our model achieved 87.7 F1-score points in the type resolution and 83.4 points in the value resolution. The model surpassed UWTime, the state-of-the-art system, in Precision, Recall, and F1-score by 1.0 points. However, the standard deviation was high, 1.8 points in F1-score of value. We believe that the dataset is small for the neural network.

Three ablation studies are also shown in Table 2.5. In DCT+3Rels, we ablated the ability to select reference and always used DCT for the reference. The result shows that the scores dropped only a little. Since the dataset comes from newspaper domain, most of the narrative time is DCT. In DCT+Nearest, we furthermore ablated the ability to select the before/after reference. The F1-score of value dropped by 2.3 points. It shows that considering not only the nearest but other relations are essential. In NoReferenceResolution, we did not apply the reference resolution. For example, “May” is normalized as

[2]Downloaded from `https://nlp.stanford.edu/projects/glove/`.

XXXX-05 and "last year" is normalized as XXXX. The F1-score of value dropped by 30 points, which shows the effects of the reference resolution.

Errors in the recognition can be categorized into two groups: wrong detection of modifiers, and wrong detection of numerals. Wrong detection of modifiers affects the score of the strict metric. For example, the model wrongly detected "at least 72 hours" as "least 72 hours," and "1990" as "since 1990." Wrong detection of numerals affects both of the relaxed metric score and the strict metric score. The model wrongly detected "25" from "25 cents," and "three" from "three soldiers."

Most of the errors in the normalization due to reference resolution. For example, in the following sentence, the model wrongly normalized "Tuesday" as 1989-10-24. Since the sentence is long and there are several verbs between "will" and "Tuesday," the model could not correctly consider the tense.

(4) [DCT:1989-10-27]
But sources said he will be urging his allies to boost their stakes in Navigation Mixte, which is being traded in London and is to resume trading in Paris ***Tuesday***.
→ 1989-10-31 (gold), 1989-10-24 (sys)

2.6 Summary of this Chapter

In this chapter, we described a neural network model for recognizing and normalizing temporal expressions. Previous studies utilized temporal expression vocabularies and related combination rules. However, there are many loose structured temporal expressions in the wild and it is difficult to comprehensively prepare combination rules. For example, "6th November" and "November 6" are sometimes written as "6, November." To overcome the issue, we proposed a model which learns the composition of atomic temporal information. We only prepare a set of basic temporal interpretation rules for basic temporal expressions, namely time entities, and the neural network model robustly composes these time entities to absorb the rich diversity of temporal expressions. Our experimental results showed that In experiments, our proposed model surpassed the state-of-the-art system by 1.0 F-score points in the temporal resolution task of TempEval-3.

Chapter 3

Construction of Temporal Corpus

As described in Chapter 1, many previous corpora (e.g., TimeBank Corpus, TimeBank Dense Corpus) have been focused on intra-document relative temporal relations. Recently, annotation schemes and corpora which directly anchor events to the time axis have been constructed. The corpora have an advantage in anchoring events to the time axis accurately so that it is more appropriate for training and evaluating timelines. In this chapter, we designed a new annotation scheme that anchors expressions in text to the time axis comprehensively and constructed a temporal corpus using Japanese newspaper.

The rest of this chapter is organized as follows. In Section 3.1, we sort issues about temporal information annotation. In Section 3.2, we present related works about temporal corpus. In Section 3.3, we present the proposed annotation scheme. In Section 3.4, we present the statistics of the constructed corpus. In Section 3.5, we present the tag disagreements and discuss the difficulty of the proposed annotation scheme. In Section 3.6, we present conclusion of this chapter.

3.1 Temporal Information Annotation

There have been many studies and tasks to understand the relationship between event and temporal information in text. For example, temporal ordering of events that estimates the temporal relations of event-event and event-time was studied in TempEval 1, 2, 3 [87, 88, 85], and the timeline generation task that links event and time in multiple documents was studied in SemEval 15 [55].

In order to train models and evaluate results in these tasks, corpora in which event information is correlated with temporal information in text have been developed [64, 14, 70]. In these studies, expressions which have clear temporality were annotated, but in order to know how people understand texts from the perspective of time, it is essential to know how the expressions with weak temporality are interpreted. To understand temporal information in text exhaustively, we propose an annotation scheme that represents temporal information of various expressions in text, including expressions with ambiguous temporality.

The points of our annotation scheme are two-fold. One of the points is to annotate various expressions that can have temporality. We annotate not only expressions with strong temporality but also expressions with weak temporality. Many previous studies annotate "events" that express situations that happen or occur, which are defined in the guideline of TimeML [72]. Therefore, expressions as in the following example are not annotated.

(5) Businesses are ***emerging*** on the Internet so quickly that no one, including government regulators, can keep track of them.

However, the temporal information of expressions other than "event" also can be a clue to understand text. In the case of the above example, the temporal information of *"emerging,"* i.e., several years ago to the present, should be annotated to clarify the temporal common sense implied in the text. Therefore, we annotate all the expressions that can have temporality, that is, all the predicates and the eventive nouns in text. Annotators judge whether the expressions have temporality, and annotate the corresponding time tags.

The other point of our annotation scheme is that various types of time information such as frequency and duration can be anchored to the time axis. Reimers et al. [70] proposed an annotation scheme that represents an event period using its starting and ending points. However, it cannot represent "non-continuous period" or "a period in a long duration" as in the following examples.

(6) He ***plays*** baseball every Sunday.

(7) I will ***take*** a business trip for three days next week.

(8) He often used to ***have*** a tea with us.

In this chapter, we propose new time tags that can more accurately anchor various types of time information to the time axis.

By annotating various types of temporal information with the expressive time tags, personal interpretation of text and common sense appear as tag disagreements. In this research we consider that such disagreements are also important in understanding how time information is interpreted, and thus we do not eventually integrate time tags annotated by several annotators into one. Instead, we introduce an annotation method that keeps differences in interpretation and only corrects obvious annotation errors.

Using the annotation scheme, we annotated 113 documents with 4,534 expressions in Kyoto University Text Corpus. 76% of the expressions are judged to have temporality, and approximately 35% of them (26% of the total) are annotated with the notation newly proposed. Since the corpus has already been annotated with predicate-argument structures and coreference relations, our annotation makes it possible to utilize for integrated information analysis of events, entities and time.

3.2 Related Work

There are many corpora which associate event information with time information, and they can be roughly divided into two approaches. One approach is annotating temporal relations between events. Pustejovsky et al. [64] annotated events and times based on the TimeML guideline, and annotated relations between event-event, event-time, and event-time. Originally, the annotation was sparse because there were only the relations which are judged to be important by annotators, but TempEval competitions [87, 88, 85] annotated all the relations in same sentence to improve the coverage. Asahara et al. [5] applied TimeML guideline to Japanese language, and constructed BCCWJ-TimeBank Corpus which annotated event and temporal information on Balanced Corpus of Contemporary Written Japanese (BCCWJ) Corpus [49].

Some corpora densely annotated such temporal relations. Kolomiyets et al. [41] annotated temporal order relations with the nearest event expressions in a corpus of children's stories. TimeBank-Dense [14] used events and temopral expressions, and annotated all pairs of temporal relations in the same sentence and neighbouring sentences: (1)

event-event, event-time, and time-time pairs in the same sentence, (2) event-event, event-time, and time-time pairs in the neighbouring sentences, (3) all event-DCT pairs and (4) all time-DCT pairs.

The other approach is anchoring events to the time axis. Huang et al. [36] annotated one of five temporal status categories with events in newspaper articles on civil unrest: *Past, On-going, Future Planned, Future Alert,* and *Future Possible.* Asakura et al. [6] annotated three labels about time (PAST, PRESENT, FUTURE) and three values on facts (high probability, low probability, undescribed) for events relevant to flood disasters in the text posted on social media.

Reimers et al. [70] anchored with finer granularity. Their smallest granularity is day. They divided events into two types: single day event and multiple day event. The former is annotated with the date on which the event occurred, and the latter is annotated with the start and end dates of the event. For example, ***sent*** in the following sentence, an event which ends in one day, is annotated with *1980-05-26*, and ***spent***, an event spanning multiple days, is annotated with *beginPoint=1980-05-26 endPoint=1980-06-01.*

(9) He was ***sent*** into space on May 26, 1980. He ***spent*** six days aboard the Salyut 6 spacecraft.

In the case that the exact event date is not mentioned, notations *before* and *after* are used. In the following sentence, ***appointed*** is annotated with *after 1996-01-01 before 1996-12-31*, and ***part*** is annotated with *beginPoint=after 1984-10-01 before 1984-10-31 and endPoint=after 1984-10-01 before 1984-10-31.*

(10) In 1996 he was ***appointed*** military attache at the Hungarian embassy in Washington. [...] McBride was ***part*** of a seven-member crew aboard the Orbiter Challenger in October 1984.

They annotated events in TimeBank. In their annotation, about 60% of all the events ends in a day, and about 40% is events that span multiple days. 56% of the former has precise dates, and of the latter, 20% has precise start dates and 16% have precise end dates. 64% of the total is represented using *after* or *before*, and 1.6% does not have temporality.

In our work, we extend the anchoring to the time axis approach, and propose annotation scheme that can deal with various time information in text.

Table 3.1: List of time tags. Time tags with * are newly proposed in this work.

Temporality	Time Base Unit (TBU)	1. Date tag (Day, Month, Year) e.g. *t:1995-01-05* 2*. Vague time tag (Past, Present, Future) e.g. *t:PRESENT* 3. Interval tag (start∼end) e.g. *t:1995-01-05∼1995-01-07* 4*. Relative tag (Time Coreference) e.g. *t:*選挙の 5*. Utterance date tag (UD) e.g. *t:UD+P1D*
	Part of TBU	a*. Specific span in a TBU (span) e.g. *t:1995-01,span:P1W* b. Unspecific span in a TBU (span:part) e.g. *t:1995-01,span:part* c*. Repetition of TBUs (freq) e.g. *t:1995-01,freq:2/P1W*
No Temporality		e.g. *t:n/a*

3.3 Annotation Scheme

We annotate all the basic-phrases which consist of predicates and eventive nouns in text (hereinafter referred to as "target expressions"). Here, basic-phrase is defined as one independent word and successive attached words. By using basic-phrase as an atomic unit of annotation, verb phrases including postpositions and suffixes are regarded as one target expression. For example, phrases such as "勝つつもりだ (going to win)" and "勝てたかもしれない (might have won)" are regarded as target expressions, and time values are annotated. Although many previous studies annotate "events" which are defined in the guideline of TimeML, we annotate various expressions which may have temporality. Some examples of target expressions are shown below. "行くつもりだ (am planning to

go)" in Example (11) is a basic-phrase which includes verb, and "所属 (affiliation)" in Example (12) is a basic-phrase which is eventive noun, so that they are target expressions. In Example (13), there are two target expressions: "結婚を (getting marriage)" which is an eventive noun, and "考えたい (consider)" which is a basic-phrase including a verb.

(11) 明日京都に行くつもりだ。
(I ***am planning to go*** to Kyoto tomorrow.)

(12) 連合所属議員
(Representative of the Union ***affiliation***)

(13) そろそろ 結婚を 考えたい。
(It is about time to ***consider getting marriage***).

We first apply morphological analysis to text and extract base phrases of predicates and eventive nouns. Annotators first judge whether the target expressions have temporality. Expressions that are judged to have temporality are annotated with time tags which represent the corresponding time value in consideration of the document creation time (DCT) and the context. When an expression is judged to have no temporality, it is annotated with the time tag *not applicable (t:n/a)*.

A time tag that has temporality is represented as a Time Base Unit (TBU) or a combination thereof (Table 3.1). TBU represents a specific time point or period, and there are five types of tags. Furthermore, we introduced three ways of combining TBUs, which enable to represent various types of temporal information. Tags which Reimers et al. [70] used are 1, 3 and b in Table 3.1.

As in the previous studies, the finest granularity of time tags is day. This is because the granularity which is often attentioned in information analysis is days to years. For example, in the example below which is written in April 29, 2017, though a target expression "帰った (came)" happend at 18 o'clock of the day, the information of the granularity below the day is discarded and annotate the information of April 29, 2017.

(14) [DCT: 2017-04-29]
今日は夜 6 時に帰った。
(Today I ***came*** home at 6 pm.)

3.3.1 Judgement of Temporality

The presence of temporality is judged by whether the target expression implies a change in the behavior or state between the past and future. Since the presence of change and the variation depend on contexts, following examples are showed as a common recognition and individual judgement are entrusted to each annotator. Target expressions in the following examples have temporality.

(15) 明日京都に行く。
(I ***will go*** to Kyoto tomorrow.)

(16) 言語処理研究が盛んだ。
(Language processing research ***is thriving***.)

(17) あの子は背が低かった。
(The boy ***was short***.)

In Example (15), "行く (will go)" has temporality since it happens in a specific day tomorrow. Although the beginning and ending point of "盛んだ (thriving)" in Example (16) is vague, its limited range brings temporality. "低かった (was short)" in Example (17) implies that it is not the case now, so that it has temporality. In the examples above, "行く (will go)" in Example (15) is only an expression which is subject to annotate in the previous studies.

Next, we show some target expressions which do not have temporality.

(18) ウサギは草を食べる 動物だ。
(Rabbit ***is an animal*** that ***eats*** grass.)

(19) 彼の目は黒い。
(His eyes ***are black***.)

"食べる (eats)" and "動物だ (is an animal)" in Example (18) are common matters that do not change from long ago so that it has no temporality. The same is true on Example (19), though in the case of expressions suggesting that it is different now as in the following example, it is interpreted as having temporality.

(20) 以前は目の色が黒かった。
(His eyes ***were black*** in the past.)

3.3.2 Time Base Unit (TBU)

Date Tag

The temporal information of a date is represented by annotating the time value in *t* tag. The time value notation in Japanese TimeBank Corpus, BCCWJ-TimeBank [5], is used, such as *t:YYYY* and *t:YYYY-MM-DD*. For example, "到着した (arrived)" in the following sentence is annotated with *t:2017-04-28*.

(21) [DCT: 2017-04-29]
昨日大統領がニューヨークに**到着した**。
(The president ***arrived*** in New York yesterday.)
→ *t:2017-04-28*

Unlike the previous studies, our annotation scheme allows time tags with larger granularity than day. For example, "暑かった (was hot)" in the following example is annotated with *t:2016-08*. This tag does not necessarily mean exactly from August 1st to 31st. An expression "August" represents vaguer period than that of "August 1st to 31st." The granularity of the time tags in our research implies such vagueness.

(22) [DCT: 2017-04-29]
昨年の 8 月は**暑かった**。
(It ***was hot*** last August.)
→ *t:2016-08*

To reduce the annotation cost, we introduce the following shorthand notations:

- The date tag of the document creation time can be written as *t:DCT*.

  (23) [DCT: 2017-04-29]
  今日大統領がニューヨークに**到着した**。
  (The president ***arrived*** in New York today.)
  → *t:DCT*

- Annotators can represent the date before/after a certain period from a day by subtraction/addition. In this case, the time value notation of the period expression defined in BCCWJ-TimeBank is used. For example, 1 year is represented as *P1Y*,

1 month as *P1M*, 1 week as *P1W*, and 1 day as *P1D*. In the following examples, the time value of "行く (will go)" is represented as *t:DCT+P1W* and that of "行った (went)" is represented as *t:DCT–P1W*.

(24) 来週小樽に行く。
(I ***will go*** to Otaru next week.)
→ *t:DCT+P1W*

(25) 私も先週小樽に行った。
(I also ***went*** to Otaru last week.)
→ *t:DCT–P1W*

Vague Time Tag

There are many expressions that represent vague time in text. In the following sentence, it is not clear when and how long "住んでいた (used to live)" represents in the past.

(26) 昔広島に住んでいた。
(I used to ***live*** in Hiroshima.)

Reimers et al. [70] interpreted the temporal information of this expression as "a period from one day to another day until today," and annotated *beginPoint=before DCT endPoint=before DCT*. To represent these temporal information more accurately, we introduce some special tags.

Based on document creation time, the vague past, present and future are represented as *t:PAST*, *t:PRESENT* and *t:FUTURE*, respectively. *t:PRESENT* includes not only today but also a little past and future. In the following sentence, "持ち込める (can bring)" is annotated with *t:PRESENT* since it represents not only today but also a little before and after today.

(27) 国内線では飲み物を持ち込める。
(You ***can bring*** liquids on domestic flights.)
→ *t:PRESENT*

To represent the past and future, *t:PAST-M*, *t:PAST-Y*, *t:FUTURE-M* and *t:FUTURE-Y* tags are also available according to the temporal distance. *t:PAST-M* represents a few

months ago and *t:PAST-Y* represents a few years ago[1]. For more than a few years ago, or when the granularity is unknown, *t:PAST* is used. It is the same for future.

There is another vague time expression. In the case of expressions that represent numerical ambiguity, such as "around 1980" or "about 3 years," *ap* (approximately) is attached to the ambiguous numerical value of the time tag. In the following sentence, "建てられた (was built)" is annotated with *t:1980ap*.

(28) そのホテルは 1980 年頃に**建てられた**。
(The hotel ***was built*** around 1980.)
→ *t:1980ap*

Interval Tag

Time values of period are represented by connecting the starting point and the ending point with ∼. This notation corresponds to the *beginPoint* and *endPoint* tags in Reimers et al. [70]. "過ごした (spent)" in the following sentence is annotated with *t:1980-05-26∼1980-06-01*.

(29) 1980 年 5 月 26 日、彼は宇宙に向けて出発した。サリュート 6 号で 6 日間を**過ごした**。
(He was sent into space on May 26, 1980. He ***spent*** six days aboard the Salyut 6 spacecraft.)
→ *t:1980-05-26∼1980-06-01*

If either of the starting or ending point is near past/future and cannot be guessed, it is omitted. The time tag of "忙しかった (was busy)" in the following sentence is *t:∼2017-04-28*.

(30) [DCT: 2017-04-29]
昨日まで**忙しかった**。
(I ***was busy*** up until yesterday.)
→ *t:∼2017-04-28*

When the starting or ending point is far past/future, *PAST/FUTURE* is utilized.

[1]Period from current to several weeks ago is represented as *t:∼DCT* using a notation "∼."

Relative Tag

In texts with few temporal expressions, such as novels, it is difficult to anchor events to the time axis. In such a case, the TimeBank Corpus' annotation scheme, i.e., annotating the temporal relation between events, provides richer information. Therefore, in the case where the specific date is unknown but the temporal relation with another phrase in the same sentence is known, that phrase is used as a time value (*Time Coreference*). In the following sentence, though the date on which the demonstration took place is unknown, it can be understood that it is the day after the election. In this case, "起きた (was held)" is annotated with *t:*選挙の*+P1D*.

(31) 選挙の翌日、大規模なデモが**起きた**。
(The day after the election, a large demonstration ***was held***.)
→ *t:*選挙の*+P1D*

If there are two or more phrases that can be referred to, priority is given as follows and one with the highest priority is selected: 1. phrase with absolute time value tag, 2. phrase with the closest distance.

Utterance Date Tag

In conversational sentences and interviews, the date of the speech is often unknown. If the date of the utterance cannot be guessed from the context, the date can be described as *t:UD* (*Utterance Day*). In the following sentence, "頑張るしかない (work hard)" is annotated with *t:UD+P1D*. Note that "言った (said)" in the sentence, which is an expression outside the utterance, is annotated with the absolute time value.

(32) 「明日**頑張るしかない**」と監督は言った。
("I have no choice but to ***work hard*** from now," said the director.)
→ *t:UD+P1D*

3.3.3 Part of Time Base Unit (TBU)

Span in a TBU

A part of the period in a long TBU, e.g., a part of the period in August, is represented by combining the *t* tag representing the large period and the *span* tag representing the small

period. When the length of the small period is guessed, the *span* tag is represented using the notation of the duration expressions defined in the Japanese TimeBank Corpus. For example, three years is represented as *span:P3Y*, three weeks is represented as *span:P3W* and three days is represented as *span:P3D*. If the length of the small period cannot be guessed, it is represented as *span:part*. In the following sentence, "滞在した (stayed)" is annotated with *t:1984-10,span:part* since it happened sometime in October 1984, and "選ばれた (was honored)" is annotated with *t:2014,span:P1D* since it happened one day in 2014.

(33) サリバンは 1984 年 10 月、チャレンジャー号のメンバーとして宇宙に**滞在した**。2014 年にはタイム 100 に**選ばれた**。
(In October 1984, Sullivan ***stayed*** in space as a member of the Challenger. In 2014, she ***was honored*** in the Time 100 list.)
→ "滞在した (stayed)" *t:1984-10,span:part*
"選ばれた (was honored)" *t:2014,span:P1D*

The *span:part* tag is equivalent to the *before* and *after* tags in Reimers et al. [70].

Repetition of TBU

There are many target expressions that are not represented as continuous periods, such as "every Sunday" and "once every three days." Target expressions occurring across multiple days repeatedly are represented with *freq* tag, in addition to the *t* tag and the *span* tag.

The *freq* tag is used in three ways.

1. When the repetition is expressed as a number of occurrences during a certain period, such as "twice a week" and "once every three days," the *freq* tag is represented as *the number of times / period*. In the following sentence, "通っている (go)" is annotated with *t:2016-07∼DCT,freq:2/P1W*.

(34) [DCT: 2017-04-29]
2016 年 7 月から週に 2 回プールに**通っている**。
(I ***go*** to the pool twice in a week since July 2016.)
→ *t:2016-07∼DCT,freq:2/P1W*

2. When the repetition is expressed as a repetition of specific date, such as "every 25th day" and "every Sunday," the date is used as a value of the *freq* tag. The Japanese TimeBank Corpus' notation is extended by allowing to include the symbol @ in each part of *YYYY-MM-DD* in the sense that it can represent any number. In Example (35), "開催される (is held)" is annotated with *t:PRESENT,freq:@@@@-@@-25*, and "行く (go)" in Example (36) is annotated with *t:PRESENT,freq:@@@@-@@-Sun*.

   (35) 骨董市は毎月 25 日に**開催される**。
   (The antique market ***is held*** on the 25th of every month.)
   → *t:PRESENT,freq:@@@@-@@-25*

   (36) 毎週日曜はプールに**行く**。
   (I ***go*** to the pool every Sunday.)
   → *t:PRESENT,freq:@@@@-@@-Sun*

3. When the repetition or the frequency cannot be guessed from the context, one of the following four abstract tags is used: *usually, often, sometimes* and *rarely*. In the following sentence, "行く (go)" is annotated with *t:PRESENT,freq:sometimes*.

   (37) [DCT: 2017-04-29]
   スターバックスに時々**行く**。
   (I sometimes ***go*** to Starbucks.)
   → *t:PRESENT,freq:sometimes*

3.4 Annotation Study

3.4.1 Annotation Method

Using our annotation scheme, we annotated a subset of documents in Kyoto University Text Corpus [38]. The corpus consists of approximately 40,000 sentences from Mainichi newspaper in 1995 with various linguistic information, such as predicate-argument structures and coreference relations. Eleven hot topics are listed, and the related 113 documents are selected (Table 3.2). The subset consists of 856 sentences including 4,534

Table 3.2: Distribution of the number of annotated articles.

	Document Creation Time (In 1995)						
Topic	Jan. 1	Jan. 3	Jan. 4	Jan. 5	Jan. 6	Jan. 7	Total
Trends in Chechnya	1	3	5	7	7	4	27
The new party	2	0	0	4	4	7	17
Trends in US	2	1	2	2	3	2	12
Prime Minister's movement	1	2	4	3	1	0	11
America's Cup (yacht race)	10	0	0	0	0	0	10
Trends in China	4	0	1	4	0	0	9
Development of Mekong River	7	0	0	0	0	0	7
Avalanche in Nagano	0	0	0	4	2	0	6
Kansai International Airport	1	0	0	4	1	0	6
Movement in Russia	1	1	1	0	1	1	5
Mexican economy	0	0	2	1	0	0	3

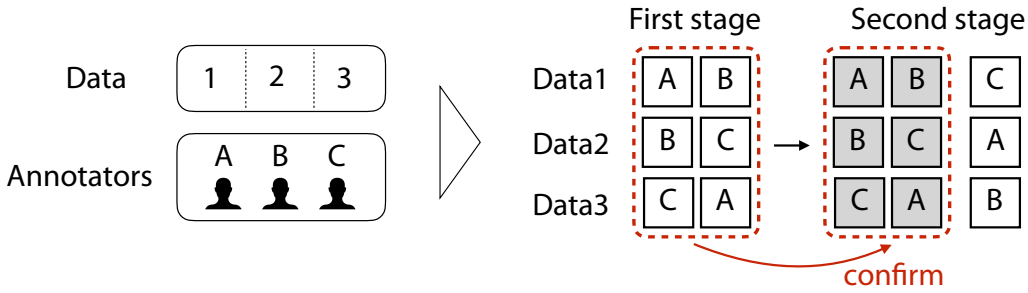

Figure 3.1: The annotation method by three annotators.

target expressions. In the 4,534 target expressions, there are 3,072 predicates and 1,462 eventive nouns.

3.4.2 Annotation method

The time tags were annotated by three annotators. Since we annotate expressions whose interpretation varies depending on the individual's common sense, we do not eventually combine the annotators' tags into one. We introduce a two-step annotation method that keeps the interpretation of other annotators and modifies only obvious annotation errors.

The document set is divided into three parts. Each annotator annotates two of them in the first step, and the remaining one is annotated in the second step. Figure 3.1 shows the method. In the first step, each annotator independently annotates, and in the second step they annotate tags by confirming the others' tags in the first step. If an obvious error is found in the already annotated tags, it is just marked. The marked tags are 2% of the total and are treated as missing values in the analysis in section 3.5

3.4.3 Distribution of the Annotated Time Tags

The distribution of the annotated time tags is shown in Table 3.3. Approximately 75% of the target expressions have temporality. While around 25% of the annotated tags are the date tags and 15% are the interval tags, the vague time tags account for 10% and few percent are the relative tags. Since the domain of annotation is newspaper, the majority of target expressions are directly anchored to the time axis. The *freq* tag, representing repetition, is hardly used, i.e., 1% of the whole. The time tags that are newly proposed in this study account for 25% of the whole.

Table 3.4 shows the distribution of time tags annotated with predicates and eventive nouns in the second stage. While the date tags account for 27% of predicates, they are 12% in eventive nouns. On the other hand, many of eventive nouns are judged not to have temporality, and they account for 35%, which is about twice the predicates. This is because the eventive nouns often represent organizations and general events, such as "全欧安保***協力***機構 (Organization for Security and ***Co-operation*** in Europe)" and "地方旅行の自由化 (liberalization of regional ***travel***)."

3.4.4 Inter-Annotator Agreement

We compute the inter-annotator agreement using Krippendorff's α [42, 33]. Following Reimers et al. [70], two metrics are utilized. One is a strict metric that measures whether the time tags completely match. For example, while *t:1994-12-31* matches *t:1994-12-31*, it does not match *t:∼1994-12-31*. The other is a relaxed metric that permits partial matching. If the time tags are overlapped event for one day, they are regarded as matched, and if they do not overlap at all, they are regarded as mismatched. For example, *t:1994-12-31* and *t:∼1994-12-31* partially overlap their span, *t:∼1994-12-31* and *t:1995-01-01*

Table 3.3: Distribution of all the annotated time tags. Indented items represent a breakdown. Time tags with * are newly proposed.

	Annotator1	Annotator2	Annotator3	Average
1. Date tag	1,195 (26.4%)	1,145 (25.3%)	938 (20.7%)	1,093 (24.1%)
Year	35 (0.8%)	47 (1.0%)	16 (0.4%)	33 (0.7%)
Month	9 (0.2%)	14 (0.3%)	3 (0.1%)	9 (0.2%)
Day	1,151 (25.4%)	1,084 (23.9%)	919 (20.3%)	1,051 (23.2%)
2*. Vague time tag	617 (13.6%)	375 (8.3%)	249 (5.5%)	414 (9.1%)
t:PRESENT	520 (11.5%)	257 (5.7%)	195 (4.3%)	324 (7.2%)
t:PAST	40 (0.9%)	25 (0.6%)	17 (0.4%)	27 (0.6%)
t:FUTURE	57 (1.3%)	89 (2.0%)	31 (0.7%)	59 (1.3%)
t:ap	0 (0.0%)	4 (0.1%)	6 (0.1%)	3 (0.1%)
3. Interval tag (∼)	387 (8.5%)	562 (12.4%)	842 (18.6%)	597 (13.2%)
4*. Relative tag (Time Coreference)	138 (3.0%)	58 (1.3%)	207 (4.6%)	134 (3.0%)
5*. Utterance date tag(*t:UD*)	77 (1.7%)	77 (1.7%)	106 (2.3%)	87 (1.9%)
a*. Specific span in a TBU (span)	540 (11.9%)	447 (9.9%)	550 (12.1%)	512 (11.3%)
Date tag + span	46 (1.0%)	69 (1.5%)	96 (2.1%)	70 (1.6%)
∼ + span	482 (10.6%)	357 (7.9%)	434 (9.6%)	424 (9.4%)
Vague time tag + span	12 (0.3%)	21 (0.5%)	20 (0.4%)	18 (0.4%)
b. Unspecific span in a TBU (span:part)	455 (10.0%)	561 (12.4%)	478 (10.5%)	498 (11.0%)
Date tag + span:part	36 (0.8%)	56 (1.2%)	46 (1.0%)	46 (1.0%)
∼ + span:part	373 (8.2%)	475 (10.5%)	391 (8.6%)	413 (9.1%)
Vague time tag + span:part	46 (1.0%)	30 (0.7%)	41 (0.9%)	39 (0.9%)
c*. Repetition of TBUs (freq)	46 (1.0%)	52 (1.2%)	47 (1.0%)	48 (1.1%)
No Temporality (*t:n/a*)	1,071 (23.6%)	1,077 (23.8%)	1,060 (23.4%)	1,069 (23.6%)
Tags marked in the second stage	8 (0.2%)	180 (4.0%)	57 (1.3%)	82 (1.8%)
All	4,534	4,534	4,534	4,534
Newly proposed tags (*)	1,418 (31.3%)	1,009 (22.3%)	1,159 (25.6%)	1,195 (26.4%)

Table 3.4: Distribution of time tags annotated in the second step.

	Predicate	Eventive noun	All
1. Date tag	844 (27%)	172 (12%)	1,016 (22%)
2*. Vague time tag	272 (9%)	105 (7%)	377 (8%)
3. Interval tag	460 (15%)	210 (14%)	670 (15%)
4*. Relative tag (Time Coreference)	120 (4%)	42 (3%)	162 (4%)
5*. Utterance date tag	68 (2%)	8 (1%)	76 (2%)
a*. Specific span in a TBU (span)	380 (12%)	173 (12%)	553 (12%)
b. Unspecific span in a TBU (span:part)	311 (10%)	216 (15%)	527 (12%)
c*. Repetition of TBUs (freq)	39 (1%)	18 (1%)	57 (1%)
No Temporality	578 (19%)	518 (35%)	1,096 (24%)
All	3,072	1,462	4,534

Table 3.5: Inter-annotator agreement computed by Krippendorff's α. The values in parentheses indicate the agreement in (predicates / eventive nouns).

	Strict	Relax
The first step	0.417 (0.439/0.353)	0.719 (0.743/0.659)
The final result	0.554 (0.571/0.506)	0.802 (0.820/0.756)
The first step (Excluding *t:n/a*)	0.380 (0.410/0.293)	0.803 (0.803/0.798)
The final result (Excluding *t:n/a*)	0.526 (0.548/0.461)	0.867 (0.867/0.865)
[Reimers+ 16]	0.617	0.912

are mutually exclusive. In *t:n/a* and relative tags, partial matching is not defined and only the strict matching is applied.

One of the characteristics of Krippendorff's α is that it can be applied to missing data. Therefore, it can be applied to not only the final result but the first step annotation, that each expression is annotated by two out of three annotators. Table 3.5 shows the agreement at each step. The agreements of predicates and eventive nouns are represented in parentheses in the table. "Excluding *t:n/a*" means an agreement computed excluding

the expressions in which one or more annotators annotated with *t:n/a* (Around 1,300 expressions both in the first stage and the final results). The agreement of two annotators is computed in the first stage, and that of three annotators computed in the final result.

Comparing the first step and the final result of the annotation process, the latter agreement increased significantly. This is because while the documents are annotated independently in the first step, annotators can check others' tags in the second step. When the target expressions annotated with *t:n/a* are excluded, the strict agreement increased significantly. It shows that the difficulty of temporality judgement is a cause of lowering the agreement in the relaxed metric.

Comparing the agreement of predicates and eventive nouns, the former is higher. When the target expressions annotated with *t:n/a* are excluded, the former is higher in the strict metric while both are almost same in the relaxed metric. Since eventive nouns contain non-temporality expressions much more than predicates, it is difficult to judge their temporality. Furthermore, the agreement of eventive nouns is still low after the tags with *t:n/a* are excluded. It suggests that expressions with clear temporal information are few. However, note that the agreement of eventive nouns is as high as that of predicate in the relaxed metric. Although the tags do not match exactly, there are not much differences between the annotators' interpretations.

Compared with Reimers et al. [70], the agreement in the strict metric is particularly low. Due to the increase of the variation of the time tags, annotators' interpretations can be reflected a lot, and it became difficult to agree completely.

The tag disagreements in both the metrics are discussed in the following section.

3.5 Disagreement Analysis

While the proposed time tags can more accurately represent the temporal information than the previous research, it is more sensitive to the interpretation of annotators. In this section, we analyze how the time tag disagreed between annotators.

In order to analyze the annotated time tags without being limited to specific values, we abstract them from the aspect of granularity. For example, for the *t* tag, *t:1994-12-31* is abstracted as *DAY*, *t:∼1994-12-31* is abstracted as *∼DAY* and *t:1994* is abstracted as *YEAR*. For the *span* tag and the *freq* tag, their values are omitted. For example, *t:∼1994-*

Table 3.6: Frequency of agreed/disagreed time tags in the first step in the *Strict* metric

Agreed between annotators			Disagreed between annotators		
Pair of abstracted time tags		Frequency	Pair of abstracted time tags		Frequency
n/a	n/a	800	PRESENT	n/a	110
DAY	DAY	740	DAY	n/a	105
∼DAY,span	∼DAY,span	142	DAY	∼DAY,span	104
PRESENT	PRESENT	113	∼DAY,span	∼DAY,span	77
DAY∼,span	DAY∼,span	54	DAY	∼DAY	73
DAY∼DAY	DAY∼DAY	38	PRESENT	∼DAY,span	59
YEAR	YEAR	12	PRESENT	∼DAY	49
DAY∼FUTURE	DAY∼FUTURE	10	PRESENT	PAST∼DAY	49
All		2,045	All		2,275

Table 3.7: Frequency of agreed/disagreed time tags in the first step in the *Relaxed* metric

Agreed between annotators			Disagreed between annotators		
Pair of abstracted time tags		Frequency	Pair of abstracted time tags		Frequency
n/a	n/a	800	PRESENT	n/a	110
DAY	DAY	741	DAY	n/a	105
∼DAY,span	∼DAY,span	219	DAY	DAY	44
PRESENT	PRESENT	113	DAY∼,span	n/a	36
∼DAY,span	DAY	90	∼DAY,span	n/a	31
DAY∼,span	DAY∼,span	85	DAY∼FUTURE	n/a	20
DAY	∼DAY	67	DAY∼FUTURE,span	n/a	19
∼DAY,span	PRESENT	59	DAY∼DAY	n/a	14
All		3,533	All		787

12-31,span:P1D and *t:∼1994-12-31,span:part* are both abstracted as *∼DAY,span*.

In this section, we analyze the results of the first stage, where annotators independently annotated. Tables 3.6 and 3.7 show disagreements in the strict and relaxed metrics respectively. Here, the agreement is computed using the original time tags, and only the

statistics is calculated using abstracted values.

Table 3.6 indicates that in the strict metric, about 70% of agreed tags are *DAY* and *n/a*, and most of the disagreements are the judgement of temporality and the interpretation of date and period such as *DAY* and ∼*DAY*. Table 3.7 indicates that most of the disagreements in the relaxed metric are the judgement of temporality. It indicates that most of the tags that were disagreed due to the interpretation between date and period in the strict metric overlap the spans, and they are consistent in the relaxed metric.

In the following subsection, we analyze the disagreement of temporality judgment and the disagreement of interpretation of the date and the period with actual examples.

3.5.1 Judgement of Temporality

In the relaxed metric, the biggest cause of disagreements is that the judgement of temporality varies depending on annotators. When one annotator tags *n/a*, the other annotates *n/a* (76.6%), *DAY* (5.3%), *PRESENT* (5.0%), ∼*DAY,span* (1.7%) in order of frequency. This means that 75% of *n/a* tags agree, and if it is not the case, one annotates the *DAY* or *PRESENT* tag at a rate of 40%. Many of these expressions represent states, positions and organizations, and the judgment is divided according to whether it is interpreted as permanent or as a temporal period.

In the following sentence, one annotated *t:PRESENT* and the other annotated *t:n/a*.

(38) 大統領官邸の**ある**中心部
(The city center where the presidential official residence ***exists***)

The annotator who recognized temporality interpreted that there is a possibility that the place of the presidential office may change in the future, while the other interpreted it as semi-permanent.

3.5.2 Interpretation of Date and Period

As Reimers et al. [70] pointed out, it is difficult to judge whether an event ends in one day or is held for several days from a text. It is also not easy to clarify the beginning and ending date of an event. Such vagueness appears as disagreements among *DAY*, ∼*DAY*, ∼*DAY,span*, *DAY*∼, *DAY*∼*,span* and *PRESENT* in this annotation scheme. Among them, the disagreement between *DAY* and ∼*DAY,span* often occurs. In many cases, *DAY*

is DCT, which means that it is difficult to interpret whether it occurred at the written date or before that.

In the following sentence, it is difficult to judge the duration of the event ***resists*** from the text. One annotated *t:DCT* and the other annotated *t:∼DCT,span:part*. While the former interpreted that the event occurred in a day, the latter interpreted as a longer period.

(39) しかしドゥダエフ政権部隊は頑強に**抵抗**、双方の死者は数百人に達する見込みだ。
(But the Dudaev regime strongly ***resists***, and the death toll will reach hundreds.)

One of the difficulties is due to the domain being newspaper. In the following sentence, one annotated *t:DCT* and the other annotated *t:∼DCT,span:P1D*. While the former interpreted that it happened on the date when the article was written from the promptness of newspaper, the latter interpreted that it was not necessarily so.

(40) 外相は、「非民営化・再国営化」の基本方針を**打ち出した**。
(The Foreign Minister ***has laid out*** the basic policy of "non-privatization and re-nationalization.")

Thus, the major cause of the disagreements among the annotators is that there are multiple interpretations depending on the context and common sense, closely related to the writing style and theme of newspaper.

3.6 Summary of this Chapter

In this chapter, we described a new annotation scheme for anchoring expressions in text to the time axis comprehensively. The points of our annotation scheme are two-fold: annotating various expressions that can have temporality, and annotating various types of time information such as frequency and duration can be anchored to the time axis. Using this scheme, we annotated a subset of a Japanese newspaper corpus, and the new tags account for approximately 25% of all tags. The Krippendorff's α inter-annotator agreement was 0.55 in strict metric, and 0.80 in relaxed metric. Since our annotation scheme is sensitive to the interpretation of annotators, it became difficult to agree completely. The corpus has already been annotated with various linguistic information so that our

annotation makes it possible to utilize it for integrated temporal information analysis of events, entities and time.

Chapter 4

Event Analysis and Timeline Construction

The core structure of the storyline is anchoring events to the time axis, namely, timeline. The fundamental technique to generate a timeline is understanding the relations between events and temporal expressions. Furthermore, since the number of temporal expressions in an article is small, interpreting the temporal nature of events and temporal relations between two events is also important for timeline construction.

In this chapter, we describe our works on temporal information analysis of events. We first determine two kinds of relations between events: temporal ordering relation (before-after) and subevent relation (parent-child). We then design three multi-class classification tasks to anchor every event to the time axis directly. Finally, we construct timelines under the framework of the *TimeLine task* in SemEval 2015. The task focuses on events which have clear temporality and consists of two subtasks: extracting events related to a topic and anchoring those events to the appropriate time. We propose a timeline generation model which considers relations between events and external knowledge.

In Section 4.1, we present a model which determines temporal and subevent relations between events. In Section 4.2, we present three multi-class classification tasks which anchor events to the time axis. In Section 4.3, we propose a timeline generation model. In Section 4.4, we discuss our temporal information analysis of events and the remained problems. In Section 4.5, we present a conclusion of this chapter.

4.1 Event-Event Ordering

Newspaper articles contain more temporal expressions compared to texts of other domains, but their amount is not large. In this section, we propose a temporal relation prediction model, which provide a clue from a different perspective to understand temporal information of text. While many previous works focus on ordering all the events, it is important to order the topic related events for timeline construction.

We tackle a task to detect temporal relations of events focusing on the chronological order of events that occur in a script, which is proposed in the *Event Sequence Detection Task* in TAC2017 (Text Analysis Conference). In this task, eight types of events defined in DEFT Rich ERE Event Annotation Guidelines[1] are considered. Two kinds of relations are defined between events: *subevent* link (parent-child) and *after* link (before-after)[2]. A subevent link represents parent-child relations, which is a stereotypical sequence of events that occur as part of a whole event. After links are added between child events in a script when their chronological order is clearly mentioned or predicted by common knowledge of the script. In the example in Figure 4.1, there are subevent links in "*attacked* $\rightarrow$ *hit*" and "*attacked* $\rightarrow$ *stabbed*," and an after link in "*hit* $\rightarrow$ *stabbed*."

There are several approaches to estimate a temporal relation between events. One is a feature based machine learning approach, which utilizes hand-crafted rules, event attributes and external resources [29, 15]. Another is a neural network based approach, which performs comparable without using hand-crafted features or external knowledge [18, 23]. We take a neural network based approach for the event sequence classification with external knowledge about events.

Among the combinations of all events, only a small portion of the relations have a temporal relation, mostly *NONE*. In order to eliminate this class imbalance, an undersampling technique is used. Our system achieved F-score of 12.6 for the official evaluation, which ranked first among two teams.

[1] https://tac.nist.gov/2016/KBP/guidelines/summary_rich_ere_v4.2.pdf

[2] http://cairo.lti.cs.cmu.edu/kbp/2017/event/TAC_KBP_2017_Event_Coreference_and_Sequence_Annotation_Guidelines_v1.1.pdf

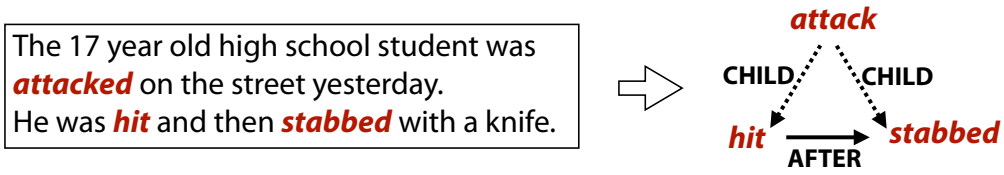

Figure 4.1: An example of event-event ordering task. Two relations between events, subevent relation (parent-child) and after relation, are predicted.

4.1.1 Related Work

There have been many studies on temporal relation classification, which estimates relations between event-event pairs or event-time pairs. TimeBank Corpus [64] and TempEval competitions [87, 88, 85] have contributed to the development of classification techniques.

Feature based approaches use hand-crafted rules, event attributes and external resources such as WordNet [54] and VerbOcean [22]. Mani et al. [50] built a Maximum Entropy classifier using annotated features in a corpus and outperformed rule-based approaches. Chambers and Jurafsky [16] focused on the constraints of temporal event ordering, such as `X before Y` and `Y before Z` implies `X before Z`. They first applied a pairwise classifier between events, and then Integer Linear Programming fixed them considering a global constraint. Yoshikawa et al. [98] expanded the model more global. They additionally predict temporal ordering between events and temporal expressions, and between events and the document creation time. Furthermore, they used the Markov Logic Network [71] to capture non-deterministic constraints. D'Souza and Ng [29] combined rule-based and data-based approaches. They first applied rules. If none of the rules was applicable, a classifier was used. In the classifier, lexical relation, semantic and discourse features were used. Specifically, in addition to WordNet and VerbOcean, predicate-argument relations and Penn Discourse TreeBank (PDTB) style [63] discourse relations were used. Chambers et al. [15] introduced a sieve-based architecture for event ordering. 12 temporal relations classifiers were applied in sequence and gradually labeled edges of a graph of events and temporal expressions. Begin with the most reliable classifier, each classifier adds edges which satisfy transitivity constraints.

Neural network based approaches perform comparably without using hand-crafted

efforts or external resources. Since the dependency path based neural network methods perform well in relation extraction tasks [73, 93, 94], the techniques are introduced to the temporal relation classification. Choubey and Huang [23] proposed a BiLSTM model to classify intra-sentence events. They generate three sequences of dependency path: the word sequence, the POS tag sequence, and the dependency relation sequence. They apply BiLSTMs for each sequence and concatenate the outputs to estimate the relationship. Cheng and Miyao [18] applied BiLSTM to dependency paths and estimated cross-sentence relationships. To estimate the relationship between two entities, they make two sequences, each entity to the common root of the entities, and apply BiLSTMs to them. For each sequence, the concatenation of word, POS, and dependency relation embeddings is used.

4.1.2 Model

The input of the system is the (gold) event pairs e_1 and e_2 (e_2 appears after e_1 in a document). The annotated directed links are normalized to an event sequence class, which is a relation from e_2 to e_1, for ease of the direction handling. The output of the system is a sequence class, which includes *BEFORE*, *AFTER*, *PARENT*, *CHILD*, and *NONE*. Figure 4.2 shows the architecture of the system.

Network Architecture

Let $\boldsymbol{x}_i$ be the embedding corresponding to the i-th word, which is represented as a concatenation of word embedding and POS (part-of-speech) embedding. First, to obtain the contextual word representation, bi-directional GRU (Gated Recurrent Unit) [24] is applied to a sequence of words for each sentence as follows:

$$\overrightarrow{\boldsymbol{h}}_i = \overrightarrow{GRU}(\boldsymbol{x}_i, \overrightarrow{\boldsymbol{h}}_{i-1}), \tag{4.1}$$

$$\overleftarrow{\boldsymbol{h}}_i = \overleftarrow{GRU}(\boldsymbol{x}_i, \overleftarrow{\boldsymbol{h}}_{i+1}), \tag{4.2}$$

and the representation for the i-th word is a concatenation of these hidden states as follows:

$$\boldsymbol{h}_i = [\overrightarrow{\boldsymbol{h}}_i; \overleftarrow{\boldsymbol{h}}_i]. \tag{4.3}$$

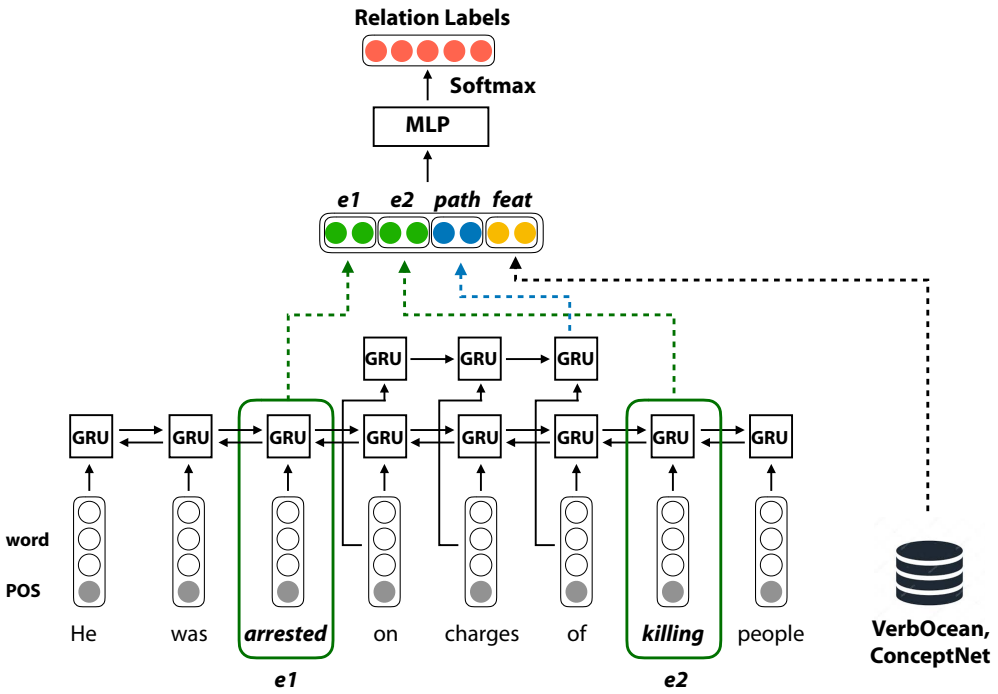

Figure 4.2: The system architecture of temporal relation task.

The input vector $\boldsymbol{v}_{in}$ for the classification is a concatenation of $\boldsymbol{v}_{e_1}$ and $\boldsymbol{v}_{e_2}$ (the representations of e_1 and e_2), a path embedding $\boldsymbol{v}_p$ and a feature vector $\boldsymbol{v}_f$ of e_1 and e_2. A word sequence between e_1 and e_2 can be a clue for the classification. GRU reads the word sequence, and the final hidden layer is adopted as the path embedding. The feature vector includes the followings:

- Event subtype of e_1 and e_2

  The events in this task are based on the definition in DEFT Rich ERE Event Annotation Guidelines and type and subtype are annotated for each event. There are 8 types, such as *Business* and *Conflict*, and 38 subtypes, such as *Declare–Bankrupt* and *Attack*. The (gold) event subtypes of e_1 and e_2 are utilized.

- Realis of e_1 and e_2

  The (gold) realis status (ACTUAL, GENERIC and OTHER) of e_1 and e_2 is used.

- Sentence distance between e_1 and e_2

  A binary vector of sentence distance between e_1 and e_2 is used.

- Exact match of lemmas between e_1 and e_2

- Existence of a semantic relation between e_1 and e_2 in external knowledge

  The semantic relation of event-pair obtained from external knowledge is used. The details are described in Section 4.1.2.

The input vector $\boldsymbol{v}_{in} \in \mathbb{R}^{d_{in}}$ (d_{in} denotes the dimension of the input vector) is fed into a Multi-layer perceptron (MLP). A hidden state $\boldsymbol{h}_c$ (for the classification) is calculated as follows:

$$\boldsymbol{h}_c = f(W_1 \boldsymbol{v}_{in}) \tag{4.4}$$

where $W_1 \in \mathbb{R}^{d_{h_c} \times d_{in}}$ (d_{h_c} denotes the dimension of the hidden layer) is a weight matrix from the input layer to the hidden layer, and f is an activation function ($tanh$ is used in our experiments). The predicted probability distribution $\boldsymbol{y}$ is calculated as follows:

$$\boldsymbol{y} = \mathrm{softmax}(W_2 \boldsymbol{h}_c) \tag{4.5}$$

where $W_2 \in \mathbb{R}^{d_{out} \times d_{h_c}}$ (d_{out} denotes the number of event class) is a weight matrix from the hidden layer to the output layer. The objective is to minimize the cross entropy between predicted and true distributions.

External Knowledge

Since the training data is small, external knowledge of event pairs is necessary. Two resources, VerbOcean [22] and ConceptNet [74], are utilized. In this system, whether the relationships described in external knowledge exist between e_1 and e_2 is represented as a binary vector.

1. VerbOcean

   VerbOcean is a resource of fine-grained semantic relations between verbs, which is extracted from Web using a semi-automatic method. There are five relations, *similar*, *stronger-than*, *opposite-of*, *can-result-in* and *happens-before*, and about

Table 4.1: Experimental results (before official evaluation).

	all			after			subevent		
	Pre.	*Rec.*	*F1*	*Pre.*	*Rec.*	*F1*	*Pre.*	*Rec.*	*F1*
dev	16.8	19.6	18.1	16.4	19.4	17.8	22.8	17.5	19.8
test	14.8	12.5	13.6	14.5	12.4	13.3	18.1	9.40	12.4

22,000 relations are extracted. For example, the pair of *attack* and *destroy* has a *happens-before* relation.

These semantic relations can be a clue in the task. In the following example, there is a *happens-before* relation between *arrested* and *extradited*, and it is a clue to estimate a *BEFORE* class.

(41) [. . .] you ask them to ***arrest*** that person and have them ***extradited***.

In the same way, *similar* relation between the events in the following example would be a clue to estimate a *PARENT* class.

(42) I called the RE's office and ***spoke*** with our nurse. She ***said*** a lot of couples opt to take a break because it is very stressful.

2. ConceptNet
   ConceptNet provides a large semantic graph that describes general human knowledge, and 21 interlingual relations are defined, such as *IsA* and *PartOf*. In this system, three relations which are related to events, *HasSubevent*, *HasLastSubevent*, and *HasFirstSubevent*, are used as a binary vector. In the following example, the semantic relation *HasSubevent* between the event pair is a clue to estimate a *PARENT* class.

   (43) In 1963, Sen. Arnon de Mello ***shot*** dead a fellow legislator on the Senate floor, only to escape imprisonment, since the ***killing*** was considered an accident because he was aiming at another senator.

Training

Adam [40] is adopted as the optimizer, and weight decay is used for regularization (0.0001). Dropout is applied for Multi-layer Perceptron. The word embeddings are initialized using pre-trained word embeddings[3], whose dimension is 300, and POS embeddings are randomly initialized, whose dimension is 10. The dimension of the hidden layer is 100.

Since the combination of the event pair is enormous, event pairs within three sentences are targeted. Event pairs that have a gold coreference relation are not utilized for training and testing.

The number of *NONE* class instances is much larger compared to other classes. To handle the class imbalance, an undersampling method is used; a part of *NONE* class instances at a specified ratio are used (the rest of the instances are discarded). The undersampling ratio is determined by using a development set.

Our system is implemented using *Chainer* [81]. Stanford CoreNLP[4] is used for tokenization, sentence segmentation, lemmatization, and POS tagging. When looking up VerbOcean and ConceptNet, a verbal noun is converted to its corresponding noun using NLTK (Natural Language Processing Toolkits) [48] (e.g., negotiation $\rightarrow$ negotiate).

4.1.3 Experiments

Corpus

We used the corpus LDC2016E130 for our experiments, which consists of 158 training documents and 202 testing documents. 30 documents among the training documents were used for the development. For the official evaluation, the system was trained using the same corpus and submitted our three runs.

Experimental Result

Table 4.1 shows our experimental results (before the official evaluation), where the undersampling ratio was set to 0.02. The evaluation measures are precision, recall, and F-score by the official scorer provided by the organizers.

[3]Downloaded from `https://nlp.stanford.edu/projects/glove/`.

[4]`https://stanfordnlp.github.io/CoreNLP/`

Table 4.2: Experimental results for development set where undersampling ratio varies.

undersampling ratio	*Pre.*	*Rec.*	*F1*
1.00	43.0	0.242	0.480
0.10	27.4	5.12	8.63
0.05	21.6	9.59	13.3
0.03	19.0	16.5	17.6
0.02	16.8	19.6	**18.1**
0.01	8.52	24.4	12.6

Table 4.3: Experimental results (official evaluation).

undersampling ratio	all			after			subevent		
	Pre.	*Rec.*	*F1*	*Pre.*	*Rec.*	*F1*	*Pre.*	*Rec.*	*F1*
0.02 (RUN2)	13.3	**12.0**	**12.6**	7.5	**15.0**	**10.0**	16.9	**11.0**	**13.3**
0.03 (RUN1)	15.5	7.7	10.2	12.5	4.4	6.5	15.8	8.5	11.1
0.05 (RUN3)	**23.0**	4.2	7.1	**15.7**	4.8	7.4	**26.6**	4.2	7.1

Table 4.2 shows our results for development set where the undersampling ratio varies. When all possible classes are used, that is, when the undersampling ratio is 1.0, F-score is 0.48, but when the undersampling ratio is 0.02 (98% of *NONE* classes are randomly abandoned), it becomes 18.1. The table shows that the recall is improved by reducing undersampling ratio.

Official Evaluation Result

We submitted the following three runs for the official evaluation where a undersampling ratio just varied (Run1: 0.03, Run2: 0.02, Run3: 0.05). Table 4.3 shows our official evaluation result. Run2 performed the best, and we ranked first among two teams.

4.1.4 Discussion

In the following example, the system correctly outputted *BEFORE* class between the event pair.

(44) Biros ***killed*** the 22 year old Engstrom near Warren in 1991 after offering to drive her home from a bar, then ***scattered*** her body parts in Ohio and Pennsylvania .

Although there is no relation described in external knowledge between events, it is supposed that the word "then" between events, which is considered by the path embedding, could be used for a clue.

In the following example, while the gold class is *NONE*, the system outputted *BEFORE* class.

(45) That way, you are completely finished with the car ***payment***, are only out the difference (instead of the entire amount left that is owed), and have ***purchased*** something cheap in cash.

In this example, the word "and" between events represents a logical relation. However, the system wrongly interpreted it as a temporal relation.

In the following example, the system did not output the correct label *PARENT* but outputted *NONE*.

(46) During testimony last month Al Jayouzi threw his shoes at prosecutors when the ***death*** of his comrades during a fire ***fight*** was discussed.

In this example, the relation is not described in the external knowledge. Thus, we are planning to acquire event knowledge from a large raw corpus, and integrate it into our system.

To reveal the importance of each clue for the classification, each clue was ablated. Table 4.4 shows the result on the development set. We found that external knowledge (both VerbOcean and ConceptNet) was effective. "- GRU" represents GRU was not used, and just word embeddings were used for the word representation. GRU was effective for capturing the context. "w/ LSTM" represents LSTM was used instead of GRU. The performance of LSTM was worse than one of GRU. That is because LSTM has more parameters to train in comparison with GRU, and the evaluation corpus is relatively small for the parameters training.

Table 4.4: Ablation study on the development set.

	F	Δ
Our method	18.1	
- VerbOcean	15.5	-2.6
- ConceptNet	17.0	-1.1
- GRU	16.1	-2.0
w/ LSTM	16.1	-2.0

4.2 Temporal Information of Events

In the previous section, we proposed a model which determines event-event temporal and subevent relations. Although the model considers an intra-sentential context, it mainly focuses on the relations of event pairs.

In this section, we focus on directly anchoring each event to the time axis. We design the following three multi-class classification tasks:

(a) Event Temporality Task (Two classes)

Whether the events have temporality. The temporality defined in chapter 3 is used.

(b) Event Span Task (Four classes)

The temporal span of events. The spans are categorized into four classes: (1) within 1 day, (2) within 1 month, (3) within 1 year, (4) more than 1 year.

(c) Event Occurrence Time Task (Five classes)

The occurrence time of event based on the document creation time. The occurrence times are categorized into five classes: (1) over 3 years ago, (2) 3 years ago∼3 days ago, (3) 3 days ago∼3 days later, (4) 3 years later∼3 years later, (5) after 3 years.

Figure 4.3 shows the relations between event, document creation time (DCT) and the three tasks. To resolve the tasks, we propose neural network models. Our experimental results on our constructed temporal corpus show that while F-score of the first event temporality task was about 90 points, that of second and third tasks were about 50 to 60 points.

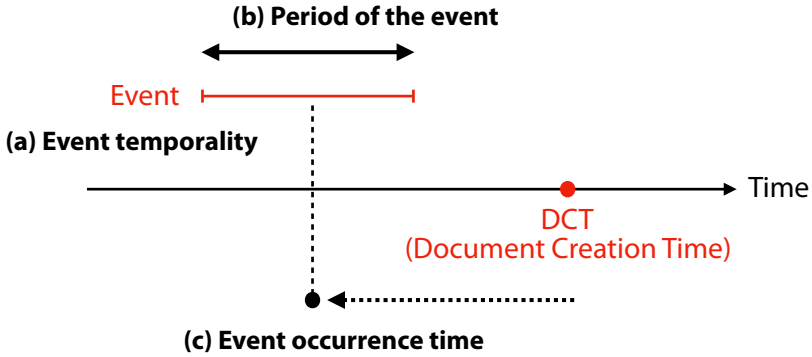

Figure 4.3: Three tasks for anchoring events to time axis.

4.2.1 Definition of the Three Tasks

In this subsection, we describe the detail of the three tasks.

(a) Event Temporality Task

Event temporality task is a task to judge whether target expressions have temporality or not, that is, whether the corresponding time tag is *t:n/a* or not. Out of 4,534 target expressions, 3,438 expressions, which is 76%, have temporality.

(b) Event Span Task

Event span task is a task to predict the span of expressions which have temporality. The spans are categorized into four classes according to the temporal granularity of length, and it is considered as a four-class classification problem. Specifically, the four categories are, (1) within 1 day, (2) within 1 month, (3) within 1 year, and (4) more than 1 year. Expressions whose span are vague, such as time coreference and expressions with span tags are removed from the data. Out of 4,534 target expressions, 2,752 expressions are used. Examples are shown below.

1. Within 1 day: *t:1994-12-31*, *t:1994,span:P1D*
2. Within 1 month: *t:1994-12-25∼1994-12-31*, *t:1994-12,span:P3D*
3. Within 1 year: *t:1994-12*, *t:1994,span:P3M*

4. More than 1 year: *t:1994*, *t:1991∼1995*, *t:PAST*

The proportion of classes is: (1) within 1 day (1,545 expressions, 56%), (2) within 1 month (640 expressions, 23%), (3) within 1 year (135 expressions, 5%) and (4) more than 1 year (432 expressions, 16%).

(c) Event Occurrence Time Task

Event occurrence time task is a task to predict the days from document creation time to the representational date of an event. The spans are categorized into five classes according to the temporal granularity of length, and it is considered as a five-class classification problem. Specifically, the five categories are, (1) over 3 years ago, (2) 3 years ago∼3 days ago, (3) 3 days ago∼3 days later, (4) 3 years later∼3 years later, and (5) after 3 years.

Here, the representational date of an event is the middle date of beginning and ending date of the event. In the case of an event whose beginning date or ending date is vague, such as events represented by the *span* tag, the middle date of the *t* tag is used. For example, when an event is associated with *t:1994-12,span:P3D*, the representational date is December 15, 1994, which is the middle day of December 1994. Expressions whose occurrence date is vague, such as time coreference, are removed from the data. Out of 4,534 target expressions, 3,276 expressions are used. Examples are shown below. The document creation date is January 1, 1995.

1. Over 3 years ago: *t:1990*, *t:PAST*
2. 3 years ago∼3 days ago: *t:1994-06-01*, *t:1994,span:P1D*
3. 3 days ago∼3 days later: *t:DCT*, *t:1994-12-29∼1994-12-31*
4. 3 years later∼3 years later: *t:1995-03,span:part*, *t:FUTURE-M*
5. After 3 years: *t:2000*, *t:FUTURE*

The proportion of classes is: (1) over 3 years ago (284 expressions, 9%), (2) 3 years ago∼3 days ago (879 expressions, 27%), (3) 3 days ago∼3 days later (1,331 expressions, 41%), (4) 3 days later∼3 years later (571 expressions, 17%), and (5) after 3 years (211 expressions, 6%).

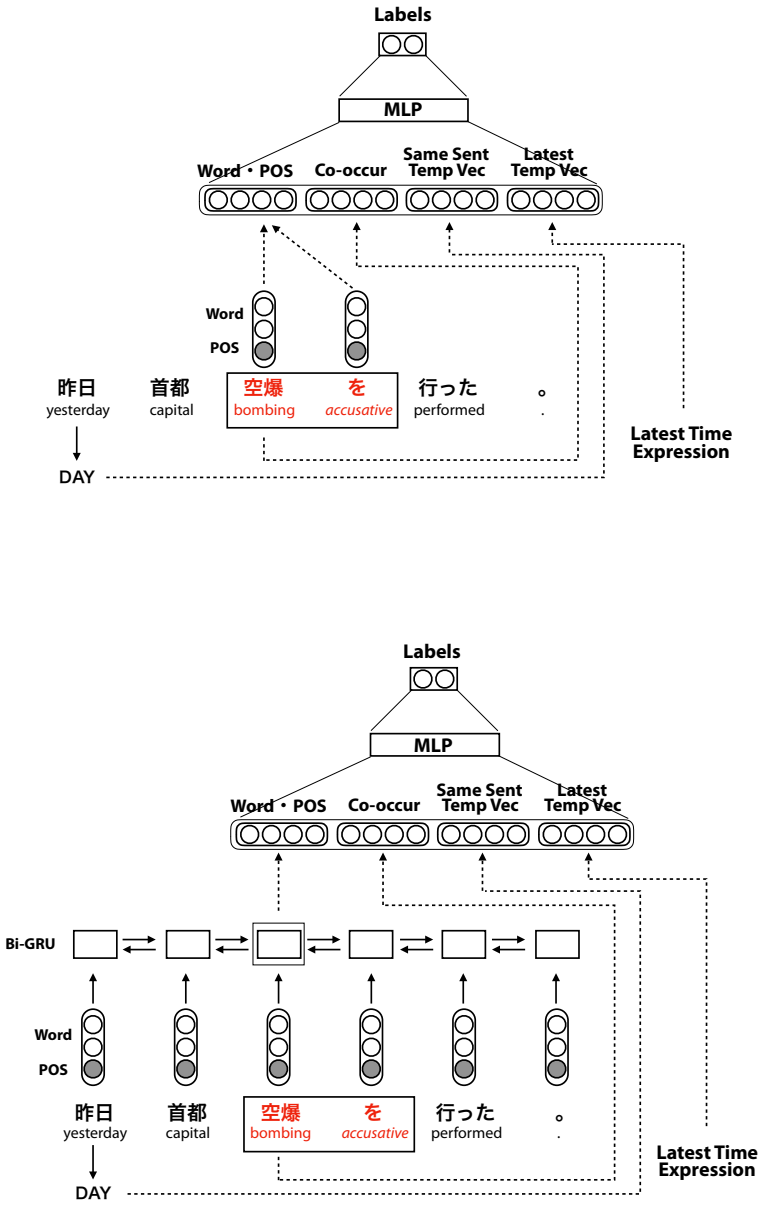

Figure 4.4: Two neural network models that estimate the temporal information of the target expression "空爆を (bombing)." Event temporality model (left) uses only vocabulary information in target expression while event span and event occurrence time prediction model (right) considers contexts.

4.2.2 Models

Lexical information of target expressions is a big clue to judge the temporality of events. For example, an expression "go to work" often has temporality while verbal nouns such as "armor" often do not have temporality. On the other hand, the temporal span of events and the occurrence time are highly dependent on the context. For example, the temporal span of an event is totally different whether the event "stay" continues for "three days" or "one year." Similarly, the occurrence time of an event is different whether it happened "yesterday" or "last year." Therefore, we propose two models, a model focusing on the target expression in the judgment of temporality, and a model focusing on the context in the estimation of the time spans of the event and the occurrence time. Since the target expressions in the corpus have diversity and sparsity, and the target expression itself is an important clue, word distributed representation, such as word2vec [53], is useful.

We propose two-layer perceptron models for the classification. The overall of the model is shown in Figure 4.4. The input vector is a concatenation of the following four vectors, and the difference between the two models is a vector representing the target expression. The detail of each vector is described below.

1. Vector representing the target expression
   In the temporality judgment model, the sum of each word vector in the target expression is used. For example, in the example of Figure 4.4, the two-word vectors "空爆 (bombing)" and "を (*accusative*)," which constitute the target expression "空爆を (bombing-accusative)" are summed up. In the context-based model, first, a bidirectional GRU (Gated Recurrent Unit) [24] is applied to the input sentence, and then the hidden vector of the independent word in the target expression is used. In the example in Figure 4.4, we use the hidden vector of "空爆 (bombing)" which is an independent word in the target expression "空爆を (bombing-accusative)."

   Each word is represented as a concatenation of word embedding and POS (part-of-speech) embedding.

2. Temporal information vector of a temporal expression in the same sentence
   A four-dimensional binary vector represents the temporal granularity of a temporal expression in the same sentence. The vector is composed of binary values whether or not the detected time expression is each granularity of (day, week, month, year).

For example, in the case of the example in Figure 4.4, a time expression "昨日 (yesterday)" in the sentence exists and the four-dimensional vector is (1, 0, 0, 0), only the dimension corresponding to the granularity of "day" is one and the others are zero. If temporal expressions do not exist in the same sentence, the vector is a zero vector. Temporal expressions are detected by a rule-based method. In the 4,534 target expressions, temporal expressions were detected in the same sentence with 1,601 expressions, which is 35% of the total.

3. Temporal information vector of the latest temporal expression
   Similar to the preceding temporal information vector, it is a four-dimensional binary vector which represents the temporal granularity of the latest temporal expression before the input sentence. In the 4,534 target expressions, 3,940 temporal expressions were detected before the sentence, which is 87% of the total.

4. Co-occurrence score vector of time expression into this object expression vector
   It is a four-dimensional real-valued vector representing the co-occurrence of target expressions and temporal granularity. For the temporal granularity, preceding four granularity, (day, week, month, year) is used.

   The vectors are computed as follows. First, temporal expressions in training text are detected by a rule-based method and they are converted to one of the four granularities. Then, co-occurrence scores between target expressions and the four granularities are computed. For the co-occurrence score, Pointwise Mutual Information (PMI) is used. Let x denotes a target expression and y denotes the temporal granularity of a temporal expression in the same sentence. $P(x)$ and $P(t)$ denote their occurrence probabilities, and $P(x,t)$ denotes a joint probability. Similarly, $C(x)$ and $C(t)$ denote the frequency of x and t, $C(x,t)$ denotes an occurrence frequency of x and t in the same sentence, N denotes the number of sentences. The score is computed as follows.

$$PMI(x,t) = \log \frac{P(x,t)}{P(x)P(t)} = \log \frac{\frac{C(x,t)}{N}}{\frac{C(x)}{N}\frac{C(x)}{N}} = \log \frac{C(x,t)N}{C(x)C(t)} \tag{4.6}$$

   13 million sentences in Asahi newspaper from 1984 to 2005 are used for calculating the co-occurrence scores.

4.2.3 Experiments

Settings

The corpus constructed in chapter 3 was used. Time tags annotated in the second stage were used, and the models were trained and evaluated by five-fold cross validation. F1-score was used in the two-class classification task, (a) the event temporality task. Micro-F1 score was used in the multi-class classification tasks, (b) the event span task and (c) event occurrence time task.

The dimension of the hidden layer in the two-layer perceptron is 50. Cross entropy was used for loss function, and Adadelta was adopted for the optimizer. The dimension of the GRU hidden layer is 100.

Word embeddings were initialized using pre-trained word embeddings whose dimension is 200. The pre-trained word embeddings were trained by 9.8 billion Web sentences. POS embeddings were randomly initialized, whose dimension was 10. These embeddings were updated by backpropagation.

Results

The experimental results of each task are shown in Table 4.5. Since the data is imbalanced, the majority class baseline was used. *COVec* in the table represents the co-occurrence score vector and *TempVec* represents the temporal information vector. The baseline achieved 86 F-score points in the event temporality task, 58 points in the event span task, and 41 points in the event occurrence time task. The proposed model achieved about 90, 60, and 50 points in each task.

Table 4.6 shows the confusion matrix under the condition where the score was the best for each task. In the event temporality task, about 70% of the errors are false positive. In the event span task, the model often wrongly predicted “within 1 month” class as “within 1 day” class. In the event occurrence time task, the model often wrongly distinguished between “3 years ago∼3 days ago” class and “3 days later∼3 years later” class. The errors are discussed in the following section.

Table 4.5: Experimental results in the three tasks: (a) Event temporality task, (b) Event span task, and (c) Event occurrence time task. *COVec* in the table represents co-occurrence score vector and *TempVec* represents temporal information vector.

Tasks	(a)	(b)	(c)
Baseline (Majority class)	86.25	58.26	40.63
Neural network model			
Word	90.09	58.76	45.57
Word+POS	90.37	59.34	48.32
Word+POS+COVec	**90.52**	**60.61**	48.72
Word+POS+ +TempVec (same sent)+TempVec (latest)	89.87	59.41	48.81
Word+POS+COVec+TempVec (same sent)+TempVec (latest)	90.35	59.81	**49.69**

4.2.4 Discussion

Event Temoprality Task

In event temporality task, the model using only word embeddings surpassed the baseline by 3 F-score points. This suggests that lexical information of the target expressions is important for judging temporality. The score was improved by using the POS embeddings. It was useful for detecting verbs and verbal nouns which do not have temporality, such as "対して (against)" and "総当たり (round-robin)."

Our approach, which focuses only on the target expression, cannot solve "開発 (development)" in the following examples correctly.

(47) 国連の支援で、総合**開発の**立案などの成果をあげた
With the support of the United Nations, results such as planning comprehensive ***development*** were obtained.
→ Temporality (gold)

(48) **開発の**ゆがみを知る人たち
People who know the distortion of ***development***
→ No Temoprality (gold)

Table 4.6: Confusion matrix of the three tasks.

(a) Event Temporality Task

Gold\System	No Temporality	Temporality
No Temporality	608	488
Temporality	192	3246

(b) Event Span Task

Gold\System	∼1D	∼1M	∼1Y	1Y∼
∼1D	1266	149	15	115
∼1M	332	178	16	114
∼1Y	35	40	14	46
1Y∼	135	74	13	210

(c) Event Occurrence Time Task

Gold\System	∼ -3Y	-3Y∼-3D	-3D∼+3D	+3D∼+3Y	+3Y∼
∼-3Y	105	56	69	41	13
-3Y∼-3D	63	387	333	64	32
-3D∼+3D	59	334	820	76	42
+3D∼+3Y	43	112	118	237	61
+3Y∼	5	35	51	41	79

(49) 経済開発区

economic ***development*** area

→ No Temoprality (gold)

The target expression in Example (47) has temporality since it is a specific development project. However, the "development" in Example (48) and (49) are unspecified or general events and therefore do not have temporality. In order to cope with such examples, it is necessary to consider information such as the expressions surrounding the target expression and corresponding predicate and arguments.

Event Span Task

In this task, the F-score was improved by introducing co-occurrence score vector. For example, although the model with word and POS embedding wrongly predicted "宿泊 (staying)" in the following example, it could correctly predict by adding the co-occurrence score vector which represents the target expression often occurs with a time granularity "day."

(50) ホテルの宿泊者が目撃した。
People ***staying*** at the hotel witnessed.
→ within 1 day (gold)

The model wrongly outputs "within 1 day" class, as it is shown in Table 4.6. The model wrongly outputted the class in all of the following examples.

(51) 兵士が首都から南に脱出している。
Soldiers are ***escaping*** from the capital to the south.

(52) 見逃せないのは労組の圧力だ。
It is the ***pressure*** of labor union that can not be overlooked.

(53) 出稼ぎ世帯の大半は、テレビやバイクを買う。
Most of migrant households ***buy*** televisions and motorbikes.

(54) 啓蒙に力を入れている。
They ***give*** high priority to enlightenment.

Spans of target expressions in the Examples (51), (52), and (53) varies depending on the context. Although all of these expressions can be interpreted as "within 1 day," it is not the case in this context. Although this model takes context into consideration, it is necessary to train with larger data. In Example (54), since the target expression is a light verb, "啓蒙 (enlightenment)" is a clue to the task. It is necessary to acquire temporal information knowledge implying such words from a large corpus.

Event Occurrence Time Task

In this task, the temporal information vector improved the score. It shows that temporal expression in text, such as "avalanches occurred on Friday," is a clue to the task.

Although the model considers only explicit time expressions such as "yesterday" and "1995," there are implicit temporal expressions such as "Vietnam War" and "last World Cup," which are clues to the task. It is necessary to consider wider temporal information using external knowledge such as Wikipedia.

4.3 Timeline Generation

In this section, we propose a timeline generation model which uses a wide context and external knowledge. In Section 4.1, we presented that most of the event pairs do not have sequence and subordinate relations. The results in Section 4.2 showed that it is difficult to directly anchor various events to the time axis. Based on the results, our timeline generation model considers specific event-event relations and anchors events and temporal expressions.

4.3.1 Timeline Generation Task

As a way of multi-document summarization, timeline construction has become popular recently [80, 82, 55]. In SemEval 2015, a shared task, *TimeLine: Cross-Document Event Ordering*, was proposed to create a timeline in which events related to a given target entity are extracted from a set of news articles, and they are ordered along the time axis [55]. For example in Figure 4.5, a timeline of the target entity "iPhone 4" is generated from articles related to the topic "Apple Inc." A timeline consists of an ordered list of <time value, event> pairs, and the finest granularity of time values is day. In the figure, underline denotes events, red denotes events related to the target entity "iPhone 4," blue denotes phrases which corefer the target entity and green denotes temporal expressions.

The timeline generation task consists of two subtasks: extraction of events related to a target entity, and anchoring those events to appropriate time values. The severe problem lies in the latter subtask. In anchoring events to time values in a document, easy cases and difficult cases are mixed up. In some cases, an event expression is explicitly modified by a time expression; in other cases, the time value cannot be estimated without understanding the context.

For example, "introduced" in the second sentence in Figure 4.5 is a relatively easy case, since it has a dependency relation with the corresponding temporal expression "7

June." On the other hand, the event "announced" in the first sentence cannot be anchored correctly to 2010-06-07 without using the event-time anchoring result of the same event, "introduced" in the next sentence.

The contribution of our work is to propose a two-stage event-time anchoring model which enables us to consider a wider context than previous work.

The TimeLine task of SemEval 2015 has two tracks: Track A and Track B. In Track A, raw texts are given as input; in Track B, texts with gold event mentions are given. Since we focus on the two problems, namely, extracting target-entity-related events and anchoring events to time values, Track B setting is used. Our experimental results show that the proposed method surpasses the state-of-the-art system by 3.5 F-score points.

4.3.2 Related Work

Several works tackled the TimeLine task of SemEval 2015.

HeidelToul team (Moulahi et al. [58]) proposed a rule based approach. They first extract sentences and events which are relevant to the target entity. They apply string matching using cosine similarity matching function with a threshold, and also apply entity coreference resolution using Stanford CoreNLP [51] to extract terms which refer to the target entity. Then, temporal expressions are extracted and normalized by HeidelTime [77], and associated with events in the same sentence. Finally, the events are pruned using the token distance between event and the closest term which refers to the target entity.

GPLSIUA team (Navarro and Saquete [59]) proposed another rule based method using two clustering processes. They first extract events which are relevant to the target entity. They resolve the named entity recognition and coreference resolution using OPENER web service[5] and extract sentences which include the target entity or its coreference entity. Events in the sentences are selected as relevant events of the target entity. Next, they apply two clustering processes in sequential order: temporal clustering and lemma clustering. The idea of the clustering is that events which occur on the same date and refer to the same fact are regarded as coreferent events. In temporal clustering, they extract temporal expressions, events and links between them using TIPSem [47], and group events which occurred on the same date. Lemma clustering groups events which have the same head word lemma, the same date, and the same target entity.

[5] http://www.opener-project.eu/webservices/

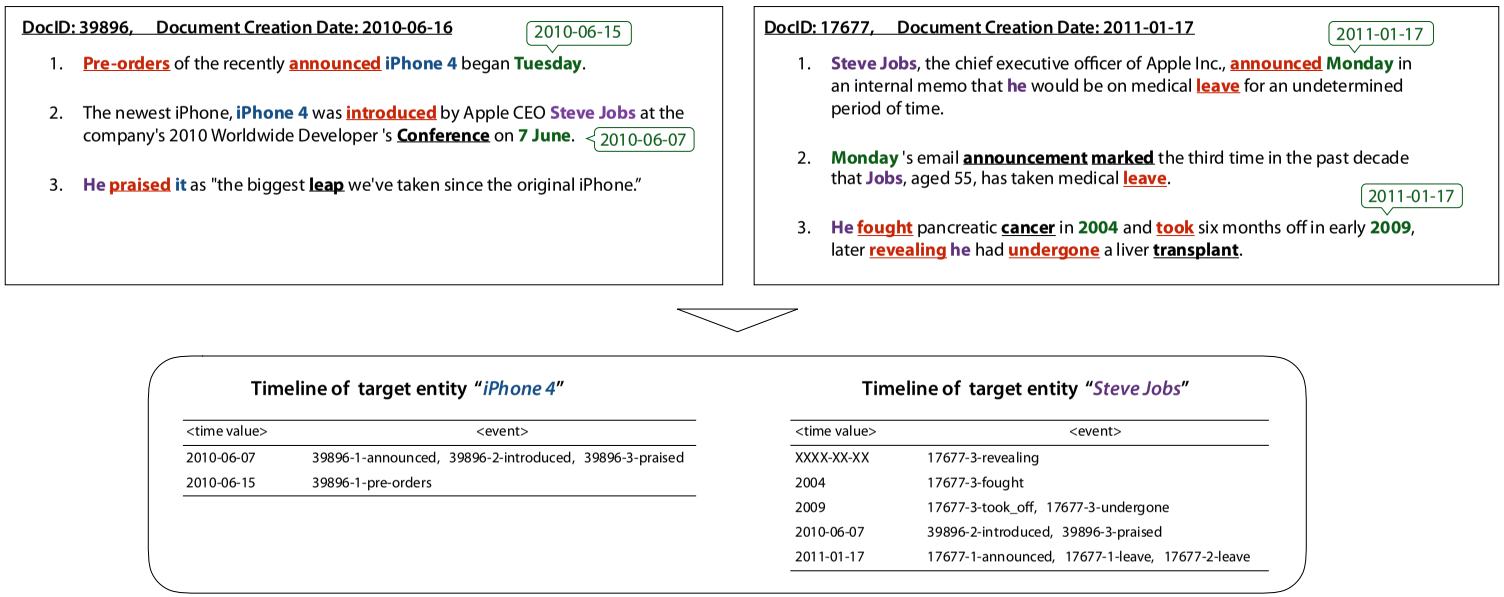

Figure 4.5: An example of timelines of target entities "iPhone 4" and "Steve Jobs." Underline denotes events, red denotes events related to the target entities, blue denotes phrases which corefer the target entities and green denotes temporal expressions.

Navarro et.al. [60] improved the GPLSIUA system. In extracting target entity related events, they additionally consider whether the event and the target entity have a *has_participant* relation with the semantic role ARG0 or ARG1 in the Propbank Project [61]. In the clustering processes, they expand the lemma clustering by using synonymy relations and added distributional clustering after the lemma clustering. Distributional clustering groups semantically compatible events which do not have the same lemma or synonyms.

Cornegruta and Vlachos [25] first introduced a supervised approach to this TimeLine task. They estimated <event, target entity> and <event, temporal expression> anchoring by machine learning. Since the gold timelines consist of <time value, event> pairs, they first generate pseudo training data using distant supervision method. They recognize entities by approximate string matching with the Stanford Coreference Resolution System [44], and extract temporal expressions using UWTime temporal parser [45]. Correct <event, target entity> labels are generated by associating each event with the nearest mention of the target entity in the same sentence. Similarly, each event is associated with the nearest temporal expression which has consistency in the <event, time value> pair in the gold timeline. After that, they train each anchoring using alignment model at the document level with global information. The difference with our event-time anchoring is that they anchor events to temporal expressions, though we anchor events to time values. Another difference is that they imposed a first order Markov assumption and used only preceding information, though we use wider context information.

Laparra et al. [43] proposed a rule based method using tense information in Track A of the TimeLine task. They extracted events and temporal expressions by a semantic role labeling tool, MATE Tools [13] and TextPro suite [30] respectively. They first expand the target entity using DBpedia and extract events which have the target entity as their ARG0 or ARG1. Events are anchored to corresponding time values by a rule-based strategy which uses tense information.

4.3.3 Model

The proposed method first anchors all the events in a document to time values. Then, events related to a target entity are extracted by matching the event specification phrases in a document to various expressions denoting a target entity. A timeline of a target entity is generated by ordering relevant events according to their anchored time values.

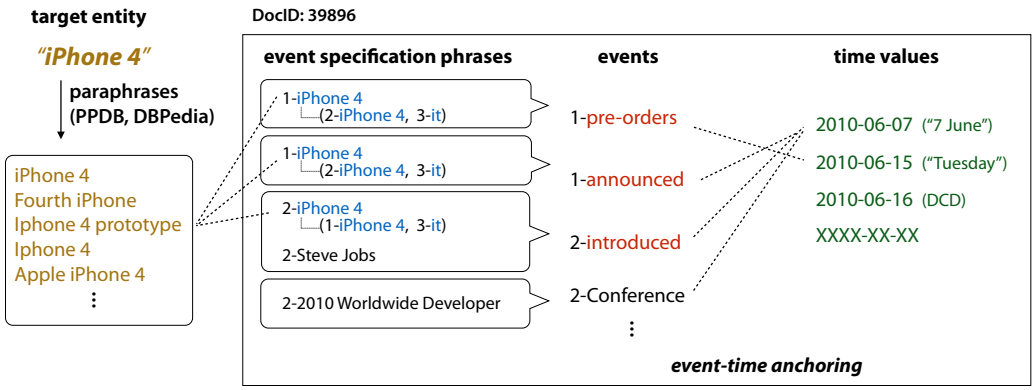

Figure 4.6: Process of timeline construction: anchoring events to appropriate time values and extraction of target-entity-related events.

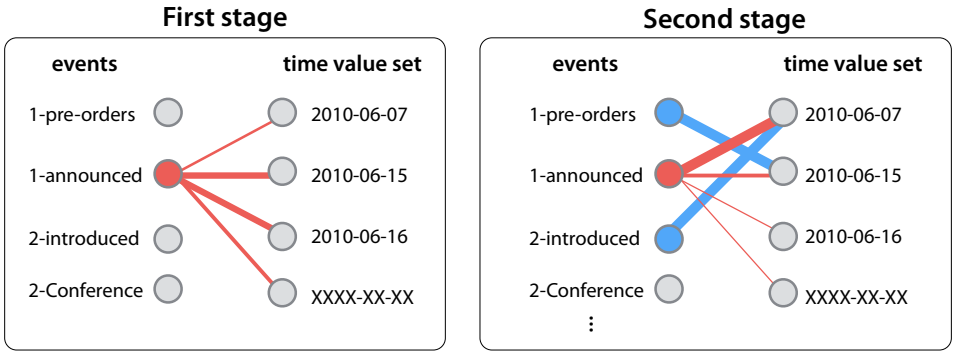

Figure 4.7: Outline of the two-stage event-time anchoring method. In the first stage, each event estimates the probabilities of associating time values. In the second stage, each event updates the probabilities considering its neighbour events (blue events), and is associated to the time value which has highest probability.

Figure 4.6 exemplifies the proposed method. We describe these steps in detail in the following subsections.

4.3.3.1 Anchoring Events to Appropriate Time Values

We start with the task of anchoring events in a document to appropriate time values. We assume that all events in a document are given as gold data since we use the setting

of Track B of the TimeLine task. First, the extraction of time values in a document is explained. Then, the two-stage event-time anchoring method is described.

Time Value Set

Each event in a document corresponds to either the time value represented by a time expression in a document, or the document creation time, or uncertain time value. When the time value of an event is uncertain, it is treated as corresponding to the special time value "XXXX-XX-XX." As Cornegruta and Vlachos [25], we use UWTime [45] to detect and normalize temporal expressions in documents. Note that temporal expressions which do not represent dates but periods like "six months" are removed. We call the set of time values in a document as *time value set*.

Training Data for Event-Time Anchoring

Although it is desirable that all events in a document are anchored to appropriate time values in event-time anchoring training data, the annotated data of the TimeLine task provides only the event-time correspondences related to specific target entities. We use the annotated data as pseudo training data by ignoring the unanchored events.

Learning to Rank in Two Stages

The selection of the most relevant time value for an event among time value set is relative, and it is appropriate to use the framework of learning to rank. Learning to rank is performed so as to make *the score of selecting the correct time value* larger than *the score of selecting other time values* for each event.

As described in the Introduction section, in anchoring events to time values in a document, easy cases and difficult cases are mixed up. To cope with such a mixed problem, we considered a two-stage method: the first stage estimates event-time relations using local features, and the second stage estimates event-time relations again using global features including the first stage estimation results (Fig. 4.7).

The local features are extracted from the event expression and a time value/expression. They are all binary and are classified into the following three types:

1. Features of the event expression:

- tense, aspect and POS tag of the event expression.
- the event expression is a communication event such as "say" and "announce" or not.
- the event expression is included in the headline of the document.
- the event expression has a direct dependency relation with any temporal expression.

2. Features of a time value/expression:

   - a time value is the document creation time (DCT), next day of DCT, future from DCT, or uncertain.
   - the granularity of a time value is day or larger.
   - a time expression depends on the dependency root of the sentence.

3. Features concerning a pair of the event expression and a time value/expression:

   - the event expression is before or after a temporal expression.
   - the event expression and a temporal expression are in the same sentence or not; they have a direct dependency relation or not.

The global features represent the relation between the event under consideration, E_c, and its four types of neighbour events: the preceding event, the following event, the nearest event that has the same stem with E_c, and the nearest event that has the same tense with E_c.

For each of these four events, E_x, we use the following global features:

- E_x exists or not.
- E_x is a communication event or not.
- E_c and E_x are in the same sentence or not.
- the sentence distance between E_c and E_x.
- E_x's time value estimated in the first stage and its confidence score.

4.3.3.2 Selection of Target-Entity-Related Events

Next, events related to a given target entity are extracted in a document. We realize a flexible extraction both by expanding a target entity expression and by collecting event-related phrases in a document.

A target entity can be expressed in various expressions in a text. For example, a target entity "Toyota" can appear in a text as "Toyota Motor" or "Toyota Company." Therefore, we expand a target entity by using two external knowledge.

- DBpedia
  Paraphrases of proper nouns are acquired from DBpedia[6], using redirect links [43]. For example, by using DBpedia, we acquired 40 paraphrases of "Toyota," such as "Toyota Motor," "Toyota Motor Corp.," "Toyota Jidosha Kabushiki-gaisha," and "Toyota cars."
- Paraphrase Database (PPDB)
  A target entity is sometimes not a named entity, but an ordinary expression like "stock markets worldwide." Since most entries of DBpedia are named entities, we employed The Paraphrase Database (PPDB)[32], to obtain paraphrases of ordinary expressions. By using PPDB, for example, we can obtain "stock markets around the world" as a paraphrase of "stock markets worldwide."

For each event, we need to extract what is the event about from the context. For example, the event "introduced" in the second sentence in Figure 4.5 is about "iPhone 4" and "Steve Jobs." We call them *event specification phrases* (ESPs in short).

First, we apply dependency parsing (Turbo Parser [52]), and for each event expression, we extract phrases in sub-trees under its children and its siblings, as ESPs. Furthermore, when a phrase in ESPs corefers other expressions in a document, they are added to ESPs. BART [89] is used for coreference resolution. For example, in the case of the event "praised" in Figure 4.5, since "it" corefers "iPhone 4," "iPhone 4" is also included in ESPs.

As a final result, if there is any exact match between ESPs of the event and the paraphrases of the target entity, the event is judged to be related to the target entity.

[6] http://dbpedia.org/

Table 4.7: Results on SemEval 2015 task-4 Track B.

	Airbus	GM	Stock	Total		
System	*F1*	*F1*	*F1*	*Pre.*	*Rec.*	*F1*
HeidelToul	16.50	10.82	25.89	13.58	28.23	18.34
GPLSIUA	22.35	19.28	33.59	21.73	30.46	25.36
(Navarro+, 2016)	26.21	21.08	31.58	23.68	30.37	26.61
(Cornegruta+, 2016)	25.65	26.64	32.35	29.05	28.12	28.58
One stage	28.32	27.49	16.49	30.60	20.54	24.58
One stage+DBpedia+PPDB	27.46	27.42	31.83	30.07	28.58	29.31
Two stages	**31.06**	**29.52**	18.71	32.42	22.57	26.94
Two stages+DBpedia+PPDB	29.63	29.44	**36.34**	**32.50**	**31.64**	**32.06**

4.3.4 Experiments

The dataset used in the SemEval 2015 TimeLine task is composed of articles from Wikinews. The development dataset consists of timelines for six target entities (e.g. "Steve Jobs," "iPhone 4"), generated from 30 documents related to "Apple Inc." The test dataset consists of three documents set, each of which related to "Airbus and Boeing," "General Motors, Chrysler and Ford," and "Stock Market," and each set has 30 documents. Each corpus is associated with a dozen of target entities. A target entity is one of person, organization, product, and financial entity. For example, the Airbus corpus is associated with "Airbus A380," "Singapore Airlines," and "United Air Force." The GM corpus is associated with "Daimler Chrysler" and "Toyota." The Stock corpus is associated with "Bank of America" and "Dow Jones Industrial Average." Output timelines are evaluated by the time value of events and the order of events, and Precision, Recall, and F-score are calculated.

In our experiments, we utilized SVM-rank [37] as a learning to rank tool. We compared our results with four systems. HeidelToul and GPLSIUA are systems participating in SemEval 2015 task-4 TrackB, and the rest are systems developed after that. We used the development dataset for training as the system of Cornegruta and Vlachos [25], which is the only machine learning approach system.

The results of the experiment are shown in Table 4.7. The proposed method surpasses the state-of-the-art by 3.5 points in F-score. Looking at the results of the proposed method in detail, the two-stage model is 2.7 points better than the one-stage model which just utilizes local features. Expansion of target entity expressions using DBpedia, PPDB improved the recall scores significantly.

4.3.5 Discussion

Our task is divided into two parts. One is anchoring events to time, and the other is selecting events which are related to the given target entities. In this section, we discuss the results of the two subtasks. Table 4.8, 4.9, and 4.10 show the detail results. #Events in the tables indicates the number of events in each gold timeline and the *all* column indicates the unique number of events. Since some events are shared between several target entities, the number of unique events and the total number of events are not the same. Event-Time row indicates the accuracy of anchoring, i.e., how many events in each gold timeline are anchored to time correctly. *One* and *two* indicate one- and two-stage models and *+X* indicates taking consideration of an uncertain time value "XXXX-XX-XX." Selection of target-entity-related events is evaluated by F-score. *+D+P* indicates the use of DBpedia and Paraphrase knowledge. Timeline evaluation is performed with the two-stage model using DBpedia and Paraphrase by using official evaluation methodology.

Anchoring Events to Time Values

Our experimental results show that the second stage improved the result of the first stage in every corpus significantly: 5 F-score points improvement in the Airbus and GM corpora and 13 points in the Stock corpus. Following the official evaluation metric, an uncertain time value "XXXX-XX-XX" was not considered in this evaluation. When it is considered in, the accuracy scores of the second stage decreased in the Airbus and GM corpora (*+X* row in the tables). The score did not decrease in the Stock corpus because there are almost no uncertain values in the corpus while about 30% of time values are uncertain in the other corpora. In the second stage, the model tends to anchor events to non-uncertain time values. The number of events anchored to an uncertain time value reduced by more than 30% in the second stage compared to the first stage.

In the first stage, events which have dependency relations with temporal expressions tend to be correctly associated with the corresponding time value. For example, the event "entered into" in the following sentence is correctly associated with the time value 2007-08-10 ("Friday," DCT).

(55) [DCT: 2007-08-10]
On Friday, the Fed entered into a $38 billion repurchase agreement of mortgage-backed securities, easing stockholder worries.

In the second stage, events which are referred in other sentences are modified. The following example consists of two consecutive sentences.

(56) [DCT: 2005-06-13]
(a) Ryanair exercises options on five Boeing 737s.
(b) Irish low cost airline, Ryanair, announced today that it is exercising its options with Boeing to purchase five new 737 aircraft.

In the first stage, while the system correctly associated the event "exercising" in the second sentence to 2005-06-13 ("today," DCT), the event "exercises" in the first sentence was wrongly associated to XXXX-XX-XX. However, in the second stage, the anchoring is modified to 2005-06-13 by using the information of "exercising" in the next sentence.

The majority of errors are due to our not considering event-event temporal and semantic relations. For example, there is an implicit temporal relation between "purchase" and "deliver." Since the amount of training data is not enough for acquiring these relations, using distant supervision or external knowledge would be needed. Some errors are related to the temporality of events. For example, a verb "plan" tends to represent events in the future. There are also errors related to event-time features. Especially in complex sentences, not only the information of direct dependency relations but also the structure of sentences and semantic roles are essential to identify the event-time relations.

Selecting Target-Entity-Related Events

The expansion of target entities improved the recall score significantly in every corpus. The score increased 52 points to 59 points in the Airbus corpus, 66 points to 79 points in the GM corpus, and 26 points to 57 points in the Stock corpus. When we focus on target

Table 4.8: Detail experimental results on Airbus corpus. #Events indicates the number of events in each gold timeline. *One* and *two* in Event-Time anchoring row indicate one and two stages models. *+X* indicates taking consideration of an uncertain time value "XXXX-XX-XX." *+D+P* in Event Selection raw indicates use of DBpedia and Paraphrase knowledge.

	#Events	Event-Time (Acc.)				Event Selection						Timeline		
				+X					+D+P					
Target Entity		One	Two	One	Two	*Pre.*	*Rec.*	*F1*	*Pre.*	*Rec.*	*F1*	*Pre.*	*Rec.*	*F1*
Airbus A380	60	48	50	48	48	49	68	57	49	68	57	42	24	30
Airbus	79	38	46	41	44	81	31	45	80	43	56	27	40	32
Boeing 777	99	56	68	54	52	60	66	63	60	66	63	75	60	67
Boeing 787 Dreamliner	9	100	83	89	78	83	56	67	78	78	78	27	29	28
Boeing	37	54	67	46	49	89	22	35	83	54	66	28	38	32
China Eastern Airlines	8	38	25	38	25	100	63	77	100	63	77	20	11	14
EADS	11	50	38	55	36	91	91	91	11	91	20	3	30	6
Louis Gallois	6	80	80	83	83	83	83	83	83	83	83	86	71	78
Northrop Grumman	11	71	71	73	64	53	82	64	53	82	64	21	33	26
Ryanair	17	70	60	65	53	90	53	67	90	53	67	45	42	43
Scott Carson	3	100	100	100	100	50	100	67	50	100	67	50	100	67
Singapore Airlines	14	46	69	50	64	56	71	63	56	71	63	31	28	29
United Air Force	27	67	33	48	19	25	7	11	39	25	30	11	17	13
all	336	52	57	51	49	62	52	56	52	59	55	26	34	30

Table 4.9: Detail results on GM corpus.

	#Events	Event-Time (Acc.)				Event Selection						Timeline		
				+X					+D+P					
Target Entity		One	Two	One	Two	*Pre.*	*Rec.*	*F1*	*Pre.*	*Rec.*	*F1*	*Pre.*	*Rec.*	*F1*
Alan Mulally	7	100	100	14	29	78	100	88	64	100	78	8	100	14
Barack Obama	30	77	96	70	87	90	84	87	90	87	89	70	85	77
Chrysler	40	27	33	38	33	60	73	66	62	78	69	2	5	3
Daimler Chrysler	8	100	100	63	25	0	0	0	83	63	71	14	17	15
Fiat	11	50	67	55	73	73	100	85	73	100	85	45	63	53
Ford	59	39	36	37	34	88	83	85	88	83	85	22	25	24
Frederick Henderson	31	14	21	19	19	0	0	0	90	81	85	23	19	21
General Motors	117	37	41	39	39	66	69	67	61	73	66	27	35	30
General Motors creditors	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Jim Press	3	100	100	100	100	100	100	100	100	100	100	100	100	100
Toyota	8	50	17	50	13	88	88	88	70	88	78	17	38	30
United Automobile Workers	5	0	0	20	40	0	0	0	50	80	62	0	0	0
all	305	41	46	40	41	73	66	69	71	79	75	25	35	29

Table 4.10: Detail results on Stock corpus.

	#Events	Event-Time (Acc.)				Event Selection						Timeline		
				+X					+D+P					
Target Entity		One	Two	One	Two	*Pre.*	*Rec.*	*F1*	*Pre.*	*Rec.*	*F1*	*Pre.*	*Rec.*	*F1*
Bank of America	17	41	47	41	47	73	47	57	60	53	56	38	15	22
CAC 40	4	75	50	75	50	75	75	75	80	100	89	86	86	86
Dow Jones Industrial Average	94	40	65	40	65	77	28	42	85	66	75	58	33	42
FTSE 100 index	24	42	54	42	54	75	25	38	82	75	78	69	37	48
General Motors	3	33	33	33	33	100	100	100	100	100	100	67	20	31
NASDAQ composite	33	46	39	46	39	69	27	39	69	27	39	50	12	20
New York Stock Exchange	3	50	50	67	33	100	100	100	100	100	100	50	40	44
NIKKEI 225	19	37	42	37	42	100	26	42	100	68	81	63	29	39
S&P 500	22	46	50	46	50	0	0	0	80	36	50	47	13	20
stock markets worldwide	27	41	67	41	67	100	11	20	79	41	54	68	24	36
US dollar	2	50	50	50	50	0	0	0	8	50	14	7	33	11
US Federal Reserve	13	62	69	62	69	50	8	13	73	85	79	53	47	50
US stock markets	16	47	53	44	50	56	31	40	46	31	37	29	15	20
all	263	43	56	43	55	76	26	39	75	57	65	54	27	36

entities, the expansion effected in four of the twelve target entities in the Airbus corpus, six of the thirteen entities in the GM corpus, and nine of the twelve entities in the Stock corpus. In total, recall scores increased in 19 of 37 entities, which is about half of target entities.

On the other hand, the precision scores decreased as the trade-off. The decrease of precision is caused by acquiring terms which are relative but not paraphrase. Results on F-score show that the expansion does not have much effect in the Airbus corpus, and 1 point decreased. In the GM and Stock corpora, the scores increased significantly: 6 points improvement in the GM corpus and 31 points in the Stock corpus.

In the Airbus corpus, the results show that the expansion did not perform well in total. It is due to one target entity, "EADS," the abbreviation for European Aeronautic Defence and Space Company. Although the scores of other entities are improved, "EADS" significantly decreased the score. Our system acquired "Airbus group" and "Airbus company" as the expansion of a target entity "EADS." However, EADS is the predecessor of the Airbus group, and they are not the same entity in the document creation time. In the expansion of time-sensitive entities (e.g., organization, position, and facility), it is necessary to consider time information.

In the GM corpus, though the F-score of selecting target-entity-related events was improved by the expansion, the F-score of generated timelines slightly decreased. Most of the improvement in the GM corpus is due to a target entity "Frederick Henderson," the Chief Executive Officer of General Motors. He is mentioned as "Fritz Henderson" in the articles, which can be extracted in DBpedia. By using DBpedia, F-score of events selection increased from 0 points to 85 points. However, since 95% of the events are in one document and the event-time anchoring model did not work well in the document, the advantage of target entity expansion did not lead to improvement in the final result.

In the Stock Market corpus, more significant improvement is achieved than the other corpora. This is due to the category of target entities. In the Airbus and GM corpora, most of the target entities are company names, product names and person names (e.g., "China Eastern Airlines," "Boeing 777," and "Barack Obama"). These entities are often written without abbreviated at their first appearances in document (e.g., "Barack Obama"), and they are abbreviated from the second appearances (e.g., "Obama"). Thus, many target-entity-related events are extracted by just using string matching and coreference

resolution. On the other hand, in the Stock Market corpus, most of the target entities are indexes and money expressions (e.g., "Dow Jones Industrial Average," "FTSE 100 index," and "US Dollar"), which are usually abbreviated or paraphrased. For example, "Dow Jones Industrial Average" is usually written as "the Dow Jones" or "the Dow Industrials." In these cases, much more related events can be extracted by using knowledge of paraphrases.

4.4 Discussion

In this chapter, we analyzed temporal information of events from three viewpoints. One is event-event ordering, the second is anchoring each event to the time axis, and the third is timeline generation by associating multiple events with time. The score in each task was relatively good. However, these tasks are not completely tied together. Especially, it is a big problem that the event-event ordering technique is not used for timeline generation. One of the causes is the difference in the granularity of time. The minimum granularity of time dealt with in the timeline generation task is day, whereas the event-event ordering task deals with finer granularity than that. Therefore, it is necessary to train each model in a different corpus, which makes the interaction between the models difficult. Recently, Cheng and Miyao [19] presented an idea which addresses this problem. They proposed a framework to automatically induce the ordering of events from a corpus that anchors events to time. Using this idea, it would be possible to train a timeline generation model and an event-event ordering model jointly. However, in order to deal with event nuggets in Section 4.1, a sequence of events that occur as part of a whole event, another idea is required.

We studied in the newspaper domain, which includes many explicit temporal clues (e.g., temporal expressions) and official events. Here we discuss issues that arise when dealing with more personal text such as blogs and tweets. The problem from the viewpoint of time is that there are few time clues. In the temporal corpus that we constructed in Chapter 3, temporal common sense is implicitly considered. For example, a conference ends in a few days, and a TV drama lasts a few months. Our experimental results in Section 4.2 showed that it is difficult to estimate these temporal spans from word representations and contexts. To overcome the issue, it is essential to construct temporal

corpora and temporal knowledge which explicitly represent the temporal common sense.

There are also several problems from the viewpoint of events. Although the definitions of the events dealt with in this research were slightly different in each task, they were realistic and relatively objective. However, in the personal text, many non-realistic things and events are written. To analyze such text, understanding various modalities such as factuality, negation, and condition are essential for generating credible timelines and storylines. In addition, although our research dealt with events uniformly, the viewpoint and the stance of events differ if the newspaper company and the reporter are different. In order to integrate and compare different types of text information, incorporating stance detection techniques is necessary.

4.5 Summary of this Chapter

In this chapter, we described three studies of temporal information of events. We first proposed a model which determines temporal relations (before-after) and subevent relations (parent-child) between events. The model is based on a neural network approach using external knowledge. Since most of the event pairs do not have temporal nor subevent relations (NONE class), the class imbalance is eliminated by using an undersampling technique. Our model achieved F-score of 12.6 points for the official evaluation in TAC2017 workshop and ranked the first among two teams. In development dataset, the model achieved 18.1 F-score points. When all event pairs are used in training, precision was 43.0 points, but F-score was 0.48 points. The highest F-score was observed when 98% of NONE class event pairs are randomly abandoned in training.

Then we tried to anchor events to the time axis. We designed three multi-class classification tasks. The first task, *event temporality task*, is a two-class classification task to judge whether target expressions have temporality or not, The second task, *event span task*, is a four-class classification task to predict the span of expressions which have temporality. The third task, *event occurrence time task*, is a five-class classification task to predict the days from document creation time to the representational date of an event. We proposed neural network models for these tasks. In the event temporality task, we designed a model which focuses on the lexical information of events and neighbor temporal expressions information. In the event span task and event occurrence time task,

we additionally considered intra-sentence context by using a bidirectional GRU. Our experimental results on our constructed temporal corpus show that F-score of the event temporality task was 91 points, that of event span task was 61 points, and that of event occurrence time task was 50 points.

Finally, we tackled the timeline generation task. The task consists of two subtasks: selection of events related to a target entity, and anchoring those events to appropriate time values. In the first subtask, we used external knowledge to detect target entities in various forms, and selected target-entity-related events using dependency relations. In the second subtask, we proposed a two-stage event-time anchoring model. In the first stage, events are anchored to time using local features, and in the second stage, the anchorings are modified using global context features. Our experimental results showed that our model surpassed the state-of-the-art system by 3.5 F-score points in the TimeLine task of SemEval 2015. By using the two-stage model, the F-score was improved by 2.4 points compared to the one-stage model. Furthermore, the expansion of target entities achieved another increase of 5.1 points.

Chapter 5

Conclusion

5.1 Summary

To understand the present and predict the future, it is essential to know the past. Every day many texts are generated on the Web, and a massive amount of text has been accumulated so far. This information space is becoming to be able to know not only the latest information but also events and knowledge in the past. To extract knowledge about a specific topic from this massive amount of text, which was written at and refers to a variety of time periods, it is necessary to interpret the temporal information implied in the text and integrate, summarize, and compare its contents along the time axis.

Storyline is a structured chronology, which organizes information along the time axis in a reader-friendly manner. It consists of various information such as events, actors, emotions, value judgment, opinions, and their relations. We mainly focused on the core skeleton of the storyline, namely timeline, which is a structure anchoring events to the time axis. The contributions of this study are three-fold: (1) we proposed a temporal expression resolution model which is robust to loose structures; (2) we constructed a temporal corpus which anchors various expressions in text to the time axis comprehensively; (3) we proposed a timeline generation model which considers a wide context.

In Chapter 2, we focused on temporal expressions in text. Temporal expressions are vital for textual temporal analysis and several methods of recognizing and normalizing them have been proposed. However, analyzing loose structures of temporal expressions was a remained problem. We tackled the problem and proposed a neural network model

which robustly composes basic temporal expressions and absorbs their rich diversity of combination. Our experimental results showed that our model achieved 83.4 F-score points in the dataset of TempEval-3.

In Chapter 3, we constructed a temporal corpus for timeline and storyline tasks. We proposed a new annotation scheme which represents various types of time information of expressions. We dealt with not only the expressions which have clear temporality but also expressions with weak temporality. Using the scheme, we annotated 4,534 expressions and constructed a new corpus based on Japanese newspaper. The tags that are newly proposed in this study account for 25% of the whole. Since the corpus has already been annotated with predicate-argument structures and coreference relations, our annotation makes it possible to utilize for integrated information analysis of events, entities, and time.

In Chapter 4, we studied temporal information analysis of events. Since the number of temporal expressions in text is small, analyzing temporal information of events is essential. We first focused on the relations between events and proposed a neural network model which determines event-event sequential and subordinate relations. Our model achieved F-score of 12.6 points in the Event Sequence Detection Task in TAC2017 workshop. We then focused on anchoring events to the time axis directly and designed three tasks: event temporality task, event span task, and event occurrence time task. Our experimental results on the temporal corpus constructed in Chapter 3 showed that while the F-score of the event temporality task was about 90 points, that of span and occurrence tasks were about 50 to 60 points. Finally, we generated timelines from newspapers. We proposed an event-time anchoring model which considers external knowledge and a wide context including event-event relations. Our experimental results showed that our model achieved 32.1 F-score points in the TimeLine task of SemEval 2015.

5.2 Future Work

In the rest of this chapter, we discuss the future work.

5.2.1 Toward More Accurate Timeline Construction

Our timeline generation model performed better than previous studies, but the accuracy is not good enough for practical use. In order to generate more accurate timelines, the following problems need to be addressed.

Transfer Learning Techniques Some models proposed in this thesis utilize neural network model. In this thesis, we used word representations which focused on word co-occurrences. Recently, context-dependent representations are actively researched [69, 26]. They pre-train representations from a raw text and train on the specific tasks by fine-tuning. It is reported that this transfer learning provides high accuracy, and it would also be effective in the tasks we tackled.

Acquisition of Expressions which Represent Temporal Information Implicitly In this thesis, we dealt with the temporal information of explicit temporal expressions. However, there are many expressions which indirectly represent temporal information. For example, "Nagano Olympic" was held in 1998, and "the earthquake in Kobe" would indicate the earthquake on January 17, 1995. By acquiring such implicit temporal information, the accuracy of temporal information analysis techniques will increase. One possible way is to utilize external knowledge on the Web. Another way is to utilize the idea of masked language model [26]. In the masked language model task, some of the tokens in the input sentence are randomly masked and systems predict the original vocabulary of the masked words based only on their contexts. Since the date of an event is often specified in the sentence, resolving the task will provide valuable temporal knowledge.

Cross-Document Event and Entity Coreference Resolution Timelines we constructed in this thesis are based on intra-document event-time anchoring techniques. However, especially in news articles, the detail temporal information of some events is typically written in other articles. For example, considering only the information in the document, events in the third sentence in Figure 1.5 are anchored to 2017. However, if a system incorporates information in the document in Figure 1.2, it is revealed that those events happened on December 5, 2017. The same is true for entities. To generate a consistent and accurate timeline of multiple documents, in-

corporating cross-document event and entity coreference resolution techniques is required.

5.2.2 Toward Storyline Construction

In this thesis, we focused on timeline construction. To construct a storyline, which incorporates various types of information into a timeline, the following problems are needed to be addressed.

Various Relations between Events In this thesis, we dealt with two relations between events, sequential and subordinate relations. To generate informative storylines, associating events with various relations is desirable. The biggest problem in dealing with other semantic relations is that there are few training data. One possible way is to resolve several relations jointly. For example, Mirza and Tonelli [56] focused on causal relations, which are deeply related to temporal relations. They proposed a sieve-based system to extract and classify both the temporal and causal relations and showed the effects of the interaction between the two relations. The accuracy of recognizing these relations is still low and there is plenty of room to improve them.

Subjective Information We generated timelines from events in news articles, which is a relatively neutral and factual domain. However, one of the characteristics of the Web is that it contains texts of various people from various perspectives, such as blogs and SNS (Social Networking Service) text. In order to construct informative storylines from these texts, incorporating subjective information such as opinions, emotions, and author information is required.

Organizing Information In this thesis, we proposed a timeline generation model which anchors all events related to a specific entity to the time axis. Since the document set dealt with in our study was small, the outputted timelines were readable. However, to apply the model to a large amount of Web text, it is necessary to organize information in a reader-friendly manner. One way is to select events and their relations that users are interested in. It will be interesting to focus not only on events many people mention, but also events with different opinions and events whose

evaluation has changed. Another way is to focus on the attributes of the entity. In our timeline task, the target entity was limited to one of person, organization, product, and financial entity. These entities were dealt with the same way, but storylines of a person and that of a product are totally different. A person has parents, lives in a specific land, earns a livelihood, and has hobbies. By focusing on such attributes of entities, it is possible to provide structured and reader-friendly storylines.

Bibliography

[1] James Allan, Rahul Gupta, and Vikas Khandelwal. 2001. Temporal Summaries of New Topics. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '01, pages 10–18, New York, NY, USA. ACM.

[2] James F. Allen. 1983. Maintaining knowledge about temporal intervals. *COMMUNICATION OF ACM*, pages 832–843.

[3] Gabor Angeli, Christopher D. Manning, and Daniel Jurafsky. 2012. Parsing Time: Learning to Interpret Time Expressions. In *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 3-8, 2012, Montréal, Canada*, pages 446–455.

[4] Gabor Angeli and Jakob Uszkoreit. 2013. Language-Independent Discriminative Parsing of Temporal Expressions. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, ACL 2013, 4-9 August 2013, Sofia, Bulgaria, Volume 1: Long Papers*, pages 83–92.

[5] Masayuki Asahara, Sachi Kato, Hikari Konishi, Mizuho Imada, and Kikuo Maekawa. 2014. BCCWJ-Timebank: Temporal and Event Information Annotation on Japanese Text. In *International Journal of Computational Linguistics & Chinese Language Processing, Volume 19, Number 3, September 2014*.

[6] Yasunobu Asakura, Masatsugu Hangyo, and Mamoru Komachi. 2016. Disaster Analysis using User-Generated Weather Report. In *Proceedings of the 2nd Workshop on Noisy User-generated Text (WNUT)*, pages 24–32, Osaka, Japan. The COLING 2016 Organizing Committee.

[7] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural Machine Translation by Jointly Learning to Align and Translate. *CoRR*, abs/1409.0473.

[8] M Bal. 2009. *Narratology introduction to the theory of narrative, third edition*.

[9] Steven Bethard. 2013. Cleartk-Timeml: A minimalist approach to TempEval 2013. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 10–14. Association for Computational Linguistics.

[10] Steven Bethard, Leon Derczynski, Guergana Savova, James Pustejovsky, and Marc Verhagen. 2015. SemEval-2015 Task 6: Clinical TempEval. In *SemEval@NAACL-HLT*, pages 806–814. The Association for Computer Linguistics.

[11] Steven Bethard, Guergana Savova, Wei-Te Chen, Leon Derczynski, James Pustejovsky, and Marc Verhagen. 2016. SemEval-2016 Task 12: Clinical TempEval. In *SemEval@NAACL-HLT*. The Association for Computer Linguistics.

[12] Steven Bethard, Guergana Savova, Martha Palmer, and James Pustejovsky. 2017. SemEval-2017 Task 12: Clinical TempEval. In *SemEval@ACL*, pages 565–572. Association for Computational Linguistics.

[13] Anders Björkelund, Love Hafdell, and Pierre Nugues. 2009. *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL 2009): Shared Task*, chapter Multilingual Semantic Role Labeling. Association for Computational Linguistics.

[14] 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)*, pages 501–506, Baltimore, Maryland. Association for Computational Linguistics.

[15] Nathanael Chambers, Taylor Cassidy, Bill McDowell, and Steven Bethard. 2014. Dense Event Ordering with a Multi-Pass Architecture. *TACL*, 2:273–284.

[16] Nathanael Chambers and Daniel Jurafsky. 2008. Jointly Combining Implicit Constraints Improves Temporal Ordering. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 698–706. Association for Computational Linguistics.

[17] 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)*. European Language Resources Association (ELRA).

[18] 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)*, pages 1–6. Association for Computational Linguistics.

[19] Fei Cheng and Yusuke Miyao. 2018. Inducing Temporal Relations from Time Anchor Annotation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1833–1843. Association for Computational Linguistics.

[20] Jianpeng Cheng, Siva Reddy, Vijay Saraswat, and Mirella Lapata. 2017. Learning an Executable Neural Semantic Parser. *CoRR*, abs/1711.05066.

[21] Hai Leong Chieu and Yoong Keok Lee. 2004. Query Based Event Extraction Along a Timeline. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '04, pages 425–432, New York, NY, USA. ACM.

[22] Timothy Chklovski and Patrick Pantel. 2004. VerbOcean: Mining the Web for Fine-Grained Semantic Verb Relations. In *Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP-04)*, pages 33–40.

[23] Prafulla Kumar Choubey and Ruihong Huang. 2017. A Sequential Model for Classifying Temporal Relations between Intra-Sentence Events. In *Proceedings*

of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 1797–1803. Association for Computational Linguistics.

[24] Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. 2014. *Empirical evaluation of gated recurrent neural networks on sequence modeling*.

[25] Savelie Cornegruta and Andreas Vlachos. 2016. Timeline extraction using distant supervision and joint inference. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1936–1942. Association for Computational Linguistics.

[26] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *CoRR*, abs/1810.04805.

[27] 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*, EMNLP-CoNLL '12, pages 677–687, Stroudsburg, PA, USA. Association for Computational Linguistics.

[28] Li Dong and Mirella Lapata. 2016. Language to Logical Form with Neural Attention. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*.

[29] Jennifer D'Souza and Vincent Ng. 2013. Classifying Temporal Relations with Rich Linguistic Knowledge. In *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 9-14, 2013, Westin Peachtree Plaza Hotel, Atlanta, Georgia, USA*, pages 918–927.

[30] Christian Girardi Emanuele Pianta and Roberto Zanoli. 2008. The textpro tool suite. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco. European Language Resources Association (ELRA). Http://www.lrec-conf.org/proceedings/lrec2008/.

[31] Michele Filannino, Gavin Brown, and Goran Nenadic. 2013. Mantime: Temporal expression identification and normalization in the TempEval-3 challenge. *CoRR*, abs/1304.7942.

[32] Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2013. PPDB: The paraphrase database. In *Proceedings of NAACL-HLT*, pages 758–764, Atlanta, Georgia. Association for Computational Linguistics.

[33] Andrew F. Hayes and Klaus Krippendorff. 2007. Answering the Call for a Standard Reliability Measure for Coding Data. *Communication Methods and Measures*, 1(1):77–89.

[34] P. Hu, M. Huang, P. Xu, W. Li, A. K. Usadi, and X. Zhu. 2011. Generating Breakpoint-based Timeline Overview for News Topic Retrospection. In *2011 IEEE 11th International Conference on Data Mining*, pages 260–269.

[35] Po Hu, Min-Lie Huang, and Xiao-Yan Zhu. 2014. Exploring the Interactions of Storylines from Informative News Events. *Journal of Computer Science and Technology*, 29(3):502–518.

[36] Ruihong Huang, Ignacio Cases, Dan Jurafsky, Cleo Condoravdi, and Ellen Riloff. 2016. Distinguishing Past, On-going, and Future Events: The Eventstatus Corpus. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 44–54. Association for Computational Linguistics.

[37] T. Joachims. 1998. Making large-Scale SVM Learning Practical. LS8-Report 24, Universität Dortmund, LS VIII-Report.

[38] Daisuke Kawahara, Sadao Kurohashi, and Kôiti Hasida. 2002. Construction of a Japanese Relevance-tagged Corpus. In *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC-2002)*, Las Palmas, Canary Islands - Spain. European Language Resources Association (ELRA).

[39] Remy Kessler, Xavier Tannier, Caroline Hagège, Véronique Moriceau, and André Bittar. 2012. Finding Salient Dates for Building Thematic Timelines. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics:*

Long Papers - Volume 1, ACL '12, pages 730–739, Stroudsburg, PA, USA. Association for Computational Linguistics.

[40] Diederik P. Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *CoRR*, abs/1412.6980.

[41] 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, The 50th annual meeting of the association for computational linguistics, Jeju, Republic of Korea, 8-14 July 2012*, pages 88–97. ACL.

[42] Klaus Krippendorff. 2004. *Content Analysis: An Introduction to Its Methodology (second edition)*. Sage Publications.

[43] Egoitz Laparra, Itziar Aldabe, and German Rigau. 2015. Document Level Time-anchoring for TimeLine Extraction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 358–364. Association for Computational Linguistics.

[44] Heeyoung Lee, Angel Chang, Yves Peirsman, Nathanael Chambers, Mihai Surdeanu, and Dan Jurafsky. 2013. Deterministic coreference resolution based on entity-centric, precision-ranked rules. *Computational Linguistics*, 39(4).

[45] Kenton Lee, Yoav Artzi, Jesse Dodge, and Luke Zettlemoyer. 2014. Context-dependent Semantic Parsing for Time Expressions. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1437–1447, Baltimore, Maryland. Association for Computational Linguistics.

[46] Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. A Structured Self-Attentive Sentence Embedding.

[47] Hector Llorens, Estela Saquete, and Borja Navarro. 2010. TIPSem (English and Spanish): Evaluating CRFs and Semantic Roles in TempEval-2. In *Proceedings of*

the 5th International Workshop on Semantic Evaluation, pages 284–291, Uppsala, Sweden. Association for Computational Linguistics.

[48] Edward Loper and Steven Bird. 2002. NLTk: The Natural Language Toolkit. In *Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics - Volume 1*, ETMTNLP '02, pages 63–70, Stroudsburg, PA, USA. Association for Computational Linguistics.

[49] Kikuo Maekawa. 2008. Balanced Corpus of Contemporary Written Japanese. In *Proceedings of the 6th Workshop on Asian Language Resources*.

[50] 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 the 44th Annual Meeting of the Association for Computational Linguistics*, ACL-44, pages 753–760, Stroudsburg, PA, USA. Association for Computational Linguistics.

[51] Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60.

[52] André F. T. Martins, Noah A. Smith, Eric P. Xing, Pedro M. Q. Aguiar, and Mário A. T. Figueiredo. 2010. Turbo Parsers: Dependency Parsing by Approximate Variational Inference. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, EMNLP 2010, 9-11 October 2010, MIT Stata Center, Massachusetts, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 34–44.

[53] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed Representations of Words and Phrases and Their Compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, pages 3111–3119, USA. Curran Associates Inc.

[54] George A. Miller. 1995. WordNet: A Lexical Database for English. *COMMUNICATIONS OF THE ACM*, 38:39–41.

[55] 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)*, pages 778–786, Denver, Colorado. Association for Computational Linguistics.

[56] Paramita Mirza and Sara Tonelli. 2016. CATENA: CAusal and TEmporal relation extraction from NAtural language texts. In *COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, December 11-16, 2016, Osaka, Japan*, pages 64–75.

[57] Alessandro Moschitti, Siddharth Patwardhan, and Chris Welty. 2013. Long-Distance Time-Event Relation Extraction. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 1330–1338. Asian Federation of Natural Language Processing.

[58] Bilel Moulahi, Jannik Strötgen, Michael Gertz, and Lynda Tamine. 2015. HeidelToul: A Baseline Approach for Cross-document Event Ordering. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 825–829. Association for Computational Linguistics.

[59] Borja Navarro and Estela Saquete. 2015. Gplsiua: Combining temporal information and topic modeling for cross-document event ordering. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 820–824. Association for Computational Linguistics.

[60] Borja Navarro and Estela Saquete. 2016. Cross-document Event Ordering Through Temporal, Lexical and Distributional Knowledge. *Know.-Based Syst.*, 110(C):244–254.

[61] Martha Palmer, Paul Kingsbury, and Daniel Gildea. 2005. The Proposition Bank: An Annotated Corpus of Semantic Roles. *Computational Linguistics*, 31.

[62] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global Vectors for Word Representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

[63] Rashmi Prasad, Nikhil Dinesh, Alan Lee, Eleni Miltsakaki, Livio Robaldo, Aravind Joshi, and Bonnie Webber. 2008. The Penn Discourse TreeBank 2.0. In *In Proceedings of LREC*.

[64] James Pustejovsky, Patrick Hanks, Roser Saurí, Andrew See, Robert Gaizauskas, Andrea Setzer, Dragomir Radev, David Day Beth Sundheim, Lisa Ferro, and Marcia Lazo. 2003. The TIMEBANK Corpus. In *Proc. Corpus Linguistics 2003*, pages 647–656.

[65] James Pustejovsky, Robert Ingria, Roser Saur, Jos Castao, Jessica Moszkowicz, and Graham Katz. 2005. The Specification Language TimeML.

[66] James Pustejovsky and Amber Stubbs. 2011. Increasing Informativeness in Temporal Annotation. In *Proceedings of the 5th Linguistic Annotation Workshop*, LAW V '11, pages 152–160, Stroudsburg, PA, USA. Association for Computational Linguistics.

[67] James Pustejovsky, Marc Verhagen, Roser Saur, Jessica Moszkowicz, Rob Gaizauskas, Graham Katz, Inderjeet Mani, Robert Knippen, and Andrea Setzer. 2006. *TimeBank 1.2*.

[68] Maxim Rabinovich, Mitchell Stern, and Dan Klein. 2017. Abstract Syntax Networks for Code Generation and Semantic Parsing. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1139–1149. Association for Computational Linguistics.

[69] Alec Radford. 2018. Improving Language Understanding by Generative Pre-Training.

[70] 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)*, pages 2195–2204, Berlin, Germany. Association for Computational Linguistics.

[71] Matthew Richardson and Pedro Domingos. 2006. Markov Logic Networks. *Mach. Learn.*, 62(1-2):107–136.

[72] Roser Sauri, Jessica Littman, Robert Gaizauskas, Andrea Setzer, and James Pustejovsky. 2006. TimeML Annotation Guidelines, Version 1.2.1.

[73] Richard Socher, Jeffrey Pennington, Eric H. Huang, Andrew Y. Ng, and Christopher D. Manning. 2011. Semi-supervised Recursive Autoencoders for Predicting Sentiment Distributions. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 151–161, Stroudsburg, PA, USA. Association for Computational Linguistics.

[74] Robert Speer and Catherine Havasi. 2012. Representing General Relational Knowledge in ConceptNet 5. In *LREC*, pages 3679–3686. European Language Resources Association (ELRA).

[75] Mark Steedman. 1996. *Surface Structure and Interpretation*. MIT Press, Cambridge, MA, USA.

[76] Mark Steedman. 2000. *The Syntactic Process*. MIT Press, Cambridge, MA, USA.

[77] Jannik Strötgen and Michael Gertz. 2013. Multilingual and Cross-domain Temporal Tagging. *Language Resources and Evaluation*, 47(2):269–298.

[78] William F. Styler IV, Steven Bethard, Sean Finan, Martha Palmer, Sameer Pradhan, Piet C. de Groen, Bradley James Erickson, Timothy A. Miller, Chen Lin, Guergana K. Savova, and James Pustejovsky. 2014. Temporal Annotation in the Clinical Domain. *TACL*, 2:143–154.

[79] Russell Swan and James Allan. 2000. Automatic Generation of Overview Timelines. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '00, pages 49–56, New York, NY, USA. ACM.

[80] Russell C. Swan and James Allan. 2000. TimeMine: visualizing automatically constructed timelines. In *SIGIR*, page 393.

[81] Seiya Tokui, Kenta Oono, Shohei Hido, and Justin Clayton. 2015. Chainer: a Next-Generation Open Source Framework for Deep Learning. In *Proceedings of Workshop on Machine Learning Systems (LearningSys) in The Twenty-ninth Annual Conference on Neural Information Processing Systems (NIPS)*.

[82] Giang Tran, Mohammad Alrifai, and Eelco Herder. 2015. Timeline Summarization from Relevant Headlines. In *ECIR*.

[83] Giang Binh Tran, Eelco Herder, and Katja Markert. 2015. Joint Graphical Models for Date Selection in Timeline Summarization. In *ACL (1)*, pages 1598–1607. The Association for Computer Linguistics.

[84] Zhaopeng Tu, Zhengdong Lu, Yang Liu, Xiaohua Liu, and Hang Li. 2016. Modeling Coverage for Neural Machine Translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*.

[85] Naushad UzZaman, Hector Llorens, Leon Derczynski, James Allen, and James Pustejovsky. 2013. SemEval-2013 Task 1: TempEval-3: Evaluating Time Expressions, Events, and Temporal Relations. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 1–9. Association for Computational Linguistics.

[86] Marc Verhagen, Rob Gaizauskas, Frank Schilder, M Hepple, Jessica Moszkowicz, and James Pustejovsky. 2009. The tempEval challenge: Identifying temporal relations in text. *Language Resources and Evaluation*, 43:161–179.

[87] Marc Verhagen, Robert Gaizauskas, Frank Schilder, Mark Hepple, Graham Katz, and James Pustejovsky. 2007. SemEval-2007 Task 15: TempEval Temporal Relation Identification. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 75–80. Association for Computational Linguistics.

[88] 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, pages 57–62. Association for Computational Linguistics.

[89] Yannick Versley, Simone Paolo Ponzetto, Massimo Poesio, Vladimir Eidelman, Alan Jern, Jason Smith, Xiaofeng Yang, and Alessandro Moschitti. 2008. BART: A Modular Toolkit for Coreference Resolution. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Demo Session*, HLT-Demonstrations '08, pages 9–12, Stroudsburg, PA, USA. Association for Computational Linguistics.

[90] Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer Networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 2692–2700. Curran Associates, Inc.

[91] Piek Vossen, Tommaso Caselli, and Panagiota Kontzopoulou. 2015. Storylines for structuring massive streams of news. In *Proceedings of the First Workshop on Computing News Storylines*, pages 40–49, Beijing, China.

[92] Yan Wang, Xiaojiang Liu, and Shuming Shi. 2017. Deep Neural Solver for Math Word Problems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 845–854. Association for Computational Linguistics.

[93] Kun Xu, Yansong Feng, Songfang Huang, and Dongyan Zhao. 2015. Semantic Relation Classification via Convolutional Neural Networks with Simple Negative Sampling. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 536–540, Lisbon, Portugal. Association for Computational Linguistics.

[94] Yan Xu, Lili Mou, Ge Li, Yunchuan Chen, Hao Peng, and Zhi Jin. 2015. Classifying Relations via Long Short Term Memory Networks along Shortest Dependency Paths. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1785–1794, Lisbon, Portugal. Association for Computational Linguistics.

[95] Rui Yan, Liang Kong, Congrui Huang, Xiaojun Wan, Xiaoming Li, and Yan Zhang. 2011. Timeline Generation Through Evolutionary Trans-temporal Summarization. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 433–443, Stroudsburg, PA, USA. Association for Computational Linguistics.

[96] Rui Yan, Xiaojun Wan, Jahna Otterbacher, Liang Kong, Xiaoming Li, and Yan Zhang. 2011. Evolutionary timeline summarization: a balanced optimization framework via iterative substitution. In *SIGIR*, pages 745–754. ACM.

[97] Pengcheng Yin and Graham Neubig. 2017. A Syntactic Neural Model for General-Purpose Code Generation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 440–450. Association for Computational Linguistics.

[98] Katsumasa Yoshikawa, Sebastian Riedel, Masayuki Asahara, and Yuji Matsumoto. 2009. Jointly Identifying Temporal Relations with Markov Logic. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1 - Volume 1*, ACL '09, pages 405–413, Stroudsburg, PA, USA. Association for Computational Linguistics.

[99] Xin Wayne Zhao, Yanwei Guo, Rui Yan, Yulan He, and Xiaoming Li. 2013. Timeline Generation with Social Attention. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '13, pages 1061–1064, New York, NY, USA. ACM.

[100] Victor Zhong, Caiming Xiong, and Richard Socher. 2017. Seq2SQL: Generating Structured Queries from Natural Language using Reinforcement Learning. *CoRR*, abs/1709.00103.

[101] Xiaoshi Zhong and Erik Cambria. 2018. Time Expression Recognition Using a Constituent-based Tagging Scheme. In *WWW*, pages 983–992. ACM.

[102] Xiaoshi Zhong, Aixin Sun, and Erik Cambria. 2017. Time Expression Analysis and Recognition Using Syntactic Token Types and General Heuristic Rules. In

Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 420–429. Association for Computational Linguistics.

List of Publications

- Tomohiro Sakaguchi and Sadao Kurohashi. KYOTO at the NTCIR-12 Temporalia Task: MachineLearning Approach for Temporal Intent Disambiguation Subtask. In *Proceedings of the 12th NTCIR Workshop Meeting on Evaluation of Information Access Technologies*, pages 288–292, NTCIR-12, 2016.

- Tomohiro Sakaguchi and Sadao Kurohashi. Timeline Generation based on a Two-stage Event-time Anchoring Model. In *Proceedings of the 18th International Conference on Intelligent Text Processing and Computational Linguistics*, 2017.

- Tomohide Shibata, Hongkai Li, Tomohiro Sakaguchi and Sadao Kurohashi. KYOTOU at TAC KBP 2017 Event track: Neural Network-based Event Sequence Classification Model. In *Proceedings of the Tenth Text Analysis Conference*, 2017.

- Tomohiro Sakaguchi, Daisuke Kawahara and Sadao Kurohashi. Comprehensive Annotation of Various Types of Temporal Information on the Time Axis. In *Proceedings of the 11th Edition of its Language Resources and Evaluation Conference*, pages 332–338, 2018.

- 坂口 智洋, 河原 大輔, 黒橋 禎夫. 事象に対する網羅的な時間情報アノテーションとその分析. 自然言語処理, 26(1), 2019.

List of Other Publications

- 浅原 正幸, 坂口 智洋, 渡邊 友香. 「現代日本語書き言葉均衡コーパス」に対する時間情報表現アノテーションの再修正作業. 第 8 回コーパス日本語学ワークショップ, pages 37–46, 2015.

- 坂口 智洋, 河原 大輔, 黒橋 禎夫. 京都大学テキストコーパスに対する網羅的な時間情報アノテーション. 情報処理学会 第 233 回自然言語処理研究会, 2017.

- 齋藤 純, 坂口 智洋, 柴田 知秀, 河原 大輔, 黒橋 禎夫. 述語項構造に基づく言語情報の基本単位のデザインと可視化. 言語処理学会 第 24 回年次大会, pages 93–96, 2018.