Studies on Fundamental Problems in Event-Level Language Analysis

Hirokazu Kiyomaru

February 2022

Abstract

Natural language processing (NLP) is a research field that aims to make computers understand natural language. NLP has many crucial applications, including machine translation, automatic summarization, and dialogue systems. In recent years, the performance of many NLP applications has significantly improved with the advent of end-to-end learning based on a deep neural network. In end-to-end learning, a problem is solved by optimizing a deep neural network to transform an input into the corresponding output. This simple learning framework works surprisingly well, and the current state-of-the-art models are now performing as well as, or better than, humans in some language understanding tasks. However, some NLP tasks cannot be solved by end-to-end learning; the most representative and crucial example of such a task is exploratory text analysis. The goal of exploratory text analysis is to find valuable information about one's interests from textual data. Because the criteria for determining the value of information vary depending on the purpose, each analysis has an entirely different goal, making end-to-end learning inapplicable. What can be done by NLP to support exploratory text analysis is to organize text at a granularity that is understandable to humans. Such language analysis is collectively called *structural language analysis*.

This thesis focuses on structural language analysis, particularly at the event level. Events, which have long been one of the main interests of NLP, are informative yet convenient information units. Event-level language analysis can be divided into tasks that take events as input and those that take events as output; the former can be further divided into tasks for predicting the properties of events and those for recognizing the relations between events. We refer to the task

of predicting event properties as *event classification*, the task of predicting relations between events as *event-to-event relation analysis*, and the task of predicting events as *event prediction*.

This thesis investigates each of the three types of tasks listed above. First, we study *volitionality classification* as an event classification task in which where models are required to recognize volitionality, a fundamental event property that indicates whether someone is volitionally involved in the event. Despite its importance, volitionality classification has not been studied so actively. As a result, there was no readily available volitionality classifier or no dataset for training volitionality classifiers, making it difficult to employ volitionality classification for downstream tasks. To solve this problem, we propose a minimally-supervised method to learn volitionality classifiers.

Second, we study *discourse relation analysis* as an event-to-event relation analysis task: this is the task of recognizing the pragmatic relation between two events. Since neural networks were introduced to solve this task, researchers have devoted considerable effort to exploring good neural network architectures and effective language resources that facilitate discourse relation analysis. However, recently, it was recently found that general-purpose language models pretrained on raw text achieve greatly improved performance, although they do not use the above techniques. Consequently, we propose a novel self-supervised pretraining framework to learn event representations that are effective in capturing discourse relations.

Finally, we tackle *next event prediction* as an event prediction task: this is the task of predicting events that are likely to happen after a given event. Recently, with the advance of deep learning techniques, this task has been formulated as a generation task. Previous studies have employed simple sequence-to-sequence methods to learn next event prediction. However, such methods are inherently deterministic and hardly capture one-to-many relations. In order to consider one-to-many relations, we propose the use of a probabilistic generation model to learn event prediction.

Acknowledgments

京都大学で過ごした 5 年間は私の人生の中で最も充実した 5 年間でした．その充実した時間を支えてくださった方々に感謝の意を表します．

まず初めに，修士課程に入学してから 5 年間ご指導いただきました黒橋禎夫教授にお礼申し上げます．黒橋先生は，常に的確かつ建設的な助言によって私の研究を導いてくださりました．頂いた助言の数々 — そして研究報告の合間に挟まれる雑談 — は，研究に留まらない，あらゆる問題解決に通ずる示唆に溢れており，一生ものの学びとなりました．また，企業との共同研究に携わらせていただいたり，社会人向け講座の講師・アシスタントに採用していただいたりと，研究を通して培った知識・技術を社会に還元する機会を与えてくださりました．この経験は，自然言語処理研究の社会的意義を，教科書的な知識から確かな実感に変え，私の視野を大きく広げてくれました．

河原達也教授と鹿島久嗣教授には，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ことに感謝いたします．

研究室の現在と過去の方々には，様々な形でご支援をいただきましたことにお礼を申し上げます．河原大輔教授と村脇有吾講師は，黒橋先生の大所高所からの助言を補完するように，研究のきめ細やかな軌道修正をしてくださりました．Chenhui Chu 特定准教授，田中リベカ特任講師，Fei Cheng 特任助教，岡久太郎特定研究員には，研究報告の場面で多くの有益な助言を頂きました．先輩の岸本裕大さん，Yin-Jou Huang 特定研究員，Tareq Alkhaldi さんは，研究に関する議論にしばしば付き合ってくださり，また，執筆中の論文にコメントをお願いすると，いつも快く引き受けてくださりました．後輩の児玉貴志くん，Qianying Liu さん，Haiyue Song くん，植田暢大くん，大村和正くん，Zhuoyuan Mao くん，吉越卓見くん，清水周一郎くんには，研究の議論や雑談に日常的につきあっていただきました．後輩の成長はよく実感できるもので，その成長速度に突き上げられるようにして，私も

ますます研究に打ち込むようになりました．卒業生の栗田修平さん，坂口智洋さん，Arseny Tolmachev さん，大谷直樹さんは，折に触れて連絡をくださりました．先輩方が博士課程で研究に取り組む姿を見て，私はこの研究室で博士課程に進学することに決めました．その先輩方の活躍を見聞きするのは，私にとって非常に刺激的でした．秘書の吉利菜帆さん，石田幸美さん，小杉照美さんには，事務的手続き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．

最後に，長い学生生活を支えてくれた家族に感謝し，謝辞を終えたいと思います．

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Natural language is the basis of intellectual activity. Natural languages constitute a fundamental tool for thinking, communication, and recording. All these activities play a central role in intellectual activity, and would be severely limited in the absence of natural language.

Natural language processing (NLP) is a research field that aims to make computers understand natural language. Because of the importance of natural language, NLP has always been an important research field. Since the development of the Internet, the amount of information transmitted and accumulated as textual data on a daily basis has increased dramatically. The unprecedented scale of textual data further increases the importance of NLP.

NLP has several important applications. For example, machine translation is the key application of NLP that breaks the language barrier and helps people to communicate with foreigners. Automatic summarization reduces the effort required to obtain an overview of the contents of a long text. Dialogue systems provide manual-free interfaces for products and applications. Other practical applications include information retrieval, question answering, and grammatical error correction. This wide range of applications demonstrates how crucial natural language is to our lives.

In recent years, the performance of many NLP applications has greatly im-

proved with the advent of end-to-end learning based on deep neural networks. In end-to-end learning, a problem is solved by optimizing a deep neural network to transform an input to the corresponding output. For example, in Japanese-to-English translation, a deep neural network that is given a Japanese sentence and generates its English translation is trained; in automatic summarization, a deep neural network that is given a document and generates its summary is trained.

This simple framework works surprisingly well. Deep neural networks trained by end-to-end learning have outperformed models based on carefully designed linguistic features and a pipeline of fundamental language analysis, provided that a sufficient amount of training data is available. With the development of high-quality, large-scale datasets and the selection of appropriate neural network architectures, current state-of-the-art models now perform as well as, or better than, humans in some language understanding tasks.

However, some NLP tasks cannot be solved by end-to-end learning; the most representative and important example of such a task is exploratory text analysis, typified by customer feedback analysis. The goal of exploratory text analysis is to find valuable information about one ' s interests from textual data. Because the criteria for determining the value of information vary depending on the purpose, each analysis has an entirely different goal.

End-to-end learning is inherently inapplicable to exploratory text analysis because each analysis has a unique goal. Let us consider performing exploratory text analysis by end-to-end learning. Because each analysis has a unique goal, one would need to start by creating training data that includes the valuable information that one wishes to find from the analysis. This means that, if one were to attempt to perform exploratory text analysis by end-to-end learning, the goal would be achieved before end-to-end learning is performed.

What can be done by NLP to support exploratory text analysis is to organize text at a granularity that is understandable to humans, for example, in terms of words, phrases, clauses, and sentences; such language analysis is collectively called *structural language analysis*. Structural language analysis provides basic information that is typically required in language understanding. Therefore, most language understanding problems can be solved by extracting the necessary infor-

mation from the result of structural language analysis and implementing purpose-specific processing. This flexibility works effectively in supporting exploratory text analysis.

This thesis focuses on structural language analysis, particularly at the clause level. A clause is a sub-sentential linguistic unit that consists of one main predicate and its arguments. For example, the sentence in example (1) includes one clause, whereas the sentence shown in example (2) includes two clauses.

(1) [I had dinner at a recently opened restaurant.]
1st clause

(2) [The ambience was nice,] but [the service was not so great.]
1st clause 2nd clause

A clause generally represents a single event. In the following, we refer to a clause-level information unit as an *event*, and structural language analysis performed at the event level as *event-level language analysis*.

Events, which have long been one of the main interests of NLP, are informative yet convenient information units. Compared with words and phrases, events are much more informative, enabling high-level text analyses such as discourse relation analysis. In addition, compared with larger linguistic units, such as sentences, paragraphs, and documents, events are often more convenient in downstream tasks. Consider the case where we wish to apply sentiment analysis to the sentence in example (2). Because the first event (clause) is positive and the second event (clause) is negative, the overall sentiment is hard to determine. In exploratory text analysis such as customer feedback analysis, instead of treating example (2) as a sentence of unknown sentiment polarity, it is often more convenient to treat it as a sentence with one positive event and one negative event.

Event-level language analysis can be divided into tasks that take events as input and those that take events as output; the former can be further divided into tasks for predicting the properties of events and those for recognizing the relations between events. In this thesis, we refer to the task of predicting event properties as *event classification*, the task of predicting relations between events as *event-to-event relation analysis*, and the task of predicting events as *event prediction*. Figure 1.1 illustrates this categorization. All these tasks play an important role

(a) An example of *event classification*, where models are given an event or an event sequence and determine whether it has a particular property. In this example, the model is required to predict the volitionality of the given event (*volitionality classification*).

(b) An example of *event-to-event relation analysis*, where models are given two events and predict the relation between them. In this example, the model is to predict the pragmatic relation between the given two events (*discourse relation analysis*).

(c) An example of *event prediction*, where models are generally conditioned by events and predict events that have a certain relation to the given events. In this example, the model is to predict an event that is likely to happen after the given event (*next event prediction*).

Figure 1.1: A general categorization of event-level language analysis.

in exploratory text analysis. Event classification allows users to browse events possessing a particular property. Event-to-event relation analysis transforms a list of events to a structure linking events according to their relations, enabling in-depth text analysis. Event prediction helps users to find insights by predicting events that are not explicitly written in texts.

1.2 Outline of the Thesis

To conclude the introduction, we present the outline of this thesis. In the thesis, we tackle three fundamental tasks in event-level language analysis: *volitionality*

classification, *discourse relation analysis*, and *next event prediction*. Figure 1.1 shows an example of each task. Volitionality classification, discourse relation analysis, and next-event prediction are crucial tasks of event classification, event-to-event relation analysis, and event prediction, respectively.

In Chapter 2, we first give an overview of event-level language analysis. In this chapter, we begin by providing an introduction to event representations and event-to-event relations, as preliminaries to event-level language analysis. We then present an overview of event-level language analysis, according to the categorization described in Chapter 1.

In Chapter 3, we present our work on volitionality classification. Volitionality is a fundamental property of an event that indicates whether someone is volitionally involved in the event. Despite its importance and wide range of applications, volitionality classification has not been studied as actively as it should be. As a result, there was no readily available volitionality classifier or no dataset for training volitionality classifiers, making it difficult to employ volitionality classification for downstream tasks. To solve this problem, we propose a minimally-supervised method to learn volitionality classifiers.

In Chapter 4, we introduce our work on discourse relation analysis. In discourse relation analysis, models are required to recognize the pragmatic relation between two events. The accuracy of discourse relation analysis has been dramatically improved since deep learning was introduced. Researchers have devoted considerable effort to exploring good neural network architectures and effective language resources that facilitate discourse relation analysis. However, recently, it was recently found that general-purpose language models pretrained on raw text greatly improve the performance, without using the above techniques. This means that it is possible to learn event representations that capture discourse relations from raw text, indicating a promising direction for improving discourse relation analysis. For this reason, we propose a novel self-supervised framework to learn event representations that are effective in capturing discourse relations.

In Chapter 5, we present our work on next event prediction. In next event prediction, models are required to predict events that are likely to happen after a given event. Although next event prediction was traditionally treated as a

classification task, with the advance of deep learning techniques, it has recently been formulated as a generation task. Previous studies have employed simple sequence-to-sequence methods to learn event prediction; however, such methods are inherently deterministic and hardly capture one-to-many relations. In order to consider one-to-many relations, we propose the use of a probabilistic generation model to learn event prediction. In addition, we present a new evaluation dataset that we have constructed to fairly evaluate diversity-aware models.

In Chapter 6, we present the overall conclusion of this thesis. Here, we review the entire thesis and discuss the future prospects of our research.

Chapter 2

Event-level Language Analysis

In this chapter, we give an overview of event-level language analysis. As preliminaries to it, we first provide a detailed introduction to event representations and event-to-event relations.

2.1 Event Representation

An event is a fundamental information unit representing a single action or state. Events have been one of the main focuses of NLP as they are practically useful in downstream tasks. Events have been represented in several ways, depending on the purpose. In the following, we introduce three representative event representations: *syntactic structural representation*, *semantic structural representation*, and *textual representation*.

2.1.1 Syntactic Structural Representation

Syntactic structural representation represents an event by a predicate-argument structure (PAS) (Chambers and Jurafsky, 2008; Jans et al., 2012; Pichotta and Mooney, 2014; Granroth-Wilding and Clark, 2016; Shibata et al., 2014). A PAS is a good approximation of who did what to whom, consisting of a predicate and its syntactic arguments, including the subject, direct object, indirect object, etc. Figure 2.1 shows an event represented by a PAS.

Jim gave the book to Mary.

Predicate	give
Subject	Jim
Direct object	book
Indirect object	Mary

Figure 2.1: An example of syntactic structural representation.

A great advantage of employing syntactic structural representation is that they can be used regardless of topic or domain of the text of interest, thanks to the fact that syntactic dependency parsing targets any grammatically valid sentences. Therefore, syntactic structural representations are traditionally employed to extract events from a large amount of texts in which a variety of events appear (Chambers and Jurafsky, 2008; Jans et al., 2012; Pichotta and Mooney, 2014; Granroth-Wilding and Clark, 2016; Shibata et al., 2014).

Another advantage is that the cost of event extraction is generally small. Syntactic dependency parsing has long been studied, and thus there generally exist readily available syntactic dependency parsers. Therefore, in event extraction, one only needs to implement the process of formatting the output of a syntactic dependency parser.

One crucial disadvantage of using syntactic structural representation is that semantically identical events do not always have the same syntactic structure, resulting in different representations. Let us consider the following events, for example.

(3) a. I like apples.
b. Apples are my favorite.

These events are semantically identical, although there are slight differences in nuance; however, since they have different syntactic structures, they are encoded into different syntactic structural representations. In order to solve this problem, one needs to consider an event representation based on semantic relations rather than syntactic relations, motivating the use of semantic structural representation,

introduced next.

Another disadvantage is that information that does not fit into the structure (e.g., adverbs, word order, etc.) is discarded. Consider the following event, for example.

(4) I tumbled deliberately.

As long as syntactic structural representation is employed, the adverb *deliberately* will be lost from the event representation. Such dropped information may play an essential role in event-level language analysis. For example, in this case, the volitionality of the event will no longer be correctly recognized without the adverb *deliberately*, because "I tumbled deliberately" is volitional while "I tumbled" is non-volitional.

2.1.2 Semantic Structural Representation

Semantic structural representation is a structural representation based on the semantic roles of the participants of events. The design of semantic structural representation has arbitrariness in how finely it classifies the semantic relationship between an event and its participants. If one employs a coarse relation set, a variety of events can be represented with it, but the meaning of events is not captured very precisely. On the contrary, if one employs a fine-grained relation set, the meaning of events can be captured precisely, but one needs to restrict the types of events to handle due to practical constraints on the annotation cost.

Abstract meaning representation (AMR) (Banarescu et al., 2013) is a semantic structural representation based on a relatively coarse semantic relation set. AMR represents a sentence as a rooted and labeled graph, where each node corresponds to a concept in the sentence and is linked to the others with semantic role labels. Figure 2.2 shows events represented as an AMR. Thanks to that AMRs abstract away from syntax, the shown events are encoded into the same representation.

AMR concepts can be either English words (e.g., "boy"), PropBank's framesets (e.g., "buy-01") (Kingsbury and Palmer, 2002), or special keywords, including logical conjunctions (e.g., "and"). A node representing a PropBank frameset corresponds to a predicate in syntactic representation. A PropBank frameset is a

- *The man described the mission as a disaster.*
- *As the man described it, the mission was a disaster.*

Frameset	describe-01
:arg0	man
:arg1	mission
:arg2	disaster

Figure 2.2: An example of abstract meaning representation (AMR). AMRs abstract away from syntax, and thus the displayed events are encoded into the same representation.

list of arguments required by an English verb with their semantic roles. Semantic roles are numbered sequentially from :arg0 up to :arg5 and each of these is given a verb-specific mnemonic label, although there is a general rule that :arg0 refers to the subject, :arg1 refers to the direct object, :arg2 refers to the indirect object, etc. For example, a PropBank frameset "buy-01" is expected to has five arguments with the following semantic roles: :arg0 (buyer), :arg1 (thing bought), :arg2 (seller), :arg3 (price paid), and :arg4 (benefactive). AMR considers approximately 100 semantic relations, including the frame arguments:

- **Frame arguments**: :arg0, :arg1, :arg2, :arg3, :arg4, and :arg5.
- **General semantic relations**: :accompanier, :age, :beneficiary, :cause, :compared-to, :concession, :condition, :consist-of, :degree, :destination, etc.
- **Relations for quantities**: :quant, :unit, and :scale.
- **Relations for date-entities**: :day, :month, :year, :weekday, :time, :timezone, :quarter, :dayperiod, :season, :year2, :decade, :century, :calendar, and :era.
- **Relations for lists**: :op1, :op2, :op3, :op4, :op5, :op6, :op7, :op8, :op9, and :op10.

In AMR, information that does not fit into the above relationships is discarded. For example, AMR ignores inflectional morphology for tense and number. There-

The terrorists set off a bomb attack.

Type	Conflict / Attack
Trigger	attack
Argument-Attacker	the terrorists
Argument-Target	-
Argument-Instrument	bomb
Argument-Time	-
Argument-Place	-

Figure 2.3: An example of semantic structural representation in the ACE.

fore, events with different details may have the same representation. Besides, AMR relies heavily on PropBank, and as a result, it is heavily biased towards English.

There also exist representations that focus on specific event types and instead consider fine-grained semantic roles of event participants. The most representative one is the ACE (Automatic Content Extraction) (Doddington et al., 2004), whose annotation scheme is often employed in subsequently constructed datasets such as TAC-KBP (Zhang et al., 2017). An event is represented as a structure with the following information:

- **Event mention**: a phrase or a sentence in which the event is described.
- **Event trigger**: the word that most clearly describes the occurrence of the event.
- **Event type**: the semantic class of the event.
- **Event argument**: the entity that serves as a participant or attribute with a specific semantic role in the event.
- **Argument role**: the relationship between an argument and the event.

Figure 2.3 shows an example. The trigger word is *attack*. The event type is Conflict/Attack. The arguments in this type include Attacker, Target, Instrument, Time, etc. Such slots are defined for each event type. As can be

seen in the argument of TARGET, the slot values are kept empty when they have not been mentioned in the text.

In ACE, there are eight event types, including CONFLICT, and each of them has its sub-types, such as CONFLICT/ATTACK. Each event type and sub-type has its own set of participant roles. In return for limiting the types of events, the participant roles are designed to be detailed and exhaustive.

The other notable semantic structural representations include ERE (Aguilar et al., 2014) and FrameNet (Baker et al., 1998). ERE was created as a lightweight alternative to ACE, aiming at making annotation easier and more consistent. FrameNet prioritizes lexicographic and linguistic completeness over ease of annotation, which results in a much finer-grained annotation scheme. Aguilar et al. (2014) provide a detailed comparison of these semantic structural representations.

These are two major advantages of using semantic structural representation. First, unlike syntactic structural representation, semantically identical events are encoded into the same representation, though events with minor differences may be encoded into the same representation. This facilitates statistical analyses of events. Second, semantic structural representation is flexible in terms of that semantic frames can be freely designed considering the information required for downstream tasks.

On the other hand, the disadvantage lies in that the semantic structural representations need to be designed carefully according to the downstream tasks and domains. Existing annotation schemas, such as ACE, ERE, and FrameNet, do not define the semantic frame of every event type, nor do they assume every downstream task. In order to employ a semantic structure representation for a task of interest, one certainly needs to start by defining an appropriate typology of events and the semantic frame of each event type. It requires a deep understanding and insight into both the task and language.

Along with this nature, one also certainly needs to construct annotated data to train an event extractor. This is also a crucial drawback because it takes a long time and much money to build annotated data with sufficient quantity and quality.

For this reason, semantic structural representations are often employed in

research that focuses on specific types of events and concentrates on an in-depth analysis within the scope.

2.1.3 Span Representation

Span Representation is a natural language text that consists of one main predicate and its arguments, corresponding to a clause (Hu et al., 2017; Nguyen et al., 2017; Prasad et al., 2018; Kawahara et al., 2014; Inui et al., 2003; Sap et al., 2019; Saito et al., 2019). In terms of that span representation is constructed according to the syntactic structure of a text, it is close to syntactic structural representation. The crucial difference from syntactic structural representation is that span representation does not have an internal structure.

There are three major advantages of adopting span representation. First, as well as syntactic structural representation, events can be extracted using a readily available syntactic dependency parser. Because many languages have a readily available syntactic dependency parser, one does not need to build a new annotated dataset nor train an event extractor for event extraction. Second, span representation is robust to parsing errors compared to structural representation. This is because the dependency inside a text span does not necessarily need to be correctly parsed. This reduces the error propagation caused by event extraction. Third, span representation is more informative than syntactic structural representation. As span representation does not have an internal structure, it can naturally include any modifiers such as adverbs and preserve word order, which have been discarded in syntactic structural representation. This allows a more high-level semantic analysis of events.

The main drawback lies in the difficulty of applying statistical methods. We mentioned that, in syntactic structure representation, semantically identical events are not always given the same representation. This problem is even more severe in span representation. For example, if one tries to count the frequency of events, almost every event will be unique due to the high flexibility of the representation, resulting in meaningless results. This problem was fatal when semantic analysis of texts was immature, and we had to rely heavily on superficial and structural information. This is the reason why structural representations have been traditionally

employed.

However, with the development of text analysis techniques, this drawback is being overcome. The key idea is to represent events in a continuous vector space in which semantically similar events are encoded in neighborhoods. In particular, an increasing number of techniques have been recently developed to provide powerful vector representations of raw text, and the use of span representation has the advantage of directly benefiting from such techniques. Accordingly, there is a growing body of work employing span representation.

2.2 Event-to-event Relation

A text is usually organized with multiple events. In order to understand the meaning of the text, it is necessary to understand not only the meaning of each event but also the relations between the events. Roughly speaking, event-to-event relations can be categorized into syntactic and semantic relations. This section introduces event-to-event relations that play an important role in event-level language analysis.

2.2.1 Syntactic Relation

A syntactic relation is a relation between two events (clauses) with a syntactic dependency. Syntactic relations are essential to combine events to construct a larger information unit. Information that a single event can convey is limited. By using syntactic relations, one can create larger information units with a granularity appropriate for downstream tasks. For example, in exploratory text analysis, syntactic relations are effectively used to present the details of an event by showing its syntactically dependent events.

Coordination

A coordination relation is a syntactic relation where two events serve as equivalent grammatical elements. Example (5) shows two events in a coordination relation.

(5) [I am majoring in computer science,] and [he is majoring in chemistry.]
1st event 2nd event

Subordination

A subordination relation is a syntactic relation where one event modifies the other. In linguistics, the head event is called the main clause, and the modifier event is called the dependent clause.

Example (6) shows two events in an adverbial subordination relation.

(6) [When I was in school,] [I liked to draw.]
Dependent clause Main clause

Example (7) shows two events in an adjectival subordination relation. The dependent clause in example (7) is also called the relative clause.

(7) [I broke the computer] [that I bought yesterday.]
Main clause Dependent clause

Example (8) shows two events in a subordination relation where the dependent clause is used as the complement of the direct object of the verb in the main clause, *expected*. The dependent clause in example (8) is also called the complement clause.

(8) [He never expected] [that I could pass the exam.]
Main clause Dependent clause

2.2.2 Semantic Relation

A semantic relation is a relation between two events with a semantic connection. There is a wide range of semantic relations considered important in language understanding. We introduce discourse, temporal, and causal relations as actively studied semantic relations.

Discourse Relation

Discourse relation is a pragmatic relation between events[1]. Discourse relations provide a high-level linguistic structure of a text that helps understand the logi-

[1]In the context of discourse relation analysis, the processing unit is called the elementary discourse unit (EDU). While the definition of EDUs varies depending on the corpus, an EDU generally corresponds to a clause, i.e., an event.

cal flow. Therefore, it is beneficial for tasks that require contextual text understanding, such as document-level machine translation, automatic summarization, information retrieval, etc.

One of the representative corpora with discourse relation labels is the Penn Discourse Treebank (PDTB) (Prasad et al., 2008, 2018). In the PDTB, discourse relation labels are basically assigned to adjacent two events. The PDTB uses 54 discourse relation labels that are organized in a hierarchy consisting of three levels. Table 2.1 lists the discourse relation labels. Level-1 is the top level that contains four major semantic classes. Level-2 further categorizes the major semantic classes into finer-grained classes. Level-3 is the most fine-grained level that considers the direction of asymmetric discourse relations. Examples (9), (10), and (11) show events with their discourse relation labels.

(9) [Pressed on the matter,] [he is more specific.]
1st event (Arg1) 2nd event (Arg2)
Label: TEMPORAL/ASYNCHRONOUS/SUCCESSION[2]

(10) [Walk down the center of the path,] or otherwise, [you might trip and fall.]
1st event (Arg1) 2nd event (Arg2)
Label: CONTINGENCY/NEGATIVE-CONDITION/ARG1-AS-NEGCOND

(11) [That the debt is equity] and therefore, [it isn't deductible.]
1st event (Arg1) 2nd event (Arg2)
Label: CONTINGENCY/CAUSE/RESULT

Another representative corpus with discourse relation labels is the RST Discourse Treebank (RST-DT) (Carlson et al., 2002). The annotation framework is based on the Rhetoric Structure Theory (RST) proposed by Mann and Thompson (1988). In the RST-DT, discourse relation labels are annotated to represent the entire document as a tree structure. The leaf nodes of an RST tree are events. Neighboring nodes are connected according to their discourse relation and form a new node representing a larger discourse unit. By recursively connecting adjacent nodes, an RST tree is eventually obtained. When connecting two nodes, the nodes are categorized into either *nucleus* or *satellite* nodes, indicating their relative importance. Nucleus nodes describe important information, while satel-

[2]The labels indicate, from left to right, the level-1, level-2, and level-3 labels, respectively.

Level-1	Level-2	Level-3
TEMPORAL	SYNCHRONOUS	-
	ASYNCHRONOUS	PRECEDENCE PRECEDENCE
CONTINGENCY	CAUSE	REASON RESULT NEGRESULT
	CAUSE+BELIEF	REASON+BELIEF RESULT+BELIEF
	CAUSE+SPEECHACT	REASON+SPEECHACT RESULT+SPEECHACT
	CONDITION	ARG1-AS-COND ARG2-AS-COND
	CONDITION+SPEECHACT	-
	NEGATIVE-CONDITION	ARG1-AS-NEGCOND ARG2-AS-NEGCOND
	NEGATIVE-CONDITION+SPEECHACT	-
	PURPOSE	ARG1-AS-GOAL ARG2-AS-GOAL
COMPARISON	CONCESSION	ARG1-AS-DENIER ARG2-AS-DENIER
	CONCESSION+SPEECHACT	ARG2-AS-DENIER+SPEECHACT
	CONTRAST	-
	SIMILARITY	-
EXPANSION	CONJUNCTION	-
	DISJUNCTION	-
	EQUIVALENCE	-
	EXCEPTION	ARG1-AS-EXCPT ARG2-AS-EXCPT
	INSTANTIATION	ARG1-AS-INSTANCE ARG2-AS-INSTANCE
	LEVEL-OF-DETAIL	ARG1-AS-DETAIL ARG2-AS-DETAIL
	MANNER	ARG1-AS-MANNER ARG2-AS-MANNER
	SUBSTITUTION	ARG1-AS-SUBST ARG2-AS-SUBST

Table 2.1: Discourse relations and their hierarchy in the PDTB 3.0.

lite units contain supplementary information for a nucleus node. The discourse relations used in the RST-DT can be divided into relations connecting a nucleus node and a satellite node (called *mononuclear relations*) and those connecting two nucleus nodes (called *multinuclear relations*).

Table 2.2 lists the discourse relation labels used in the RST-DT. The RST-DT contains 78 relation types, including 53 mononuclear and 25 multinuclear relations. The first column lists mononuclear relations where the satellite node characterizes the relation. The second column lists mononuclear relations in which the nucleus node characterizes the relation. The third column lists multinuclear relations. Corresponding mononuclear and multinuclear relations are shown across a single row.

Although both PDTB and RST-DT are English corpus, in languages other than English, many annotated corpora follow the manner of PDTB mainly due to the simplicity. For example, Kawahara et al. (2014), Kishimoto et al. (2018), and Kishimoto et al. (2020) constructed a Japanese web corpus with discourse relation labels in the PDTB style. Zhou and Xue (2012) created a PDTB-style Chinese annotated corpus. Zeyrek et al. (2018) built TED-Multilingual Discourse Bank, which is a PDTB-style corpus of TED talks transcripts in six languages, including English, German, Polish, European Portuguese, Russian and Turkish. There also exist non-English corpora that follow the manner of the RST-DT. For example, da Cunha et al. (2011) constructed a Spanish corpus following the RST-DT style. Stede and Neumann (2014) created an RST-DT-style German corpus.

Temporal Relation

Temporal relation describes the order in which events occur. Temporal relation is a kind of discourse relation, as can be seen in Table 2.1 and Table 2.2; however, because of the importance, research focused on temporal relationships has long been conducted, forming a research field.

Temporal relation has been considered crucial information to perform question answering, information extraction, and automatic summarization (Chambers et al., 2007). Suppose we have a collection of reviews about a product, half positive and half negative. This does not necessarily mean that the opinions about

Mononuclear (satellite)	**Mononuclear (nucleus)**	**Multinuclear**
analogy		Analogy
antithesis		Contrast
attribution		
attribution-n		
background		
	cause	Cause-Result
circumstance		
comparison		Comparison
comment		
		Comment-Topic
concession		
conclusion		Conclusion
condition		
consequence-s	consequence-n	Consequence
contingency		
		Contrast (see antithesis)
definition		
		Disjunction
elaboration-additional		
elaboration-set-member		
elaboration-part-whole		
elaboration-process-step		
elaboration-object-attribute		
elaboration-generalspecific		
enablement		
evaluation-s	evaluation-n	Evaluation
evidence		
example		
explanation-argumentative		
hypothetical		
interpretation-s	interpretation-n	Interpretation
		Inverted-Sequence
		List
manner		
means		
otherwise		Otherwise
preference		
problem-solution-s	problem-solution-n	Problem-Solution
		Proportion
purpose		
question-answer-s	question-answer-n	Question-Answer
reason		Reason
restatement		
	result	Cause-Result
rhetorical-question		
		Same-Unit
		Sequence
statement-response-s	statement-response-n	Statement-Response
summary-s	summary-n	
	temporal-before	
temporal-same-time	temporal-same-time	Temporal-Same-Time
	temporal-after	
		TextualOrganization
		Topic-Comment
topic-drift		Topic-Drift
topic-shift		Topic-Shift

Table 2.2: Discourse relations used in the RST-DT.

T: I wanted to plant a tree. I went to the home and garden store and picked a nice oak. Afterwards, I planted it in my garden.

Q1: What was used to dig the hole?
a. a shovel b. his bare hands

Q2: When did he plant the tree?
a. after watering it b. after taking it home

Figure 2.4: Example questions in MCScript.

the product are divided; it may be that the opinions used to be mostly negative, but now they are mostly positive. In order to properly perform question answering, information extraction, and automatic summarization on the reviews, it is essential to take temporal information into account.

Research in this line has been greatly advanced by the creation of the TimeBank corpus (Pustejovsky et al., 2003), which is one of the most popular corpora in use today. The TimeBank corpus is based on TimeML (Pustejovsky et al., 2003), an annotation scheme for time expressions as well as the temporal relations between events. Well-studied categories of temporal relations include Before, After, Includes, Included, Simultaneous, and Vague.

Knowledge organizing stereotypical event sequences is called *scripts* (Schank and Abelson, 1975). Scripts have been considered an important class of commonsense knowledge because it is the basis of inferring events that are not explicitly described in texts. For example, if we observe the event "Bob gets on the bus," we are likely to expect events such as "Bob takes a seat" or "Bob pays the bus fare" to happen subsequently. Such commonsense knowledge is so obvious that it is rarely explicitly mentioned in texts. This phenomenon is called *reporting bias* (Gordon and Van Durme, 2013) and is considered to be one of the factors that make text understanding difficult.

There are several language resources to learn script knowledge. DeScript (Wanzare et al., 2016) is a collection of stereotypical event sequences. DeScript is constructed by crowdsourcing. Crowdworkers are given a scenario and write a stereotypical event sequence in the scenario. DeScript contains 40 scenarios and 100 event sequences for each of the scenarios. MCScript (Ostermann et al., 2018,

2019) is a benchmark dataset of narrative texts and questions about them that requires script knowledge to answer. Figure 2.4 shows two example questions. MCScript contains approximately 20,000 questions on approximately 3,500 texts.

Causal Relation

Causal relation is an important class of semantic relations, describing the cause-effect relation between two events. Causal relation plays an important role in connecting events logically and meaningfully. While causal relation is also a kind of discourse relation, due to its importance, there are many studies focusing on causal relation.

Researchers have developed several language resources with causal relation annotations. The Causal TimeBank (Mirza et al., 2014) is a corpus annotated with the following causal relations: CAUSE, ENABLE and PREVENT. ATOMIC (Sap et al., 2019) is a large-scale language resource consisting of event pairs in a causal relation. Figure 2.5 shows an example[3]. ATOMIC categorizes causal relations into nine types to distinguish causes and effects, agents and themes, voluntary and involuntary events, and actions and mental states.

There exist benchmark datasets to investigate the ability of computational models being able to recognize causal relations. The Choice Of Plausible Alternatives (COPA) (Roemmele et al., 2011) is a collection of English questions to ask commonsense causal reasoning. Each question in COPA is composed of a premise and two alternatives. Models are required to select the alternative that more plausibly has a causal relation with the premise. The Kyoto University Commonsense Inference (KUCI) (Omura et al., 2020) is a Japanese dataset that is similar to COPA; each question consists of a context and four choices, where the task is to select the choice that has the strongest causal relation with the context. In KUCI, to make challenging questions, false choices are chosen so that they are close to the correct choice in some respects.

[3]The figure was generated by the official online browser, available at `https://mosaickg-graph-viz.apps.allenai.org/kg_atomic2020`.

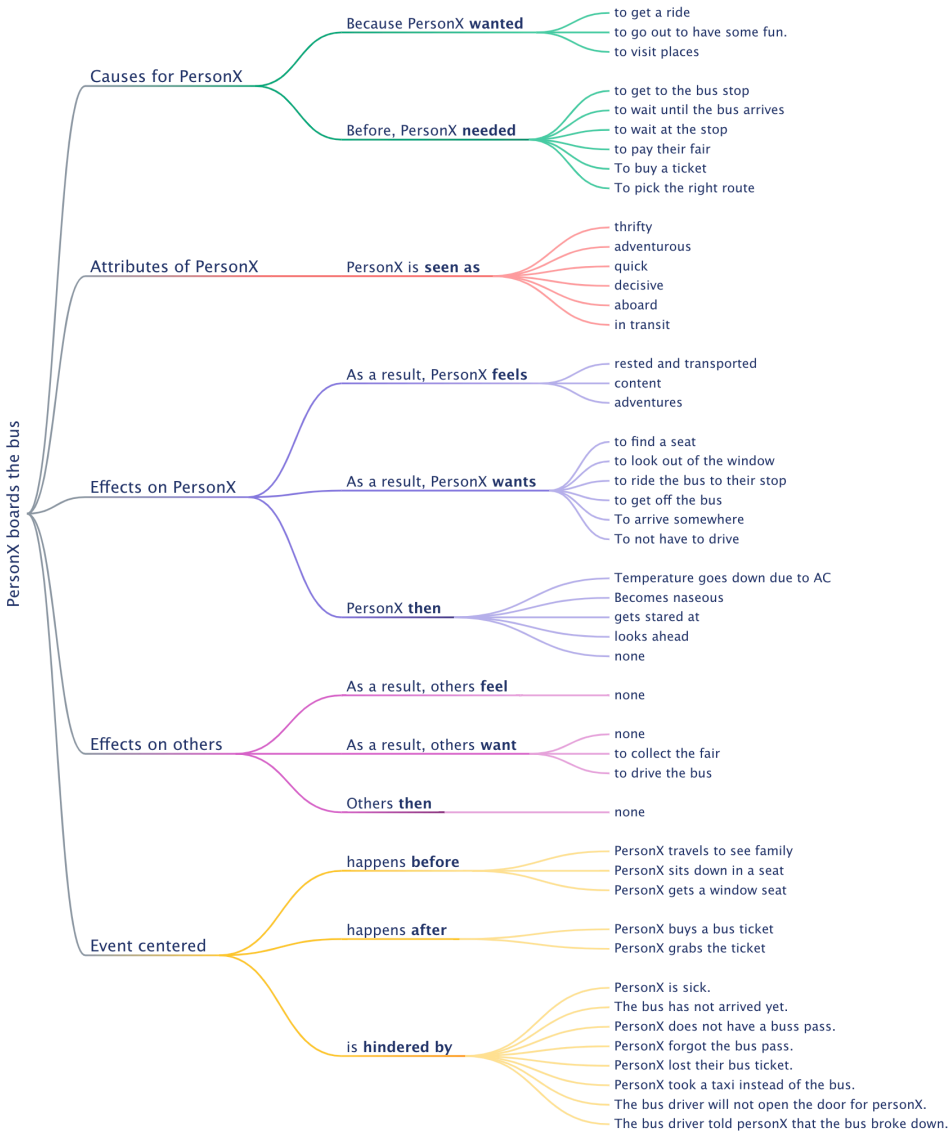

Figure 2.5: An example of ATOMIC, showing categorized events that happen if *a person X boards the bus*.

2.3 Event-Level Language Analysis

There is a large body of work on event-level language analysis. Event-level language analysis begins with extracting events from texts. This task is called *event extraction*. Language analysis tasks performed at the event level can be divided

into *event classification*, *event-to-event relation analysis*, and *event prediction*. Figure 1.1 illustrates the categorization. We describe event-level language analysis tasks according to this categorization in the following.

2.3.1 Event Extraction

Event extraction is the task of extracting events from a given text. Event extraction is the essential first step to performing event-level language analysis.

The formulation of event extraction depends on how events are represented. When employing a syntactic event representation (i.e., syntactic structural representation or span representation), event extraction is performed by syntactic dependency parsing. In this case, no special analysis is required for event extraction, although it is necessary to implement a process to transform the results of syntactic parsing into event representations. For example, aiming at obtaining an event chain, Chambers and Jurafsky (2008) and Jans et al. (2012) obtained events by extracting tuples (v, d) where each v is a verb that has a particular entity as its dependency d. Pichotta and Mooney (2014), Granroth-Wilding and Clark (2016), Ding and Riloff (2018) and many others extracted events as predicate-argument structures (PASs) consisting of a verb and its subject, direct object, and indirect object. Del Corro and Gemulla (2013) extracted events as a span representation, i.e., clauses. In Japanese, Shibata et al. (2014) extracted events as PASs. Saito et al. (2019) and Omura et al. (2020) employed a span representation.

When employing a semantic event representation, event extraction is formulated as an information extraction task. In this case, event extraction is typically performed by a model trained on an annotated corpus, such as the ACE corpus (ace, 2005; Doddington et al., 2004) and the TAC-KBP corpus (Zhang et al., 2017). Event extraction is performed in two steps: trigger extraction and argument extraction. In trigger extraction, a model identifies the trigger word of an event and recognize the event type to determine the semantic frame. In argument extraction, another model finds the arguments of the event and recognizes the role of each argument. Traditionally, these two tasks are performed in a sequential pipeline manner (Ji and Grishman, 2008; Liao and Grishman, 2010; Hong et al., 2011; Chen et al., 2015). The well-known drawback of pipeline methods is

the error propagation problem. To alleviate this problem, recent studies propose to perform trigger extraction and argument extraction jointly (Li et al., 2013; Nguyen et al., 2016). Another interesting approach is to formulate event extraction as a question answering (or reading comprehension) problem, aiming to make the best use of the knowledge embedded in general-purpose pretrained language models, which have been used with great success in recent years (Liu et al., 2020; Du and Cardie, 2020).

2.3.2 Event Classification

Event classification is the task of determining if an event has a particular property. Events have a wide variety of properties, and their recognition is the basis for performing downstream tasks.

The most actively addressed tasks are event type classification (Naughton et al., 2008; Haneczok et al., 2021) and sentiment analysis (Saito et al., 2019; Zhuang et al., 2020). Event type classification is the task of classifying an event into one of the pre-defined types. In the case of using semantic a structural representation, event type classification is necessary to be solved for event extraction because the event type decides the semantic structure, as can be seen in Figure 2.3.

Event-level sentiment analysis is the task of recognizing whether a given event typically affects humans positively or negatively. Because an event consists of a verb and its arguments and can construct a simple sentence, the techniques for sentence-level sentiment analysis can also be applicable in many cases. The most straightforward way to solve sentiment analysis is to train a classifier on an annotated corpus. In sentence-level sentiment analysis, a large annotated corpus, the Stanford Sentiment Treebank (SST) (Socher et al., 2013), is commonly used to train a sentiment analysis classifier. However, human annotation is expensive, especially when one wishes to learn a deep NLP model, which is promising to achieve high performance but requires a huge amount of training data. This prevents us from obtaining a classifier that performs well in a low-resource language or a specific domain. Therefore, several studies propose weakly-supervised methods. In weakly-supervised methods, training data is automatically collected using heuristics, and a model is trained on it. For example, Kaji and Kitsure-

gawa (2006) propose a method of building a collection of sentences with sentiment polarity labels. They automatically gather sentences describing positive or negative opinions, utilizing HTML layout structures in addition to linguistic patterns. Saito et al. (2019) propose to obtain labeled events by exploiting discourse relations that propagate sentiment polarity from seed predicates that report one's emotions.

A task that has not received much attention despite its importance is volitionality classification (Inui et al., 2003; Abe et al., 2008,). Volitionality classification focuses on *volitionality*, an event property that indicates whether someone is volitionally involved in the event. Volitionality is a fundamental event property, and thus the recognition has a wide range of applications. For example, in customer feedback, volitional classification is helpful to extract the voluntary actions of customers. There is a handful of work on volitionality classification. Abe et al. (2008) and Abe et al. (2008) manually build a lexicon of verbs with volitionality labels and classify volitionality by looking it up. Inui et al. (2003) learn an SVM with hand-crafted linguistic features of events on a small amount of manually labeled data.

While the above tasks focus on fundamental event properties, there are a variety of tasks that focus on the specialized properties of an event to achieve a specific goal. For example, Lareau et al. (2011) consider an event property indicating whether the event is worth reporting and work on its automatic detection. Agarwal and Rambow (2010) consider a property of social events, indicating whether only one or both parties are aware of the social contact. Recently, fake news detection has been gaining attention, and is often formulated as an event classification task (Oshikawa et al., 2020).

2.3.3 Event-to-event Relation Analysis

Event-to-event relation analysis is the task of identifying the relation between two events. By event-to-event relation analysis, a list of events is transformed into a graph where events are linked according to their relations, enabling in-depth text analysis.

As described in Section 2.2, event-to-event relations can be divided into syn-

tactic and semantic relations. The analysis of syntactic relations has been studied at the level of syntactic parsing rather than event-level language analysis.

Discourse relation analysis is one of the semantic relation analysis tasks that has long been addressed. Discourse relation analysis is the task of recognizing the pragmatic relation between two events. The difficulty greatly depends on the presence or absence of discourse markers, which are words explicitly indicating discourse relations, such as *because* and *however*. Early studies developed a list of discourse markers for each discourse relation and recognized discourse relations based on the list. This method can achieve high precision; however, it is helpless in recognizing discourse relations between events without discourse markers.

Therefore, recent studies focus on the analysis of discourse relations that are not explicitly indicated by discourse markers; this task is called implicit discourse relation analysis. As described in 2.2, there exist labeled corpora for discourse relation analysis. Recent studies generally solve implicit discourse relation analysis by learning a neural network on the corpora. In order to improve the performance, researchers have considered good neural network architectures (Chen et al., 2016; Liu and Li, 2016; Bai and Zhao, 2018), incorporation of external knowledge (Kishimoto et al., 2018), knowledge transfer from explicit discourse relations (Rutherford et al., 2017; Qin et al., 2017), etc. However, the current best method is to fine-tune a general-purpose language model pretrained on a large-scale raw corpus in a self-supervised manner (Kim et al., 2020; Devlin et al., 2019), which does not rely on the above techniques. This suggests that by designing an appropriate self-supervised task, it is feasible to learn event representations capturing discourse relations from raw text, indicating a promising direction to improve discourse relation analysis.

Event ordering is also an actively studied task. The goal of event ordering is to order events based on the time they occurred. As with discourse relations, for temporal relations, there are linguistic expressions that specify the order in which events occur, such as *after* and *before*. However, not all temporal relations can be recognized by such expressions, and thus a deep semantic analysis of events is required to solve the task. Most studies use the standard benchmark TimeBank (Pustejovsky et al., 2003) to train and test models.

Causal relation analysis has been an important task of event-to-event relation analysis. There are several benchmark datasets that require commonsense knowledge about causal relations to solve, such as COPA (Roemmele et al., 2011) and KUCI (Omura et al., 2020). Researchers have worked on improving the performance on these tasks in various ways. While early studies relied on statistical and linguistic features obtained from superficial information Gordon et al. (2012), recent studies use pretrained general-purpose language models (Wang et al., 2019; Sap et al., 2019; Omura et al., 2020).

2.3.4 Event Prediction

Event prediction is the task of predicting events that are in a particular relationship to a given event. Humans often omit facts that can be inferred from commonsense, making it difficult for computers to understand language. Accordingly, event prediction is an important task that contributes to language understanding by computers. Besides, event prediction based on large-scale knowledge can be effective in providing insights to humans. For example, by learning what happens next after an event, it would be possible to predict the future impact of an event that one is interested in.

Event prediction is mainly studied for modeling causally or temporally ordered event sequences. Models are given an event sequence and try to restore a missing portion of it.

Event prediction can be categorized into two tasks: classification and generation. In the classification task, a model is required to choose one from a pre-defined set of candidates for a missing event. A popular strategy is to rank candidates by similarity with the remaining part of the event sequence (Chambers and Jurafsky, 2008; Jans et al., 2012; Granroth-Wilding and Clark, 2016). In the generation task, a model directly generates a missing event, usually in the form of a word sequence (Pichotta and Mooney, 2016; Hu et al., 2017; Nguyen et al., 2017; Du et al., 2019). For example, Nguyen et al. (2017) worked on the task of generating an event that is likely to happen after a given event, called next event prediction. They proposed to solve the task using a recurrent neural network-based model with the attention mechanism (Bahdanau et al., 2014).

2.4 Applications of Event-Level Language Analysis

There is a variety of real-world applications and high-level NLP tasks that event-level text analysis can contribute.

The most representative real-world application is customer feedback analysis. Customer feedback is one of the exploratory text analysis tasks and does not have a specific purpose. There, the role of NLP is to organize text at a granularity that is understandable to humans, helping humans find insights from the text.

CausalityGraph (Kiyomaru et al., 2020), developed by us, is an example of customer feedback systems. Figure 2.6 shows an example of text analysis by CausalityGraph. CausalityGraph organizes information by analyzing the causality of events and categorizing events into causes, results, and solutions. In CausalityGraph, first, events are extracted by syntactic dependency parsing. Then, event pairs in a causal relation are obtained using the result of discourse relation analysis. Finally, causal relations are further categorized into a cause-result relation and a cause-solution relation according to the properties of events.

Other real-world applications include experience mining (Inui et al., 2008). Experience mining extracts rich semantic structures called experiences, which roughly correspond to events, from a raw corpus. Extracted experiences are exploited for information search and knowledge discovery. While CausalityGraph aims to grasp the relation between events, experience mining focuses on capturing the relation between the components within an experience (i.e., an event).

Event-level text analysis contributes not only to the development of real-world applications but also to high-level NLP tasks. For example, event-level text analysis is beneficial for text summarization. The approaches for text summarization are divided into two types: abstractive and extractive. The abstractive approach focuses on generating a summary word-by-word after encoding the document. On the other hand, the extractive approach assembles a summary by selecting text spans from the document. The abstractive approach is more flexible and generally produces less redundant summaries, while the extractive approach enjoys better factuality (Cao et al., 2018). In the extractive approach, most studies employ sentences as the processing units. However, some recent studies employ events

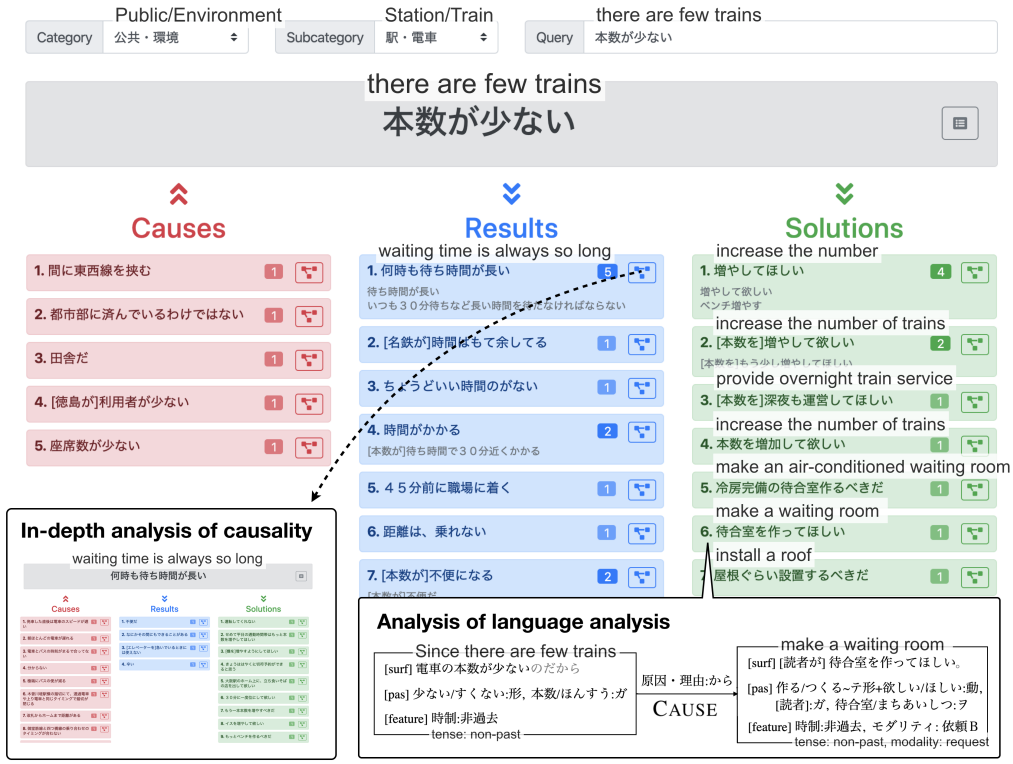

Figure 2.6: An example of an analysis by CausalityGraph. Given a query ("本数が少ない (there are few trains)," in this example), CausalityGraph displays its causes, results, and solutions. Each colored block represents a cluster containing one or more events. The first line in a block shows the representative event with the index number. The second or later lines show the other events in the cluster, if any. The number with a colored background shown at the right side indicates the number of events in the cluster. The button at the right-most position is linked to the analysis where the query is the events in the cluster, enabling an in-depth analysis of causality. By selecting a block, its language analysis is displayed in the form of a graph, where nodes and edges correspond to events and discourse relations, respectively.

as the processing units to reduce redundant or uninformative phrases (Li et al., 2016; Xu et al., 2020; Huang and Kurohashi, 2021).

Event-level language analysis also works effectively in the evaluation of lan-

guage generation. Language generation is an elemental technology for many NLP tasks, including machine translation, text summarization, and dialogue response generation. The widely used metrics calculate the superficial correspondence between machine-generated and human-generated texts. However, even if word sequences are similar, there may be fatal differences in the facts. To consider such errors, researchers have proposed to extract events from text and check whether their structures and properties are consistent or not. Giménez and Màrquez (2008) and Lo et al. (2012) focused on using semantic parsing to extract event structures and evaluate the generated texts at the level. Joty et al. (2017) proposed to compare discourse structures of events for the evaluation. Now that it is becoming possible to generate texts as fluent as humans, it is likely that such evaluation methods attract more and more attention.

2.5 Summary of This Chapter

In this chapter, we first introduced widely used event representations as preliminaries to event-level language processing. One representative event representation is structural representation. Structural representation is suitable for statistical analyses. On the other hand, because information that does not fit into structure is discarded, structure needs to be designed carefully according to the purpose. Another representative event representation is span representation. Span representation is a textual representation corresponding to a clause and does not have an explicit internal structure. Thanks to the recent advances in technology to embed semantically similar texts in a neighborhood in a continuous vector space, span representation is now used as an informative event representation suitable to apply high-level language analysis.

We then provided an overview of event-level language analysis. There is a large body of work on event-level language analysis. We categorized event-level language analysis into event extraction, event classification, event-to-event relation analysis, and event prediction, and introduced representative works on each of them. In this thesis, we present our works on event classification, event-to-event relation analysis, and event prediction. While we focus on a specific task of each

class, our proposed methods are potentially effective in performing tasks in the same class.

Finally, we presented several applications that build on event-level language analysis. Even in an end-to-end learning era, there exist applications where event-level language analysis is effectively used. In such applications, all of event classification, event-to-event relation analysis, and event prediction play an important role.

Chapter 3

Volitioanlity Classification

Volitionality and subject animacy are fundamental and closely related properties of events. Their classification, however, is challenging because it requires contextual text understanding and a huge amount of labeled data. This paper proposes a novel method that jointly learns volitionality and subject animacy at a low cost, heuristically labeling events in a raw corpus. Volitionality labels are assigned using a small lexicon of volitional and non-volitional adverbs such as deliberately and accidentally; subject animacy labels are assigned using a list of animate and inanimate nouns obtained from ontological knowledge. Since our labeling method assigns labels only to a biased set of events, a classifier is trained with regularization to take into account the property. This paper explores the following two approaches: bias reduction and adversarial representation learning. In bias reduction, the words used for labeling are regarded as bias that should not be over-exploited to make predictions, and their estimated contribution towards predictions is penalized. In adversarial representation learning, the classifier is given unlabeled events as well and makes their latent representations closer to labeled events' ones in an adversarial manner while learning classification on labeled events. We conduct experiments with crowdsourced gold data in Japanese and English and show that our method effectively learns volitionality and subject animacy without manually labeled data.

3.1 Introduction

Volitionality is a fundamental property of events, which indicates whether an event represents an action that someone is volitionally involved in. In this study, we particularly focus on whether the entity represented by the subject is volitionally involved in the event or not. For example, eating and writing are usually volitional; crying and getting injured are non-volitional. Event volitionality classification has been used for causal knowledge categorization (Lee and Jun, 2008; Inui et al., 2003; Abe et al., 2008,) and has various potential applications such as conditional event prediction (Du et al., 2019), script induction (Chambers and Jurafsky, 2008), and customer feedback analysis (Liu et al., 2017).

On the other hand, *animacy* is a fundamental property of nouns, which indicates whether the entity described by a noun is capable of human-like volition (Bowman and Chopra, 2012). In this study, because we focus on whether or not the entity represented by the subject is volitionally involved in the event, the fact that the subject of an event is an animate noun is a necessary condition for an event to be volitional. Focusing on this close relationship, we consider the event property of subject animacy. It is expected that the joint learning of subject animacy classification will help models learn event volitionality.

The challenge of identifying volitionality and animacy lies in limited language resources and contextual dependence. The volitionality of an event is largely decided by its predicate. However, existing language resources such as ConceptNet (Speer et al., 2017) do not provide an exhaustive list of volitional predicates.

Even with a rich language resource, however, due to its context-dependent nature, volitionality cannot be entirely identified. Let us consider the following Japanese examples.

(12) a. *shawa-o* *abiru* (V)
shower-ACC[1] take

b. *hinan-o* *abiru* (NV)
criticism-ACC get

[1]ACC is the accusative case marker.

Examples (12-a) and (12-b) have the same predicate "*abiru* (take/get)," but the former is volitional, while the latter is non-volitional.[2]

Similarly, example (13-a) is non-volitional, but example (13-b) is volitional because of the adverb "*fukaku* (deeply)."

(13) a. *iki-o* *suru* (NV)
breath-ACC take
b. *fukaku* *iki-o* *suru* (V)
deeply breath-ACC take

Coupled with the unbounded combinatorial nature of language, such contextual dependence entails the demand for learning from a huge amount of labeled data.

As for animacy, although there exist some available language resources listing animate/inanimate nouns, they are far from exhaustive. Besides, identifying animacy also requires contextual text understanding. For example, although examples (14-a) and (14-b) have the same subject "*shirobai* (white motorcycle)," the former describes an inanimate entity, a motorcycle, while the latter describes an animate entity, a police officer, as metonymy.[3]

(14) a. *shirobai-ga* *tometearu* (IA)
white motorcycle-NOM[4] be parked
b. *shirobai-ga* *oikaketekuru* (A)
white motorcycle-NOM chase

This paper proposes a minimally supervised method to jointly learn volitionality and subject animacy. Figure 3.1 shows the overview of the proposed method. We first assign labels to events in a raw corpus in a heuristic manner. Volitionality labels are assigned using a small lexicon of volitional and non-volitional adverbs, collectively called the *volitionality indicating words*. For example, example (15) is

[2]We use "V" and "NV" to indicate that an event is volitional and non-volitional from the viewpoint of the subject, respectively.

[3]We use "A" and "IA" to indicate that the subject of an event is animate and inanimate, respectively.

[4]NOM is the nominative case marker.

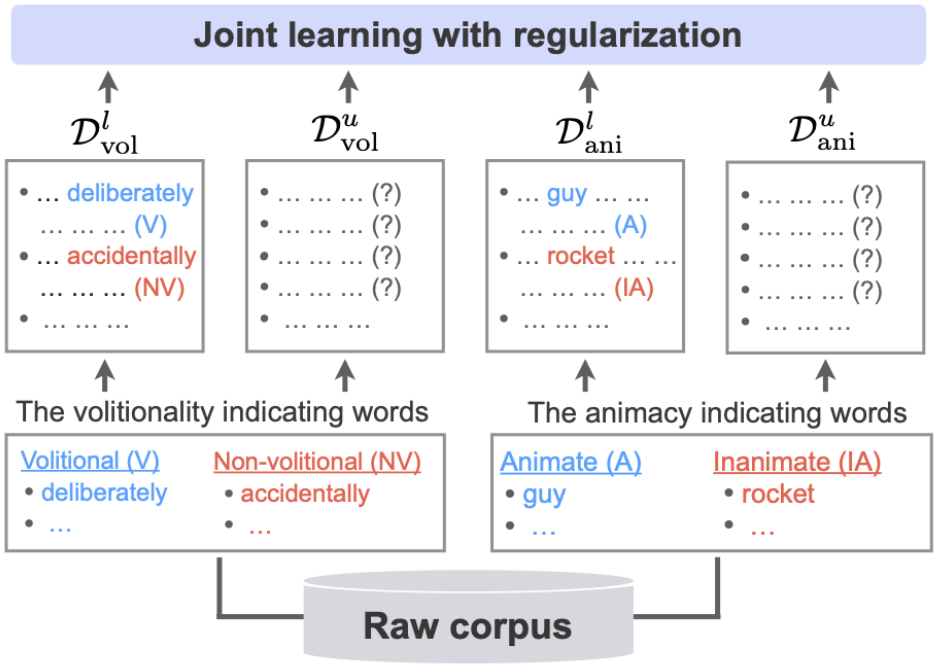

Figure 3.1: Overview of our method. We construct labeled and unlabeled datasets for volitionality and subject animacy classification by heuristically labeling events in a raw corpus using the volitionality/animacy indicating words. Our model jointly learns volitionality and subject animacy on them with regularization.

regarded as volitional because the volitional adverb "*aete* (deliberately)" modifies the predicate.

(15) *aete* *shinjitsu-o* *hanasu* (V)
deliberately truth-ACC tell

Example (16) is regarded as non-volitional because the non-volitional adverb "*ukkari* (accidentally)" modifies the predicate.

(16) *ukkari* *keitai-o* *otosu* (NV)
accidentally mobile-ACC drop

Subject animacy labels are assigned using a list of animate/inanimate nouns, collectively called the *animacy indicating words*, obtained from ontological knowledge. By using this labeling method, a large number of labeled events can be collected at a low cost.

As examples (15) and (16) suggest, we can consider that volitionality is preserved after removing the volitionality indicating words in most cases. The same can be said for subject animacy.

However, this is not always true. For example, example (17-a) is volitional, but example (17-b) is non-volitional. Here, the volitionality indicating word "*aete* (deliberately)" plays an essential role.

(17) a. *aete* *kokeru* (V)
deliberately tumble

b. *kokeru* (NV)
tumble

Such cases also exist in subject animacy classification. While the subject of example (18-a) "*shogeki* (impact)" is inanimate, the omitted subject of example (18-b) is normally assumed to be animate.

(18) a. *shogeki-ga* *hashiru* (IA)
impact-NOM run

b. *hashiru* (A)
run

To obtain classifiers that generalizes to events that do not volitionality/animacy indicating words, it is important to basically learn to predict labels from the text that co-occurs with the words without relying on the words, but not to learn to predict labels from the text that co-occurs with the words for examples where excluding the words changes the label. In this study, we expect that the classifier learns the former principle by introducing a regularization that suppresses classification based only on the words, while learning classification on generalized event representations produced by general-purpose language models, so that cases

in which the use of the words is essential for prediction will be learned from data.

In this paper, we explore the following two approaches: bias reduction and adversarial representation learning. In bias reduction, the volitionality/animacy indicating words are regarded as bias and should not be over-exploited to make predictions. During training, the classifier learns to reduce the contribution of the volitionality/animacy indicating words towards predictions (Kennedy et al., 2020; Jin et al., 2020). In adversarial representation learning, the classifier is given unlabeled events as well and learns to make their latent representations closer to labeled events' ones using an adversarial learning framework while learning classification on labeled events (Ganin and Lempitsky; Ganin et al., 2016).

We conduct experiments with crowdsourced gold data in Japanese and English and verify the effectiveness of the proposed method to learn volitionality and subject animacy without manually labeled data.

3.2 Related Work

Our work mainly builds on event volitionality classification, bias reduction, and unsupervised domain adaptation.

3.2.1 Event Volitionality Classification

Previous work on event volitionality classification can be categorized into a targeted setting and a non-targeted setting. In the targeted setting, a model is given the predicate and its argument of an event and predicts whether the argument is volitionally involved in the action or state the predicate represents. This setting has been tackled as a sub-task of semantic proto-role labeling (Reisinger et al., 2015; White et al., 2016; Teichert et al., 2017).

In the non-targeted setting, which we tackle in this paper, a model is given an event and predicts whether the subject is volitionally involved in the event. To this end, Abe et al. (2008) and Abe et al. (2008) manually built a lexicon of verbs with volitionality labels and classified event volitionality by looking it up. This method is constrained by its inability to take context into account; as examples (12-a) and (12-b) suggest, volitionality depends on context.

Inui et al. (2003) proposed a data-driven approach; they learned an SVM with hand-crafted linguistic features of events on a small amount of manually labeled data. However, the non-compositionality of event volitionality prevents us from learning from a small dataset. We use a massive amount of heuristically labeled events to learn a wide range of language phenomena and world knowledge related to volitionality.

3.2.2 Bias Reduction

Bias reduction is a technique to prevent a model from exploiting a specific bias to make predictions. While bias reduction has been actively studied in the field of fairness in machine learning (Bolukbasi et al., 2016; Zhao et al., 2017, 2019; Kennedy et al., 2020), we use this technique to prevent our model from over-exploiting the volitionality/animacy indicating words. Specifically, we employ two bias reduction methods proposed in Kennedy et al. (2020): word removal and explanation regularization based on sampling and occlusion (Jin et al., 2020). These methods were originally proposed to learn a hate speech classifier robust to group identifiers such as "gay." The details of these methods are deferred to Section 3.4.3.

3.2.3 Unsupervised Domain Adaptation

It is reasonable to employ semi-supervised learning techniques to solve our problem because our training data includes both labeled and unlabeled events. In the context of semi-supervised learning, given that our primary focus is on classifying unlabeled events to which our heuristics cannot assign labels, it is natural to view our problem as an unsupervised domain adaptation problem (Ramponi and Plank, 2020).

Unsupervised domain adaptation is a technique to learn a model that better performs on a target domain, using labeled data from a source domain and unlabeled data from the target domain. We employ this technique regarding labeled events and unlabeled events as source domain data and target domain data, respectively. Specifically, we adopt adversarial domain adaptation (ADA) that has been used successfully in NLP tasks, including text classification such as sentiment

analysis (Ganin et al., 2016; Ganin and Lempitsky; Shah et al., 2018; Shen et al., 2018). In ADA, a model learns a latent feature space to reduce the discrepancy between the source and target distributions while learning a task using the source domain data, using an adversarial learning framework. The detail is deferred to Section 3.4.3.

3.3 Problem Setting

This section describes the representation, scope, and annotation of events we target in the present paper.

3.3.1 Representation

We represent an event as a clause, that is, a text that contains one main predicate. Compared to structured representations such as predicate-argument structures (Gildea and Jurafsky, 2000), clauses can more flexibly represent the meaning of events. Besides, by representing events by clauses, we can obtain powerful event representations using strong pretrained text encoders like BERT (Devlin et al., 2019).

3.3.2 Scope

This paper focuses on events whose volitionality cannot be identified by simple linguistic features: POS tags and voice. We use POS tags to filter out events whose predicates are either an adjective or copula because they always represent a state and thus never represent a volitional action, as shown in examples (19) and (20).

(19) *sora-ga* *kireida* (NV)
sky-NOM be beautiful

(20) *kare-wa* *gakuseida* (NV)
he-NOM be student

As for voice, we filter out events in the passive or potential voice because they

are not volitional from the viewpoint of their subjects, as shown in examples (21) and (22).

(21) *sensei-ni* *shikarareru* (NV)
teacher-DAT[5] be scolded

(22) *watashi-wa* *hashireru* (NV)
I-NOM can run

Besides, we filter out events with modality, linguistic expressions representing the writer's opinions or attitudes towards an event. Example (23) contains the modality of CERTAINTY expressed by "*hazuda* (should)."

(23) *kare-wa* *kuru* *hazuda*
he-NOM come should

Because our focus is on recognizing the volitionality of an event itself, we exclude such an event from the scope.

3.3.3 Annotation

An event is given volitionality and subject animacy labels.

Volitionality An event is considered *volitional* if the subject is volitionally involved in the event. Otherwise, it is considered *non-volitional*.

Subject Animacy The subject of an event is considered *animate* if the entity described by it can take volitional actions. Otherwise, it is considered *inanimate*. Since we consider a model that is given an event and predicts its subject animacy, we tie an animacy label to an event rather than the subject.

[5]DAT is the dative case marker.

3.4 Proposed Method

Our goal is to train a model that is given an event x and predicts its volitionality y_{vol} and subject animacy y_{ani}. Both y_{vol} and y_{ani} take the value of 1 if positive (volitional/animate) and 0 if negative (non-volitional/inanimate). First, labeled events are collected from a raw corpus with our heuristic labeling method. Then, considering the property of the labeled events discussed in Section 3.1, our model jointly learns volitionality and subject animacy with regularization.

3.4.1 Constructing Training Dataset

We construct four types of datasets: events with volitionality labels $\mathcal{D}^l_{\text{vol}}$, events without volitionality labels $\mathcal{D}^u_{\text{vol}}$, events with subject animacy labels $\mathcal{D}^l_{\text{ani}}$, and events without subject animacy labels $\mathcal{D}^u_{\text{ani}}$.

First, events that satisfy the conditions described in Section 3.3.2 are extracted from a raw corpus, using an off-the-shelf syntactic dependency parser and POS tagger. Each of the events is then given its volitionality and subject animacy labels by our heuristic labeling method. According to the given label, the event is added to the corresponding dataset.

To assign the volitionality label, we prepare a small lexicon of volitional and non-volitional adverbs. If an adverb in the lexicon modifies the predicate of the event, the event is given the corresponding label and added to $\mathcal{D}^l_{\text{vol}}$. Otherwise, the event is added to $\mathcal{D}^u_{\text{vol}}$ without being given a label.

To assign the subject animacy label, we first find the subject of the event using a semantic dependency parser. If the subject is found, its animacy is then examined by looking up the animacy indicating words obtained from ontological knowledge and using the result of named entity recognition. If the animacy is identified, the event is associated with the corresponding label and pushed into $\mathcal{D}^l_{\text{ani}}$. If the subject is not found — which is not rare in pro-drop languages, including Japanese — or its animacy is not identified, the event is added to $\mathcal{D}^u_{\text{ani}}$ without being given a label.

3.4.2 Model

Our model consists of the following three neural networks: a text encoder E, a volitionality classifier C_{vol}, and a subject animacy classifier C_{ani}. The text encoder transforms an event x into a distributed representation. The volitionality classifier is given the representation and predicts the probability of x being volitional. Likewise, the subject animacy classifier predicts the probability that the subject of x is animate.

3.4.3 Training with Regularization

Our model jointly learns volitionality classification and subject animacy classification with regularization. As their training is done in a unified manner, we introduce placeholders for convenience. We refer to a labeled dataset as $\mathcal{D}^l$, an unlabeled dataset as $\mathcal{D}^u$, the label assigned to events in $\mathcal{D}^l$ as y, and the classifier to predict y as C. When learning volitionality, these placeholders are accompanied by the suffix "vol"; as for subject animacy, they are accompanied by the suffix "ani."

Our model learns classification using the labeled dataset. Formally, the objective is written as follows:

$$\mathcal{L}_{\text{cls}} = \mathbb{E}_{(x,y)\sim\mathcal{D}^l}\text{BCE}(y, C(E(x))), \tag{3.1}$$

where BCE is binary cross-entropy.

We explore the following regularization methods.

Word Removal (WR) WR is a bias reduction method that decreases reliance on a word to make predictions by removing the word from training data. We use this method to reduce reliance on the volitionality/animacy indicating words. The objective is written as follows:

$$\mathcal{L}_{\text{WR}} = \mathbb{E}_{(x,y)\sim\mathcal{D}^l}\text{BCE}(y, C(E(x\backslash w))), \tag{3.2}$$

where w is the volitionality/animacy indicating word in x and $x\backslash w$ is x from which w is removed.

Explanation regularization by sampling and occlusion (SOC) SOC is a bias reduction method that penalizes the context-independent contribution of a word towards predictions (Kennedy et al., 2020). In order to estimate a context-independent contribution, SOC calculates the difference of model output after masking out the word, marginalized over all the possible context of the word. We use this method to reduce reliance on the volitionality/animacy indicating words. Formally, the objective is written as follows:

$$\mathcal{L}_{\mathrm{SOC}} = \mathbb{E}_{x\sim\mathcal{D}^l}[\phi(x)]^2, \tag{3.3}$$

$$\phi(x) = \frac{1}{|\mathcal{S}|}\sum_{x'\in\mathcal{S}}[C(E(x')) - C(E(x'\backslash w))]^2, \tag{3.4}$$

where w is the volitionality/animacy indicating word in x, $\mathcal{S}$ is a set of events created by sampling the context of w according to a pretrained language model, and $x'\backslash w$ is x' that w is replaced with a padding token.

Adversarial Domain Adaptation (ADA) ADA is an unsupervised domain adaptation technique (Ganin et al., 2016; Ganin and Lempitsky). In ADA, a model learns to make the features of unlabeled data from a target domain closer to the features of labeled data from a source domain while learning a task using the labeled data. This training is done in an adversarial manner. During training, an additional neural network called discriminator D is trained. The discriminator is given the output of the encoder and predicts 1 if the input is source domain data and 0 otherwise. The encoder learns to fool the discriminator. We use ADA considering the labeled dataset as source domain data and the unlabeled dataset as target domain data. Formally, the objective is written as follows:

$$\begin{aligned}\mathcal{L}_{\mathrm{ADA}} =& \mathbb{E}_{x\sim\mathcal{D}^l}\mathrm{BCE}(0, D(E(x))) \\ &+ \mathbb{E}_{x\sim\mathcal{D}^u}\mathrm{BCE}(1, D(E(x))).\end{aligned} \tag{3.5}$$

This training is done efficiently by employing a gradient reversal layer (Ganin et al., 2016; Ganin and Lempitsky).

Consistency (CON) CON learns the consistency of volitionality classification and subject animacy classification on the unlabeled datasets. Recall that animacy

Volitional	Non-volitional
aete (5,293)	*omowazu* (18,115)
isoide (4,187)	*tsui* (15,897)
jikkuri (4,017)	*jidoutekini* (14,212)
shinchoni (3,743)	*futo* (12,050)
wazawaza (3,262)	*tsuitsui* (10,054)

Table 3.1: The five most frequent Japanese volitionality indicating words in our lexicon. The numbers in parentheses indicate frequency.

is a necessary condition for volitionality. Therefore, it is implausible to predict that an event is volitional while predicting that its subject is inanimate. CON learns this relationship by:

$$\mathcal{L}_{\text{CON}} = \mathbb{E}_{x \sim \mathcal{D}^u_{\text{vol}} + \mathcal{D}^u_{\text{ani}}} \max(0, C_{\text{vol}}(E(x)) - C_{\text{ani}}(E(x))). \tag{3.6}$$

These regularization objectives are combined with the classification objective with a weight. Our training objective is finally written as follows:

$$\mathcal{L} = \mathcal{L}_{\text{cls}} + \alpha \mathcal{L}_{\text{WR|SOC|ADA}} + \beta \mathcal{L}_{\text{CON}}, \tag{3.7}$$

where α and β are weights selected as hyper-parameters and $\mathcal{L}_{\text{WR|SOC|ADA}}$ is either $\mathcal{L}_{\text{WR}}$, $\mathcal{L}_{\text{SOC}}$, or $\mathcal{L}_{\text{ADA}}$.[6]

3.5 Experiments

We conducted experiments on Japanese and English.

3.5.1 Training Dataset

We constructed training datasets following the procedure described in Section 3.4.1.

[6]It is possible to combine WR, SOC, and ADA, in theory. We did not try that due to the computational cost.

Volitional	Non-volitional
carefully (13,594)	unfortunately (13,070)
thoroughly (12,468)	automatically (12,824)
actively (10,379)	accidentally (5,272)
deliberately (3,366)	unexpectedly (3,106)
intentionally (2,713)	luckily (1,894)

Table 3.2: The five most frequent English volitionality indicating words in our lexicon. The numbers in parentheses indicate frequency.

Japanese We used 30M documents in CC-100 as a raw corpus (Conneau et al., 2020; Wenzek et al., 2020). Events were parsed and extracted using KNP, a widely used Japanese parser (Kawahara and Kurohashi, 2006). For volitionality labeling, we manually constructed a lexicon of 15 volitional and 15 non-volitional adverbs. Table 3.1 shows the most frequently matched adverbs. Refer to Appendix A for the full list. For animacy labeling, we used the dictionary on which KNP builds[7] as ontological knowledge. It contained approximately 30K nouns with animacy labels. We also used the named entity recognizer built into KNP to recognize animacy. We did not delete duplicate events to preserve frequency information.

English We again used 30M documents in CC-100 as a raw corpus. Events were parsed and extracted using spacy[8]. For volitionality labeling, we manually constructed a lexicon of 10 volitional and 10 non-volitional adverbs. Table 3.2 shows the most frequently matched adverbs. Appendix A includes the full list. For animacy labeling, we obtained animate/inanimate nouns from ConceptNet (Speer et al., 2017). Specifically, we used the hyponyms of "person" and "organization" as animate nouns, and the hyponyms of "object," "item," "thing," "artifact," and "location" as inanimate nouns. As a result, we obtained 2,604 animate nouns and 430 inanimate nouns. Besides, we used the named entity recognizer built into spacy for animacy recognition.

[7] https://github.com/ku-nlp/JumanDIC

[8] https://spacy.io

3.5.2 Evaluation Dataset

We constructed an evaluation dataset for each of $\mathcal{D}^l_{\text{vol}}$, $\mathcal{D}^u_{\text{vol}}$, $\mathcal{D}^l_{\text{ani}}$, and $\mathcal{D}^u_{\text{ani}}$. We first randomly extracted 1,200 unique events from each of the datasets. We then assigned the ground truth to them by crowdsourcing.

As for volitionality labeling, crowdworkers were given an event and assigned one of the following labels:

- The subject is volitionally involved in the event.
- The subject is not volitionally involved in the event.
- Unable to say either.
- Unable to understand.

As for animacy labeling, crowdworkers were given an event and assigned one of the following labels:

- The subject is a person(s) or organization(s).
- The subject is neither a person(s) nor organization(s).
- Unable to say either.
- Unable to understand.

Each event was annotated by five crowdworkers. One crowdworker annotated ten events.

For Japanese, we used Yahoo! Crowdsourcing[9] as a crowdsourcing platform. Figure 3.2 and Figure 3.3 show the user interfaces. For quality control, we used the function provided in the platform to reject workers who made a mistake on an easy question that we manually prepared in advance. The total cost was 24,000 JPY.

For English, we used Amazon Mechanical Turk (MTurk). Figure 3.4 and Figure 3.5 show the user interfaces. For quality control, we followed common best practices (Berinsky et al., 2012); workers had to have over a 95% acceptance rate, live in the US, and have done more than 1,000 tasks. The total cost was 288 USD.

[9] `https://crowdsourcing.yahoo.co.jp/`

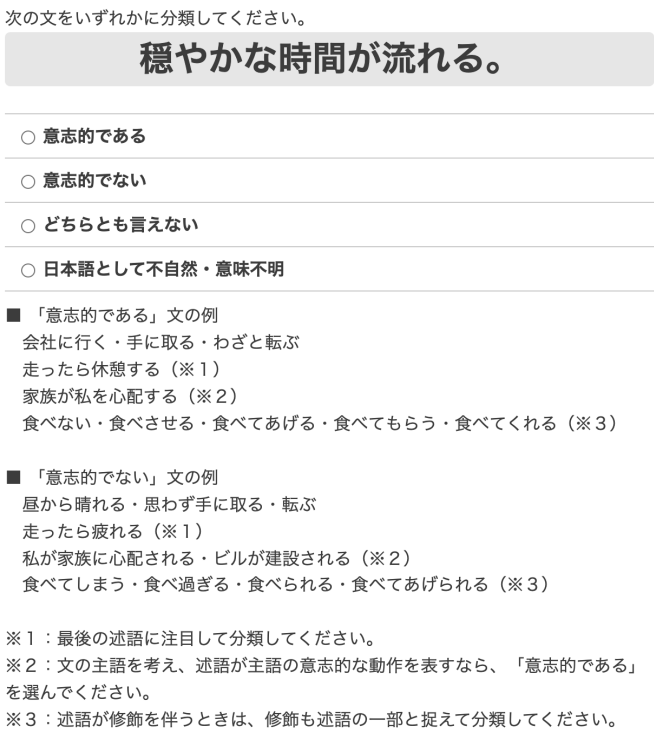

Figure 3.2: The user-interface to annotate volitionality labels to Japanese events.

Table 3.3 shows the inter-annotator agreement rates. Events with an agreement rate of 80% or more were extracted, and half were used for validation and the other half were used for testing. Table 3.4 summarizes the constructed datasets.

3.5.3 Implementation Detail

The encoder was pretrained $\text{BERT}_{\text{BASE}}$ (Devlin et al., 2019). We used the output of the classification token ([CLS]) as event representations. The classifiers were a three-layered fully-connected neural network with the ReLU nonlinearity followed by the sigmoid function. The discriminator used in Ada had the same architecture as the classifiers. Soc was used with a sample size of three. α was selected from {0.0, 0.01, 0.1, 1.0} for each of Wr, Soc, and Ada. β was selected from {0.0, 0.01, 0.1, 1.0}. We trained the model for three epochs with a batc/h size of 256. We used the Adam optimizer (Kingma and Ba, 2015) with a learning rate of 3e-5,

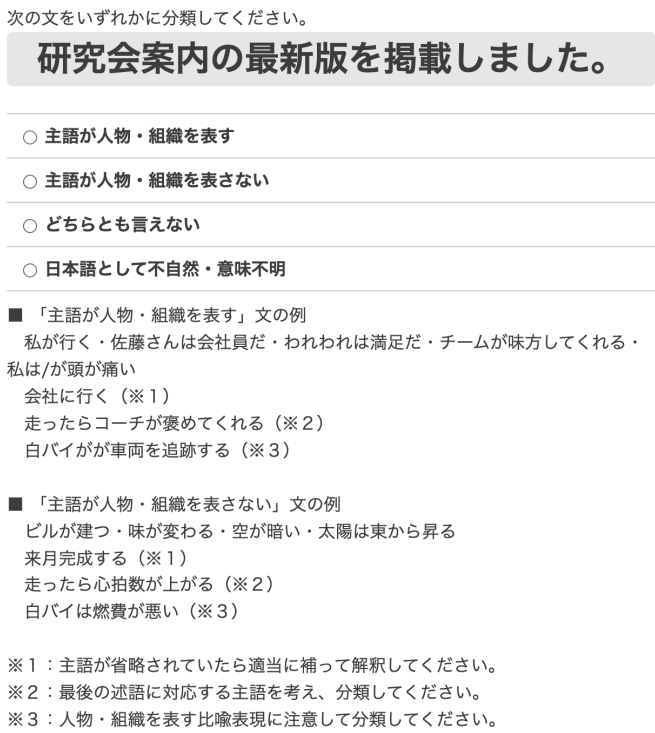

Figure 3.3: The user-interface to annotate subject animacy labels to Japanese events.

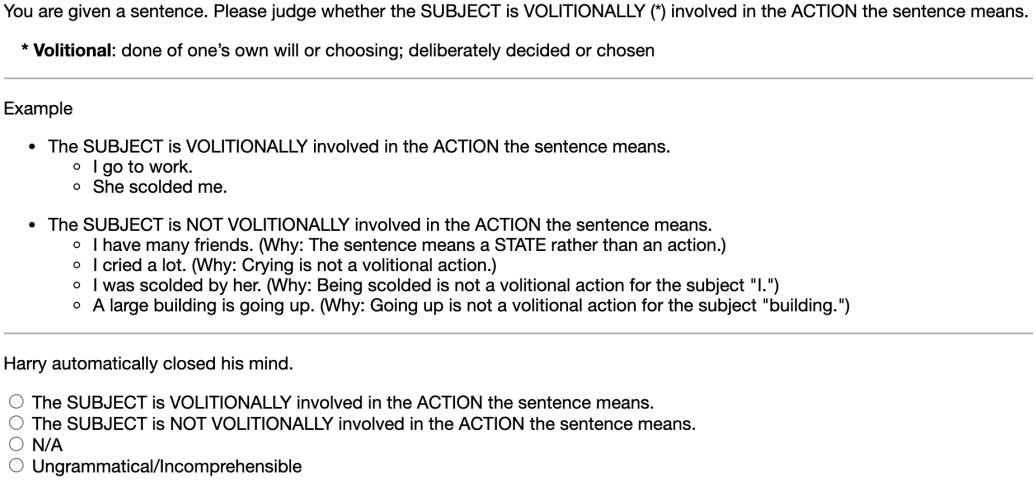

Figure 3.4: The user-interface to annotate volitionality labels to English events.

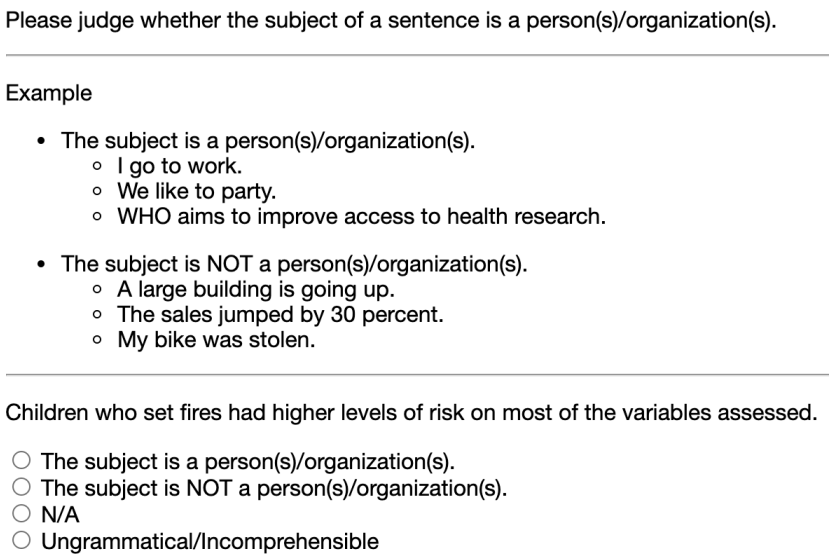

Figure 3.5: The user-interface to annotate subject animacy labels to English events.

	$\mathcal{D}^l_{\text{vol}}$	$\mathcal{D}^u_{\text{vol}}$	$\mathcal{D}^l_{\text{ani}}$	$\mathcal{D}^u_{\text{ani}}$
Japanese	78.2	76.9	77.2	74.3
English	62.4	62.8	66.8	67.1

Table 3.3: The inter-annotator agreement rate for each dataset, calculated by averaging the ratios of majority answers.

linear warmup of the learning rate over the first 10% steps, and linear decay of the learning rate. We evaluated the performance on the development dataset of $\mathcal{D}^u_{\text{vol}}$, which was our primary concern, at every 100 steps, and adopted the checkpoint that achieved the best performance. The evaluation metric was the AUC of the ROC curve. Models were trained three times with different random seeds. We used Pytorch for implementation.

3.5.4 Results

Table 3.5 and Table 3.6 show the result. In both Japanese and English, joint learning combined with regularization achieved the best performance on both

	Split	Label	Japanese	English
$\mathcal{D}^l_{\text{vol}}$	Train	Volitional	31,812	47,926
		Non-volitional	81,002	40,564
	Dev	Volitional	149	67
		Non-volitional	233	92
	Test	Volitional	149	68
		Non-volitional	233	93
$\mathcal{D}^u_{\text{vol}}$	Train	Unlabeled	112,814+	88,490+
	Dev	Volitional	206	62
		Non-volitional	164	104
	Test	Volitional	206	63
		Non-volitional	164	104
$\mathcal{D}^l_{\text{ani}}$	Train	Animate	29,344+	71,257+
		Inanimate	83,470+	17,233+
	Dev	Animate	175	170
		Inanimate	199	59
	Test	Animate	176	170
		Inanimate	200	60
$\mathcal{D}^u_{\text{ani}}$	Train	Unlabeled	112,814+	88,490+
	Dev	Animate	246	78
		Inanimate	93	152
	Test	Animate	246	78
		Inanimate	93	153

Table 3.4: Statistics of our dataset. The number with + means that the events were randomly sampled from a larger set according to the size of smallest dataset, $\mathcal{D}^l_{\text{vol}}$.

volitionality and subject animacy classification on the unlabeled datasets and most of the labeled datasets. Specifically, when joint learning was employed, SOC was constantly effective to learn volitionality classification. Without joint

Vol.	Ani.		$\mathcal{D}^l_{\text{vol}}$	$\mathcal{D}^u_{\text{vol}}$	$\mathcal{D}^l_{\text{ani}}$	$\mathcal{D}^u_{\text{ani}}$
NONE	VAN	+ CON	65.3±2.6	77.3±0.9	92.0±0.7	81.4±0.8
	WR	+ CON	73.5±1.4	85.1±1.0	94.3±0.6	86.4±0.2
	SOC	+ CON	73.7±2.9	82.3±1.7	93.9±0.2	84.5±1.3
	ADA	+ CON	69.7±1.0	81.5±0.7	92.7±0.7	81.9±4.2
VAN	NONE	+ CON	91.7±1.0	89.5±1.1	72.6±2.4	70.7±2.7
	VAN		91.8±1.3	90.6±0.1	91.3±0.3	81.7±1.7
		+ CON	92.1±0.5	89.6±1.9	87.7±3.3	83.0±1.2
	WR		93.9±0.6	92.5±1.2	94.0±0.0	86.4±0.4
		+ CON	92.1±0.9	92.8±1.0	96.0±0.5	88.5±0.3
	SOC		90.7±0.7	94.7±0.7	92.8±1.1	85.4±0.7
		+ CON	91.5±1.0	93.5±0.6	89.6±1.5	83.7±0.8
	ADA		92.3±0.4	89.9±2.7	87.2±3.3	82.2±2.0
		+ CON	92.2±0.4	90.9±3.1	87.7±3.1	82.0±2.1
WR	NONE	+ CON	91.5±1.5	91.2±0.8	57.3±10.6	57.6±10.7
	VAN		92.4±0.8	91.9±0.1	88.8±7.6	83.9±1.4
		+ CON	93.2±0.8	93.2±1.3	84.7±5.8	82.2±1.4
	WR		91.8±0.7	93.2±0.9	94.3±1.2	87.1±1.4
		+ CON	93.4±1.3	93.0±1.0	93.6±1.5	86.3±1.1
	SOC		91.3±0.6	95.1±0.2	90.8±1.8	84.8±0.9
		+ CON	92.1±0.4	94.9±0.4	88.9±3.8	83.6±2.4
	ADA		92.4±0.5	92.3±1.3	83.6±3.4	81.9±0.3
		+ CON	93.9±1.0	93.1±0.8	85.1±5.5	81.7±1.0
SOC	NONE	+ CO	94.4±0.6	92.9±0.1	67.3±2.2	67.1±0.7
	VAN		94.6±0.4	94.0±0.5	92.3±1.8	86.1±0.8
		+ CON	94.6±0.4	94.7±0.5	90.0±1.0	84.5±0.4
	WR		94.3±0.1	**96.7±0.7**	95.3±1.0	**89.9±0.6**
		+ CON	94.5±0.3	**96.7±0.4**	90.1±0.9	84.5±0.6
	SOC		94.5±0.3	95.2±0.3	91.3±1.3	86.0±0.8
		+ CON	94.4±0.4	96.0±0.5	90.0±0.4	84.6±0.8
	ADA		94.6±0.5	95.9±0.1	92.1±2.0	85.3±0.9
		+ CON	95.0±0.8	95.1±1.2	90.4±1.0	84.6±0.8
ADA	NONE	+ CON	**96.3±0.6**	93.6±0.9	73.6±2.1	73.4±1.6
	VAN		90.7±0.4	91.8±0.5	90.8±0.7	82.4±1.7
		+ CON	92.1±0.5	89.5±2.4	86.9±4.3	82.8±1.5
	WR		92.0±1.3	94.6±0.4	94.9±1.3	86.8±1.1
		+ CON	93.1±1.0	94.5±1.0	**95.9±0.3**	87.6±0.8
	SOC		91.2±0.6	94.6±0.9	91.2±2.4	84.3±0.5
		+ CON	91.6±0.6	93.6±0.2	88.7±1.0	83.3±0.2
	ADA		91.9±0.3	90.8±1.1	87.3±2.7	83.1±0.9
		+ CON	92.2±0.4	91.3±1.4	87.5±2.9	83.4±0.8

Table 3.5: The result of volitionality classification and subject animacy classification in Japanese. The bold scores indicate the highest ones over models, and the underlined scores indicate the highest ones over models trained without joint learning.

Vol.	Ani.		$\mathcal{D}^l_{\text{vol}}$	$\mathcal{D}^u_{\text{vol}}$	$\mathcal{D}^l_{\text{ani}}$	$\mathcal{D}^u_{\text{ani}}$
NONE	VAN	+ CON	64.0±0.9	69.3±0.7	83.5±1.0	82.4±1.3
	WR	+ CON	65.1±0.6	70.7±0.2	84.2±0.4	81.7±0.4
	SOC	+ CON	63.5±2.0	70.0±0.6	84.3±1.6	82.7±1.7
	ADA	+ CON	62.9±3.6	69.6±1.2	84.6±1.2	83.4±2.0
VAN	NONE	+ CON	73.7±1.8	66.2±0.9	67.2±3.6	66.0±1.9
	VAN		74.4±0.5	70.7±3.1	82.3±2.1	81.5±1.0
		+ CON	74.0±1.6	69.8±3.0	84.3±1.6	81.9±1.3
	WR		72.4±4.3	69.7±0.2	83.6±0.7	81.6±0.1
		+ CON	71.9±2.6	70.8±0.6	84.1±0.8	81.9±0.6
	SOC		72.8±1.9	72.0±0.4	83.5±2.9	78.6±2.9
		+ CON	72.8±1.4	69.5±1.1	84.9±0.4	82.3±0.3
	ADA		74.4±0.7	72.8±2.2	84.1±1.3	82.8±1.4
		+ CON	72.0±1.3	70.9±2.0	82.2±1.7	82.3±1.3
WR	NONE	+ CON	69.8±0.8	70.0±0.3	55.5±0.5	67.4±1.1
	VAN		73.0±1.0	73.1±2.3	82.6±2.1	84.0±0.5
		+ CON	72.3±0.5	72.4±1.2	82.3±0.9	83.7±0.5
	WR		72.4±0.8	75.5±1.0	82.5±2.2	83.6±0.2
		+ CON	72.6±0.8	71.9±1.4	82.0±0.8	84.5±1.0
	SOC		72.9±1.4	75.2±1.4	81.2±2.4	82.8±0.7
		+ CON	72.3±1.1	75.5±1.8	80.7±2.2	83.5±1.8
	ADA		72.3±0.7	75.6±0.6	82.3±2.1	83.6±0.2
		+ CON	72.3±0.8	72.7±0.7	81.4±1.3	83.9±0.4
SOC	NONE	+ CON	73.3±0.4	72.2±1.3	66.2±1.8	73.0±1.3
	VAN		73.9±0.5	75.4±0.8	83.2±1.9	83.3±0.2
		+ CON	73.7±0.3	75.8±0.5	85.4±0.3	**85.1±0.3**
	WR		73.2±0.3	74.0±1.9	84.0±0.1	83.0±0.4
		+ CON	73.6±0.2	75.7±0.2	84.6±0.9	84.7±0.4
	SOC		73.8±0.1	76.7±1.4	**86.2±0.4**	81.4±1.2
		+ CON	73.6±0.3	**77.1±1.1**	85.4±0.8	84.4±0.4
	ADA		73.1±0.4	74.0±1.8	83.9±0.4	83.1±0.2
		+ CON	73.5±0.3	75.4±0.3	84.7±0.8	84.7±0.5
ADA	NONE	+ CON	**74.9±1.3**	70.0±3.4	62.2±3.2	64.4±1.9
	VAN		70.5±2.3	70.5±1.3	84.9±0.6	80.4±1.4
		+ CON	72.3±4.4	69.9±1.9	84.9±0.7	81.0±2.6
	WR		71.7±0.8	70.6±0.6	83.3±0.7	82.2±1.3
		+ CON	69.7±2.4	71.5±1.1	85.2±1.0	80.4±2.5
	SOC		69.1±3.4	73.2±1.6	84.0±3.0	78.2±1.6
		+ CON	69.4±1.7	68.3±0.8	85.1±0.8	82.0±1.7
	ADA		71.5±0.2	67.6±2.6	84.0±0.6	77.0±2.3
		+ CON	71.5±2.9	71.0±2.8	84.9±0.7	81.0±2.1

Table 3.6: The result of volitionality classification and subject animacy classification in English. The bold scores indicate the highest ones over models, and the underlined scores indicate the highest ones over models trained without joint learning.

learning, the models trained with ADA often performed best.

As for subject animacy classification, the effective method depended on language. This is likely because Japanese is a pro-drop language while English is not. In Japanese, the most effective method was WR. Learning subject animacy of events produced by WR can be interpreted as learning animacy of omitted subjects. Events with omitted subjects were not given a subject animacy label by our labeling method and thus were in $\mathcal{D}^u_{\text{ani}}$. The models trained with WR successfully generalized to such events. In English, on the other hand, because subjects are not omitted, WR was not as effective as in Japanese.

We observed that the overall scores on the English datasets were lower than the Japanese ones. The reason was the quality of the evaluation datasets. As Table 3.3 suggests, the English evaluation datasets were constructed by crowdworkers with a lower agreement rate. Investigating the output manually, we found that the performance was underestimated due to labeling mistakes in the gold data.

3.6 Analysis

3.6.1 Qualitative Analysis

We investigated what is learned by our method, using the model that best performed on $\mathcal{D}^u_{\text{vol}}$. While we had three models trained with the same setting with different random seeds, we used one that achieved the second-best validation performance for analysis.

Japanese The best performing model learned volitionality with SOC, subject animacy with WR, and prediction consistency with CON. We found that the model was aware of context. Example (12-a) and (12-b) were successfully classified as volitional and non-volitional, respectively, though these events had the same predicate. Example (13-a) and (13-b) were again correctly classified as non-volitional and volitional, respectively, considering the meaning of the adverb. We observed that subject animacy was also recognized considering context; the subjects of example (14-a) and (14-b) were successfully classified as inanimate and animate, respectively. This result suggests the effectiveness of our method to

learn volitionality and subject animacy considering context. It would be interesting to quantitatively evaluate such context-awareness by constructing a dataset like Winograd Schema Challenge (Levesque et al., 2012).

However, we found that there still existed verbs that our model struggled with recognizing the volitionality. One notable verb was "*iru* (exist/stay)." While the verb "*iru* (exist/stay)" basically represents a state, it can represent a volitional action when the subject is animate. We speculate that the difficulty of recognizing the animacy of omitted subjects also contributed to this problem. A plausible solution is to consider the preceding and following events during training. If the meaning of an event is different, the distribution of its surrounding events should be too. Learning such contextual differences could lead to better performance.

English The best performing model learned volitionality with SOC, subject animacy with SOC, and prediction consistency with CON. We again found that the model successfully performed classification considering context. For example, the following examples with the same predicate "made" were correctly classified.

(24) a. I made pancakes. (V)
b. I made a mistake. (NV)

The following examples were also successfully classified, capturing the meaning of the adverbial phrase "for him."

(25) a. I tumbled. (NV)
b. I tumbled for him. (V)

3.6.2 Effect of the Choice of Volitionality Indicating Words

The proposed method requires manual preparation of volitionality indicating words in order to assign volitionality labels to events. As an analysis, using the Japanese dataset, we investigated the effect of the number of volitionality indicating words on the performance on $\mathcal{D}^{u}_{\text{vol}}$. Specifically, we explored the performance when using the top 1, 2, 4, 8, and 15 volitionality indicating words. The number of labeled data obtained from a raw corpus of the same size in-

	Top 1	Top 2	Top 4	Top 8	Top 15	Top 15*
Baseline	74.3±3.7	83.9±0.5	85.9±5.9	87.3±2.0	88.0±0.8	89.5±1.1
Ours	93.2±0.2	93.3±0.2	94.2±0.6	95.1±0.7	95.5±0.6	96.7±0.4

Table 3.7: Classification performance on Japanese $\mathcal{D}^u_{\text{vol}}$ when using different numbers of volitionality indicating words. The evaluation metric is AUC. The mean and variance of three runs with different random seeds are described. Top15* is the result when all data are used without down-sampling, which is deribed from Table 3.5.

creases as the number of volitionality indicating words increases. However, in order to examine the effect of the number of volitionality indicating words on performance, we down-sampled the obtained labeled data to match the number of the labeled data obtained when using the top 1 frequency adverb, which was 23,408 (the sum of 5,293 events matching "*aete* (deliberately)" and 18,115 events matching "*omowazu* (unexpectedly)"). As models, we used two settings: a naive setting (**Baseline**) in which only labeled data for volitionality is trained without regularization, and a setting (**Ours**) in which volitionality is trained with SOC, subject animacy with WR, and prediction consistency with CON, which achieved the highest accuracy in $\mathcal{D}^u_{\text{vol}}$ in the setting using all data. The training settings are the same as those described in Section 3.5.3.

Table 3.7 shows the results. We observed that in both models, the performance of $\mathcal{D}^u_{\text{vol}}$ is improved by increasing the number of volitionality indicating words. This can be attributed to the fact that increasing the number of volitionality indicating words increases the diversity of the labeled data and allows us to learn inferences that generalize to many events. Besides, surprisingly, the proposed method achieved a high classification performance of 93.2 points AUC even when only "*aete* (deliberately)" and "*omowazu* (unexpectedly)," which were the most frequent adverbs among the selected adverbs, were used for labeling. This suggests that "*aete* (deliberately)" and "*omowazu* (unexpectedly)" are adverbs that appear in a variety of contexts, and that volitionality classification can be fairly learned from events containing these adverbs alone.

	Label	Japanese	English
$\mathcal{D}^l_{\text{vol}}$	Volitional	88%	94%
	Non-volitional	92%	80%
$\mathcal{D}^l_{\text{ani}}$	Animate	81%	96%
	Inanimate	72%	76%

Table 3.8: The ratio of events being given a correct label.

3.6.3 Quality of Labeled Data

Because we had heuristically and automatically assigned labels to events, our labeled datasets should contain wrongly labeled events. However, if the datasets were full of errors, it is likely to fail to learn classification.

Given the fact that we could learn a classifier with fairly good performance, we report the quality of our labeled data as a reference for applying our method to other languages. We randomly extracted 100 unique positive and negative events from each of $\mathcal{D}^l_{\text{vol}}$ and $\mathcal{D}^l_{\text{ani}}$, and manually examined whether they were given a correct label or not. We considered that events incomprehensible for some reason (e.g., parsing error) were not given a correct label.

Table 3.8 shows the result. We found that most events were labeled correctly. Japanese negatively-labeled events in $\mathcal{D}^l_{\text{ani}}$ had relatively low accuracy. This was primarily because of the failure in subject recognition. In English, negatively-labeled events in $\mathcal{D}^l_{\text{ani}}$ had relatively low accuracy. While there were several reasons, one of them was that, although we regarded nouns representing a location as inanimate, they sometimes represented an organization (e.g., country name).

3.7 Summary of This Chapter

This paper focused on the close relationship between volitionality and animacy and proposed a method to jointly learn them with regularization in a minimally-supervised manner. Experiments in Japanese and English showed the effectiveness of the proposed method to learn volitionality and subject animacy without

manually labeled data.

While volitionality is a fundamental property of events and has many applications, there is a handful of studies that employ volitionality classification. This is because there is no off-the-shelf volitionality classifier and no method to construct a volitionality classifier at a low cost. In this sense, our method has the potential to promote future research.

While we focused on volitionality, the general idea behind our method is potentially applicable to learn other event properties such as sentiment polarity. It is an interesting direction to apply the proposed method to other event classification tasks.

Chapter 4

Discourse Relation Analysis

Discourse relation analysis is the task of predicting the pragmatic relation between two events. After neural networks are introduced to this task, researchers have considered better neural network architectures, incorporation of external knowledge, knowledge transfer from explicit discourse relations, etc. However, the current state-of-the-art method is to fine-tune a general-purpose language model pretrained on a large-scale raw corpus in a self-supervised manner, which does not use the above techniques. This means that, by designing an appropriate self-supervised task, it is possible to learn event representations capturing discourse relations from raw text. Given that, we propose a novel method to learn contextualized and generalized sentence representations based on contrastive self-supervised learning, which can be used at the event level. In the proposed method, a model is given a text consisting of multiple sentences. One sentence is randomly selected as a target sentence. The model is trained to maximize the similarity between the representation of the target sentence with its context and that of the masked target sentence with the same context. Simultaneously, the model minimizes the similarity between the latter representation and the representation of a random sentence with the same context. We apply our method to discourse relation analysis in English and Japanese and show that it outperforms strong baseline methods based on BERT, XLNet, and RoBERTa.

4.1 Introduction

Discourse relation analysis is the task of predicting the pragmatic relation between two events. Discourse relation analysis provides a high-level linguistic structure, which helps many crucial downstream tasks, including automatic summarization (Louis et al., 2010; Huang and Kurohashi, 2021), sentiment analysis (Somasundaran et al., 2009), and machine translation (Meyer et al., 2015). Given the wide range of applications, discourse relation analysis has long been considered an important language analysis task.

The difficulty of discourse relation analysis varies greatly depending on the presence or absence of discourse markers, which are words that explicitly indicate a discourse relation, such as *because* and *however*. Early studies developed a list of discourse markers for each discourse relation and recognized discourse relations based on the list. This method can achieve high precision, however, it is helpless in recognizing discourse relations where discourse markers do not exist.

Therefore, recent research focuses on the analysis of discourse relations without discourse markers, which requires deep semantic understanding; this task is called implicit discourse relation analysis. After neural networks are introduced to this task, in order to improve the performance, researchers have considered better neural network architectures (Chen et al., 2016; Liu and Li, 2016; Bai and Zhao, 2018), incorporation of external knowledge (Kishimoto et al., 2018), knowledge transfer from explicit discourse relations (Rutherford et al., 2017; Qin et al., 2017), etc. However, the current best method is to fine-tune a general-purpose language model pretrained on a large-scale raw corpus in a self-supervised manner (Kim et al., 2020; Devlin et al., 2019), which does not employ above techniques. This means that, by designing an appropriate self-supervised task, it is possible to learn event representations capturing discourse relations from raw text.

Consequently, we work on exploring self-supervised learning frameworks to obtain an event representation that is effective for performing discourse relation analysis. In NLP, there is a rich body of work on sentence representation learning. An event is an information unit corresponding to a simple sentence, or a clause. Therefore, we consider event representation learning as a problem of sentence

representation learning.

Sentence representation have been one of the main interests of natural language processing. While early studies employed symbol-based representations such as bag-of-words, recent studies use distributed representations due to its ability to capture the various and complex properties of sentences (Conneau et al., 2017; Arora et al., 2017; Kiros et al., 2015).

One typical way to obtain distributed sentence representations is to learn a task that is somehow related to sentence meaning. For example, sentence representations trained to solve natural language inference (Bowman et al., 2015; Williams et al., 2018) are known to be helpful for many language understanding tasks such as sentiment analysis and semantic textual similarity (Conneau et al., 2017; Wieting and Gimpel, 2018; Cer et al., 2018; Reimers and Gurevych, 2019).

However, there is an arbitrariness in the choice of tasks used for training. Furthermore, there is a size limitation on manually annotated data, which makes it hard to learn a wide range of language expressions.

A solution to these problems is self-supervised learning, which has been used with great success (Mikolov et al., 2013; Peters et al., 2018; Devlin et al., 2019). For example, inspired by skip-grams (Mikolov et al., 2013), proposed to train a sequence-to-sequence model to generate sentences before and after a sentence, and use the trained encoder to compute sentence representations. Inspired by masked language modeling in BERT, Zhang et al. (2019) and Huang et al. (2020) presented methods to learn contextualized sentence representations through the task of restoring a masked sentence from its context.

In self-supervised sentence representation learning, sentence generation is typically used as its objective. Such an objective aims to learn a sentence representation specific enough to restore the sentence, including minor details. On the other hand, in case we would like to handle the meaning of a larger block such as paragraphs and documents (which is often called context analysis) and consider sentences as a basic unit, a more abstract and generalized sentence representation would be helpful.

We propose a method to learn contextualized and generalized sentence representations by contrastive self-supervised learning (van den Oord et al., 2019;

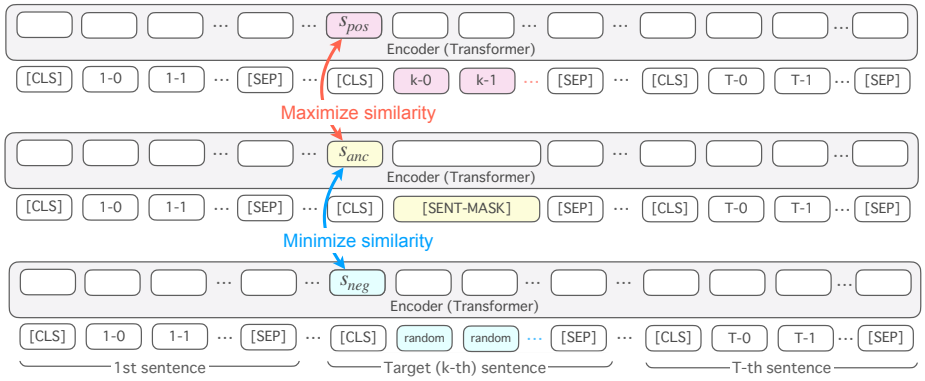

Figure 4.1: Overview of our contrastive method to learn sentence representations. The encoder takes a text consisting of multiple sentences. Each [CLS] token represents the following sentence. We maximize the similarity between s_{anc} and s_{pos}, where s_{anc} is the representation of the k-th sentence computed from the context, and s_{pos} is the representation of the k-th sentence computed by observing the content. Simultaneously, we minimize the similarity between s_{anc} and s_{neg}, where s_{neg} is the representation of a random sentence with the same context.

Chen et al., 2020). Figure 4.1 shows the overview of our method. In the proposed method, a model is given a text consisting of multiple sentences and computes their contextualized sentence representations. During training, one sentence is randomly selected as a *target sentence*. The model is trained to maximize the similarity between the representation of the target sentence with its context, to which we refer as s_{pos}, and the representation of the masked target sentence with the same context, to which we refer as s_{anc}. Simultaneously, the model is trained to minimize the similarity between the latter representation s_{anc} and the representation of a random sentence with the same context as the target sentence, to which we refer as s_{neg}.

From the viewpoint of optimizing s_{anc}, this can be seen as a task to capture a generalized meaning that contextually valid sentences commonly have, utilizing s_{pos} and s_{neg} as clues. From the viewpoint of optimizing s_{pos}, this can be seen as a task to generalize the meaning of a sentence to the level of s_{anc}.

We show the effectiveness of the proposed method using discourse relation

analysis as an example task of context analysis. Our experiments on English and Japanese datasets show that our method outperforms strong baseline methods based on BERT (Devlin et al., 2019), XLNet (Yang et al., 2019), and RoBERTa (Liu et al., 2019).

4.2 Related Work

As related works, we describe discourse relation analysis, sentence representations, and contrastive learning.

4.2.1 Discourse Relation Analysis

Discourse relation analysis is the task of predicting the pragmatic relation between two events. Discourse relations provide a high-level linguistic structure, which helps many crucial downstream tasks, including automatic summarization (Louis et al., 2010; Huang and Kurohashi, 2021), sentiment analysis (Somasundaran et al., 2009), and machine translation (Meyer et al., 2015). Given the wide range of applications, discourse relation analysis has long been considered an important language analysis task.

The difficulty of discourse relation analysis varies greatly depending on the presence or absence of discourse markers, which are words that explicitly indicate a discourse relation, such as *because* and *however*. Early studies developed a list of discourse markers for each discourse relation and recognized discourse relations based on the list. This method can achieve high precision, however, it is helpless in recognizing discourse relations where discourse markers do not exist.

Recent research focuses on implicit discourse relation analysis, where models cannot exploit discourse markers. After neural networks are introduced to this task, in order to improve the performance, researchers have considered better neural network architectures (Chen et al., 2016; Liu and Li, 2016; Bai and Zhao, 2018), incorporation of external knowledge (Kishimoto et al., 2018), knowledge transfer from explicit discourse relations (Rutherford et al., 2017; Qin et al., 2017), etc. The current best method is to fine-tune a general-purpose language model pretrained on a large-scale raw corpus in a self-supervised manner (Kim et al.,

2020; Devlin et al., 2019), which does not employ above techniques.

4.2.2 Distributed Sentence Representations

Sentence representations have been a primary concern of NLP as the basis of sentence-level semantic analysis. While early studies employed symbol-based representations such as bag-of-words, recent studies employ distributed vector representations in order to capture various and complex property of sentences flexibly.

Most of the existing studies focus on encoding a single sentence into a distributed representation. Approaches proposed in this line of research can be divided into two approaches. One approach is to compute sentence representations from the representations of the constituent words. The most straightforward method is to average the word embeddings of the constituent words of a sentence (Wieting et al., 2016). As a more sophisticated way, Arora et al. (2017) propose to compute weighted average of word embeddings according to the word frequency and then remove the projections of the average vectors on their first singular vector. Mu et al. (2018) propose to first modify pre-trained word embeddings by removing the mean vector and projecting the representations away from the most dominating directions, and then compute a sentence representation by averaging modified word embeddings.

The other approach is to learn sentence representations directly through solving a task that requires sentence-level semantic understanding. For example, natural language inference (NLI) is known to be effective to learn sentence representations that are helpful to solve many language understanding tasks. Conneau et al. (2017), Cer et al. (2018), and Reimers and Gurevych (2019) trained a single sentence encoder on a NLI dataset, while they employed different neural architectures for sentence encoders.

However, there is an arbitrariness in the choice of tasks used for training. Besides, there is a size limitation on manually annotated data, which makes it hard to learn a wide range of language expressions.

Self-supervised learning is one of the solutions to this problem. Self-supervised methods train models by supervision obtained from data itself, and thus can effectively exploit a huge amount of existing unlabeled data. Skip-gram (Zhu

et al., 2015) is a self-supervised method to learn sentence representations. Skip-gram learns to encode a sentence into a vector representation so that the following and preceding sentences can be generated from it. FastSent (Hill et al., 2016) is a light-weight alternative to Skip-gram; instead of generating sentences, it predicts the bag-of-words.

Recent advances in NLP have made it possible to understand meaning with consideration of context, prompting research on learning contextualized sentence expressions. HIBERT (Zhang et al., 2019) is a self-supervised method that learns sentence representations so that a masked sentence can be reconstructed from the sentence representations of its contextual sentences. INSET (Huang et al., 2020) is also a similar self-supervised method; it exploits pre-trained general-purpose language models for initializing sentence encoders.

In self-supervised sentence representation learning, sentence generation is typically used as its objective. Such an objective aims to learn a sentence representation specific enough to restore the sentence, including minor details. On the other hand, in case we would like to handle the meaning of a larger block such as paragraphs and documents (which is often called context analysis) and consider sentences as a basic unit, a more abstract and generalized sentence representation would be helpful. This motivates the proposed method of learning sentence expressions using a non-generative, contrastive objective function.

4.2.3 Contrastive Learning

Contrastive learning is a general framework for representation learning, and has been used in many fields including computer vision, audio processing, and natural language processing. The main idea is to encode similar data into similar representations while encoding different data into different representations. What is considered similar depends on the purpose. For example, Chen et al. (2020) propose a contrastive learning framework for visual representations that considers random transformations of the same image (e.g., cropping, flipping, distortion and rotation) similar. Image encoders are trained to encode such images into a similar representation.

In the context of sentence representation learning, there are several studies

that use contrastive learning. Iter et al. (2020) propose to encode neighboring sentence into a similar representations. Yan et al. (2021) propose to perform data augumentation, such as adversarial attack and token shuffling, and make generated sentences closer to each other. Gao et al. (2021) propose to encode the same sentences with different dropout masks into the same representation.

These methods consider a model that is given a single sentence or a sentence pair, however, in this study, we consider a model that is given a document or paragraph consisting of multiple sentences. To our knowledge, we are the first to incorporate contrastive learning to obtain contextualized sentence representations.

4.3 Learning Contextualized Sentence Representations

Figure 4.1 illustrates the overview of our method. The encoder takes an input text consisting of T (> 1) sentences and computes their contextualized sentence representations. The encoder is trained by optimizing our contrastive self-supervised objective.

4.3.1 Encoder

The encoder is a Transformer (Vaswani et al., 2017), which is initialized with BERT (Devlin et al., 2019) or RoBERTa (Liu et al., 2019). The Transformer computes token representations using a mechanism called self-attention, which is an attention mechanism relating different positions of a single sequence in order to compute a representation of the same sequence. Given sufficiently large training data, the Transformer outperforms the other text encoders such as LSTMs and CNNs in many language tasks.

Following Liu and Lapata (2019), we insert the [CLS] and SEP special tokens at the beginning and the end of each sentence, respectively. The representation of the [CLS] token is used as the sentence representation of its following sentence. Although a [CLS] special token is to represent its following sentence, we do not apply an attention mask to restrict the [CLS] special token from accessing to the tokens in the other sentences. In other words, sentence representations are computed by considering all the tokens in the given document via the self-attention

mechanism of the Transformer.

4.3.2 Contrastive Objective

We propose a contrastive objective to learn contextualized sentence representations, aiming to capture the generalized meaning of each sentence in the given text. Figure 4.1 shows the overview.

We first randomly select one sentence from the input text as a *target sentence*. In Figure 4.1, the k-th sentence ($1 \leq k \leq T$) is selected as a target sentence. We refer to the representation of the target sentence as s_{pos}. We then mask the target sentence with the [SENT-MASK] special token. We refer to the representation of the masked sentence as s_{anc}. We finally replace the target sentence with a random sentence. We refer to the representation of the random sentence as s_{neg}. We use the same encoder to compute these sentence representations.

Our contrastive objective is to maximize the similarity between s_{pos} and s_{anc} while minimizing the similarity between s_{neg} and s_{anc}. We use the dot product as the similarity measure. When using N random sentences per input text, the contrastive loss $\mathcal{L}$ is formally written as follows:

$$\mathcal{L}_{\mathrm{CON}} = -\log \frac{\exp(\langle s_{pos}, s_{anc} \rangle)}{\sum_{s \in \mathcal{S}} \exp(\langle s, s_{anc} \rangle)}, \tag{4.1}$$

where $\langle \cdot, \cdot \rangle$ is the dot product and $\mathcal{S} = \{s_{pos}, s^1_{neg}, \cdots, s^N_{neg}\}$.

In order to optimize s_{anc}, the encoder needs to make s_{anc} closer to s_{pos} than s_{neg}. This can be interpreted as training to capture a generalized meaning that contextually valid sentences commonly have. On the other hand, in order to optimize s_{pos}, the encoder needs to make it closer to s_{anc}. This can be interpreted as training to discard minor details that cannot be estimated from the context and capture the generalized meaning. Our contrastive learning optimizes these objectives simultaneously.

The encoder is trained by jointly optimizing the contrastive loss and the masked language modeling loss (Devlin et al., 2019). Masked language modeling is the fundamental training objective of BERT and its variants (Devlin et al., 2019; Liu et al., 2019). We learn masked language modeling expecting that sen-

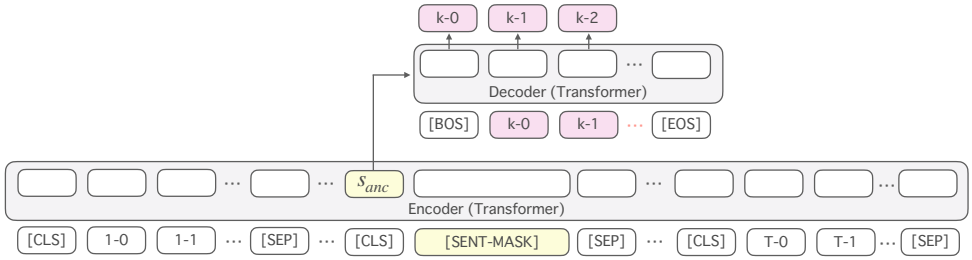

Figure 4.2: Overview of the generative method to learn sentence representations. When learning the generative objective, one of the sentences is masked with the [SENT-MASK] special token. In this figure, k-th sentence is masked. The encoder computes s_{anc}, which is the masked sentence representation. The decoder is trained to generate the masked sentence from s_{anc}.

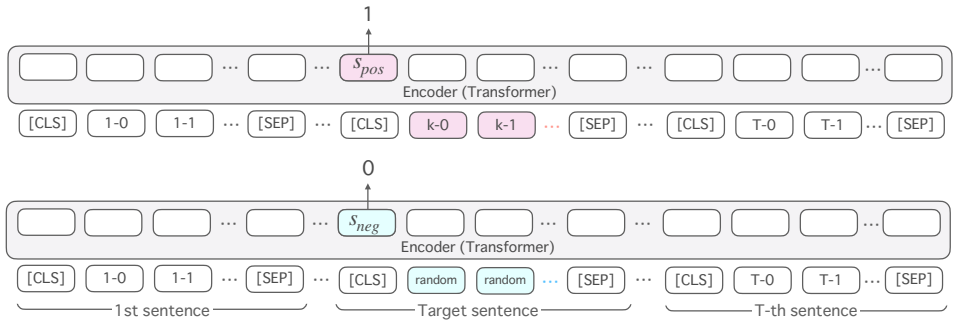

Figure 4.3: Overview of the detective method to learn sentence representations.

tence representations are learned while keeping the parameter structure acquired through pretraining.

4.3.3 Generative Objective

For comparison, we train the encoder through the task of generating a masked sentence from its context. Figure 4.2 illustrates the overview.

We first mask a sentence with the [SENT-MASK] special token. In Figure 4.2, k-th sentence is masked. The encoder computes the representation of the masked sentence. Following the contrastive method, we refer to the representation as s_{anc}. Then, given the representation, a decoder generates the masked sentence

in an auto-regressive manner. The decoder's architecture is almost the same as the encoder, but it has an additional layer on the top to predict a probability distribution over words in the vocabulary. We use teacher forcing and compute the generative loss by summing cross-entropy at each generation step.

Again, the encoder and decoder are trained by optimizing the generative loss and the standard masked language modeling loss jointly.

4.3.4 Detective Objective

As an alternative non-generative objective, we train the encoder through the task of detecting an inserted random sentence. Figure 4.3 shows the overview.

In this method, we additionally train a linear classifier that is a given sentence representation and determines whether the sentence is a replaced one or not. The detective objective can be interpreted as a method that replaces s_{anc} in the contrastive objective with a context-agnostic trainable parameters.

Again, the encoder and decoder are trained by optimizing the generative loss and the standard masked language modeling loss jointly.

4.3.5 Implementation Details

English

We used an English Wikipedia dump and BookCorpus (Zhu et al., 2015)[1] to create input texts. We first split texts into sentences using spacy (Honnibal et al., 2020). We then extracted as many consecutive sentences as possible so that the length does not exceed the maximum input length of 128. When a sentence was so long that an input text including the sentence cannot be created while meeting the length constraint, we gave up using the sentence. The number of sentences in an input text T was 4.91 on average. After creating input texts, we assigned random sentences to each of them. Random sentences are extracted from the same document. We assigned three random sentences per input text, i.e., $N = 3$.

[1]Because the original BookCorpus is no longer available, we used a replica created by a publicly available crawler (`https://github.com/soskek/bookcorpus`).

In order to learn masked language modeling, we masked 15% of tokens. We dynamically selected masked tokens following Liu et al. (2019).

We initialized the encoder's parameters using the weights of $\text{RoBERTa}_{\text{BASE}}$ (Liu et al., 2019). The other parameters were initialized randomly. We trained the model for 10,000 steps with a batch size of 512. We used the Adam optimizer (Kingma and Ba, 2015) with a learning rate of 2e-5, $\beta_1 = 0.9$, $\beta_2 = 0.999$, linear warmup of the learning rate over the first 1,000 steps, and linear decay of the learning rate.

Japanese

We used a Japanese Wikipedia dump to create input texts. We split the texts into clauses using KNP, a widely used Japanese syntactic parser (Kawahara and Kurohashi, 2006). We created input texts and assign random sentences to them in the same way as in Section 4.3.5. The number of sentences (clauses) in an input text T was 6.42 on average.

We initialized the encoder's parameters with $\text{BERT}_{\text{BASE}}$, pretrained on a Japanese Wikipedia dump[2]. The other details were the same as in Section 4.3.5.

4.4 Discourse Relation Analysis

We show the effectiveness of the proposed method using discourse relation analysis as a concrete example of context analysis. Discourse relation analysis is a task to predict the logical relation between two arguments. An argument roughly corresponds to a sentence or a clause. We conduct experiments on English and Japanese datasets.

4.4.1 Datasets

Penn Discourse Tree Bank (PDTB) 3.0

PDTB 3.0 is a corpus of English newspaper with discourse relation labels (Prasad et al., 2018). We focus on implicit discourse relation analysis, where no explicit

[2]Available at `https://alaginrc.nict.go.jp/nict-bert/index.html`.

discourse marker exists. Following Kim et al. (2020), we use the Level-2 labels with more than 100 examples and use 12-fold cross-validation.

Kyoto University Web Document Leads Corpus (KWDLC)

KWDLC is a Japanese corpus consisting of leading three sentences of web documents with discourse relation labels (Kawahara et al., 2014; Kishimoto et al., 2018, 2020). As KWDLC does not discriminate between implicit discourse relations and explicit discourse relations, we target both. KWDLC has seven types of discourse relations, including NoRelation. The evaluation protocol is 5-fold cross-validation. Following Kim et al. (2020), each fold is split at the document level rather than the individual example level.

4.4.2 Model

We train two types of models; one uses the context of arguments, and the other does not.

When a model uses context, the model is given the paragraph that contains arguments of interest. Figure 4.4 shows the overview. In this setting, first, the paragraph is split into sentences. Arguments are treated as a single sentence, and their context is split in the way described in Section 4.3.5. Then, an encoder computes the representation of each sentence in the same manner as in Section 4.3.1. Given the concatenation of the arguments' representations, a relation classifier predicts the discourse relation. As a relation classifier, we employ a multi-layer perceptron with one hidden layer and ReLU activation.

When a model does not use context, the model is given arguments of interest only. Figure 4.5 shows the overview. In this setting, we use the sentence pair classification method proposed by Devlin et al. (2019).

Our proposed method is introduced to a model which uses context by initializing its encoder's parameters using our sentence encoder. In experiments, we report a difference in performance depending on models used for initialization.

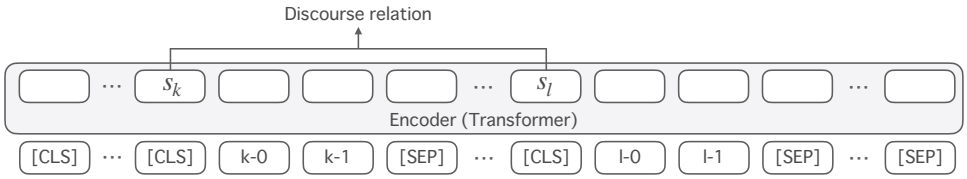

Figure 4.4: Overview of the model that uses context. When two arguments are k-th and l-th sentences, their sentence representations are concatenated and fed into the discourse relation classifier.

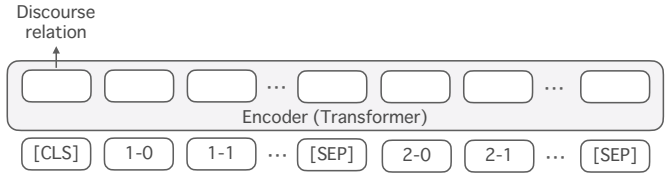

Figure 4.5: Overview of the model that does not use context. Following Devlin et al. (2019), two arguments are concatenated with the special [CLS] and [SEP] tokens and fed into the encoder. The discourse relation is decided from the representation of the [CLS] token.

4.4.3 Implementation Details

Input texts are truncated to the maximum input length of 512, which is long enough to hold almost all inputs. We train models for up to 20 epochs. At the end of each epoch, we compute the performance for the development data and adopt the model with the best performance. If the performance does not improve for five epochs, we stop the training. We use the Adam optimizer with a learning rate of 2e-5, $\beta_1 = 0.9$, $\beta_2 = 0.999$. We update all the parameters in models, i.e., pretrained sentence encoders are fine-tuned to solve discourse relation analysis.

4.4.4 Results

Table 4.1 shows the result for PDTB 3.0. The evaluation metric is accuracy. The highest performance was achieved by the proposed method. To our knowledge, this is the state-of-the-art performance among models with the same parameter size as $\text{BERT}_{\text{BASE}}$. The model that optimized the generative objective was inferior

Context	Encoder	Acc
Unused	$\text{BERT}_{\text{BASE}}$ (Kim et al., 2020)	57.60
	$\text{XLNet}_{\text{BASE}}$ (Kim et al., 2020)	60.78
	$\text{RoBERTa}_{\text{BASE}}$	61.68±1.63
Used	$\text{BERT}_{\text{BASE}}$	56.83±1.43
	$\text{RoBERTa}_{\text{BASE}}$	62.25±1.47
	$\text{RoBERTa}_{\text{BASE}}$+Gen	62.19±1.33
	$\text{RoBERTa}_{\text{BASE}}$+Det	62.59 ± 1.47
	$\text{RoBERTa}_{\text{BASE}}$+Con (ours)	**63.30±1.42**

Table 4.1: Results of implicit discourse relation analysis on PDTB 3.0 using the Level-2 label set (Kim et al., 2020). **Gen**, **Det** and **Con** indicate that the encoder is pretrained by optimizing the generative, detective and contrastive objectives, respectively. The scores are the mean and standard deviation over folds.

not only to the proposed method but also to the vanilla RoBERTa model with context. The model optimized with the detective objective was slightly better than the vanilla RoBERTa model, but inferior to the proposed method.

Table 4.2 shows the result for KWDLC. The evaluation metrics are accuracy and micro-averaged precision, recall, and F1[3]. The highest performance was again achieved by the proposed method. The decrease in performance by optimizing the generative objective is consistent with the experimental results on PDTB 3.0.

4.4.5 Qualitative Analysis

We show several examples of discourse relation analysis in KWDLC.

(26) [$_{\text{Arg1}}$ *ninshin-no shindan-o uketara*] [*kodomo ikusei-ka-ni todokedete-kudasai.*] [$_{\text{Arg2}}$ *shussho-todoke-ya kenshin-nado kongo-no boshi-no kenko-kanri-de hitsuyo-na boshi-kenko-techo-nado-o o-watashi-shimasu.*]

[3]As examples with the discourse relation of NoRelation accounts for more than 80% of the dataset, precision, recall, and F1 are calculated without examples with NoRelation to make performance difference intelligible.

Context	Encoder	Acc	Prec	Rec	F1
Unused	BERT	80.68±1.59	45.90±4.06	**41.42±8.35**	43.37±6.28
Used	BERT	84.36±2.05	62.55±10.26	39.13±7.38	47.67±6.68
	BERT+Gen	84.16±1.60	57.84±8.51	40.13±0.42	47.21±2.68
	BERT+Det	83.89±1.73	61.53±4.52	40.37±6.27	47.43±3.15
	BERT+Con (ours)	**85.02±1.85**	**63.51±5.90**	41.04±4.24	**49.74±4.11**

Table 4.2: Results of discourse relation analysis on KWDLC. The scores are the mean and standard deviation over folds.

> [$_{\text{Arg1}}$ If you are diagnosed as pregnant,] [contact the childcare division.] [$_{\text{Arg2}}$ You will be given the maternity health record book and other documents necessary for future health care.]
> **Label**: NoRelation

Sentences (clauses) are enclosed in [and]. The first argument is marked by "Arg1," and the second argument is marked by "Arg2." Without context, models wrongly predicted the discourse relation of Condition. This prediction is reasonable as the models did not know that another sentence exists between the arguments. Considering context reduces such false positive predictions, which led to improving the performance of models using context. In PDTB 3.0, as implicit discourse relation labels are assigned only to adjacent two arguments, the performance improvement brought by considering context was relatively small compared to the one in KWDLC.

However, even with context, the models that did not learn our contrastive objective erroneously predicted the discourse relation of Condition. We conjecture that this is because the models relied on the discourse marker of Condition, "*tara* (if)." Only the model that learned our contrastive objective was able to predict the correct discourse relation, NoRelation.

We show another improved example.

(27) [$_{\text{Arg1}}$ *Niigata-ken-ni aru kokuei koen echigo kyuryo koen-e, ippaku-de asobi-*

ni dekakeyo-to] [$_{\text{Arg2}}$ *omoitachi-mashita.*]
[$_{\text{Arg1}}$ I want to go to a government-managed park in Niigata Prefecture for an overnight visit,] [$_{\text{Arg2}}$ I came up with that.]
Label: NoRelation

The models except ours wrongly predicted the discourse relation of Purpose between Arg1 and Arg2. This is probably because the Japanese postpositional particle *to* can be a discourse marker of Purpose. For example, if Arg2 were "*nizukuri-o hajimeta* (I started packing)," the prediction would be correct. However, in this case, the postpositional particle *to* is used to construct a sentential complement. That is, Arg1 is the object of Arg2. It is not possible to distinguish between the two usages from its surface form. Our model correctly predicted the discourse relation of NoRelation, which implies that our method was able to understand that Arg1 is a sentential complement.

We show yet another improved example of implicit discourse relation analysis in KWDLC.

(28) [$_{\text{Arg1}}$ *izen-kara keikaku-shiteita homupeji-o kaisetsu-suru-koto-ga-deki,*] [$_{\text{Arg2}}$ *ureshii kagiri-dearu.*]
[$_{\text{Arg1}}$ I was able to launch the website that I had planned for a while,] [$_{\text{Arg2}}$ I'm happy.]
Label: Cause/Reason

While most models predicted the discourse relation of NoRelation between Arg1 and Arg2, the proposed model correctly recognized the discourse relation of Cause/Reason. We speculate that the models other than ours failed to understand Arg1 at the level of "a happy event occurred."

Finally, we show an example that no models were able to correctly predict the discourse relation.

(29) [*kyo-wa shinsaku baggu & semioda-no katto-nado-o shi-mashita.*] [$_{\text{Arg1}}$ *juni-gatsu-no sakusei-bun-o mou hajimete-imasu.*] [*nenmatsu girigirini-natte-simatte*] [$_{\text{Arg2}}$ *batabata-suru-no-ga iya-nanode.*]

[Today, I cut the new bags and semi-custom ordered ones.] [$_{\text{Arg1}}$ I've already started work for December.] [At the end of the year,] [$_{\text{Arg2}}$ I don't want to be in a rush.]
Label: CAUSE/REASON

No models were not able to capture the discourse relation of CAUSE/REASON between Arg1 and Arg2. This is possibly due to the fact that our sentence representations were trained on Wikipedia texts and hardly capture the discourse relations in diaries like this. We observed many errors when texts were very different from those in Wikipedia, such as essay, diary, and advertisement. We expect that such errors can be mitigated by using texts obtained from wide domains for pretraining sentence representations.

4.5 Sentence Retrieval

To investigate what is learned by our contrastive objective, we did sentence retrieval based on the similarity between sentence representations. For targets, we randomly sampled 500,000 sentences with context from input texts used for training. For a query, we used a sentence with its context in a Wikipedia article. Computing the sentence representations for the targets and query, we searched the closest sentences based on their cosine similarity.

Table 4.3 shows an example. In addition to the top-ranked sentences, we also picked up some highly-ranked sentences. The top two sentences were very similar to the query sentence regarding the topic, meaning, and context. While the sentences of lower rank had different topics from the query sentence, they all described a positive aspect of an entity and had a similar context in terms of that an entity is introduced in their preceding sentences.

We confirmed that almost the same results were obtained in Japanese. Table 4.4 shows an example of sentence retrieval in Japanese. Again, we observed that the top-ranked sentence was very similar to the query sentence regarding the topic, meaning, and context, and the sentences of lower rank were similar to the query sentence at a generalized level and had a similar context.

In order to investigate the degree of generalization of sentence expressions

Query:	[The Beatles were an English rock band formed in Liverpool in 1960.] [**The group, whose best-known line-up comprised John Lennon, Paul McCartney, George Harrison and Ringo Starr, are regarded as the most influential band of all time.**]
Retrieved:	1) [Britney Jean Spears (born December 2, 1981) is an American singer, songwriter, dancer, and actress.] [**She is credited with influencing the revival of teen pop during the late 1990s and early 2000s, for which she is referred to as the "Princess of Pop".**] ... 2) [Dynasty was an American band, based in Los Angeles, California, created by producer and SOLAR Records label head Dick Griffey, and record producer Leon Sylvers III.] [**The band was known for their dance/pop numbers during the late 1970s and 1980s.**] ... 3) [Lu Ban (–444BC) was a Chinese structural engineer, inventor, and carpenter during the Zhou Dynasty.] [**He is revered as the Chinese god (patron) of builders and contractors.**] ... 10) [Stacey Park Milbern (May 19, 1987 – May 19, 2020) was an American disability rights activist.] [**She helped create the disability justice movement and advocated for fair treatment of people with disabilities.**] ... 20) [The National Action Party (, PAN) is a conservative political party in Mexico founded in 1938.] [**The party is one of the four main political parties in Mexico, and, since the 1980s, has had success winning local, state, and national elections.**] ...

Table 4.3: Results of sentence retrieval based on the cosine similarity between sentence representations computed by our method. [·] indicates a sentence. The query and retrieved sentences are marked in bold, and their contexts are shown together. The numbers indicate the rank of sentence retrieval.

Query:	[ビートルズは、1960 年代から 1970 年にかけて活動したイギリス・リヴァプール出身のロックバンド。] [**20 世紀を代表する音楽グループである。**]
Retrieved:	1) [ドゥービー・ブラザーズはアメリカ合衆国のロックバンド。] [１９７１年のデビュー以来、解散時期を挟みながらも現在まで第一線で活動し続ける人気グループ。] [**１９６０年代後半から１９７０年代まで、アメリカ音楽界で大きなムーヴメントとなったウェストコースト・ロックを代表するバンドのひとつ。**] 2) [民族革命運動党は、ボリビアの政党。] [**２０世紀のボリビアの歴史で最も重要な役割を果たした政党。**] 10) [株式会社タミヤは、静岡県静岡市に本社を置く模型・プラモデルメーカー。] [**世界有数の総合模型メーカーである。**]

Table 4.4: Results of sentence retrieval in Japanese. The numbers indicate the rank of sentence retrieval. [·] indicates a sentence. The query and retrieved sentences are marked in bold, and their contexts are shown together.

learned by the proposed method qualitatively, it is essential to construct a corpus with a variety of annotations at the sentence level (e.g., emotion polarity, volitionality, discourse function, etc.). We leave this for future work.

4.6 Summary of This Chapter

We proposed a method to learn contextualized and generalized sentence representations using contrastive self-supervised training. Experiments showed that the proposed method improves the performance of discourse relation analysis both in English and Japanese. Besides, through qualitative analysis, we found that our sentence representations are context-sensitive and capture a generalized meaning.

The proposed method can be potentially used for various applications other than discourse relation analysis. For example, extractive summarization and common sense inference are promising applications of the proposed method. An important future work is to investigate the effectiveness of the proposed method for these tasks.

Another future direction is to investigate the degree of generalization of sentence expressions learned by the proposed method. Because the proposed method is a self-supervised method, the level of abstraction of the sentence representations

trained by the proposed method is not obvious. Through qualitative evaluation, we confirmed that the proposed method captures the generalized meaning of sentences and takes the context into account. For a more detailed analysis, a corpus with a variety of annotations at the sentence level (e.g., emotion polarity, volitionality, discourse function, etc.) would be effective. Such a corpus will help to clarify what kind of meaning the proposed method learns from the sentences.

Chapter 5

Next Event Prediction

Typical event sequences are an important class of commonsense knowledge. Formalizing the task as the generation of a next event conditioned on a current event, previous work in event prediction employs sequence-to-sequence (seq2seq) models. However, what can happen after a given event is usually diverse, a fact that can hardly be captured by deterministic models. In this paper, we propose to incorporate a conditional variational autoencoder (CVAE) into seq2seq for its ability to represent diverse next events as a probabilistic distribution. We further extend the CVAE-based seq2seq with a reconstruction mechanism to prevent the model from concentrating on highly typical events. To facilitate fair and systematic evaluation of the diversity-aware models, we also extend existing evaluation datasets by tying each current event to multiple next events. Experiments show that the CVAE-based models drastically outperform deterministic models in terms of precision and that the reconstruction mechanism improves the recall of CVAE-based models without sacrificing precision.

5.1 Introduction

Typical event sequences are an important class of commonsense knowledge that enables deep text understanding (Schank and Abelson, 1975; LoBue and Yates, 2011). Following previous work (Nguyen et al., 2017), we work on the task of generating a next event conditioned on a current event, which we call event pre-

diction. For example, we want a computer to recognize that the event "board bus" is typically followed by another event "pay bus fare" and to generate the latter word sequence given the former.

Early studies memorized event sequences extracted from a corpus and inevitably suffered from low generalization capability and a scalability problem. A promising approach to modeling wide-coverage knowledge is to generalize events by representing them in a continuous space (Granroth-Wilding and Clark, 2016; Nguyen et al., 2017; Hu et al., 2017). Nguyen et al. (2017) generate a next event using the sequence-to-sequence (seq2seq) framework, which was first proposed for machine translation (Bahdanau et al., 2014) and subsequently applied to various NLP tasks including text summarization (Rush et al., 2015; Chopra et al., 2016) and dialog generation (Sordoni et al., 2015; Serban et al., 2016).

One limitation of the simple seq2seq models, which are deterministic in nature, is their inability to take into account an important characteristic of events: What can happen after a current event is usually diverse. For the example of "board bus" mentioned above, "get off bus" as well as "pay bus fare" is a valid next event. The inherent diversity makes it difficult to train deterministic models, and during testing, they can hardly generate multiple next events that are both valid and diverse.

To address this problem, we first propose to incorporate a conditional variational autoencoder (CVAE) into seq2seq models (Kingma et al., 2014; Sohn et al., 2015). As a probabilistic model, the CVAE draws a latent variable, representing the next event, from a probabilistic distribution, and this distribution encodes the diversity of next events.

Through experiments, we found that, as indicated by high precision, the CVAE made learning from diverse training data more effective. However, the outputs of the CVAE-based seq2seq model concentrated on a small number of highly typical events (i.e., low recall), possibly due to the mode-seeking property of variational inference (Bishop, 2006, pp. 466–470). This tendency is also reminiscent of seq2seq models' preference to generic outputs (Sordoni et al., 2015; Serban et al., 2016).

We alleviate this problem by extending the CVAE-based seq2seq model with

a reconstruction mechanism (Tu et al., 2017). During training, the reconstruction mechanism forces the model to reconstruct the input from the hidden states of the decoder. This has an effect of restraining the model from outputting highly typical next events because they make the reconstruction more difficult.

We evaluate the proposed models using two event pair datasets provided by Nguyen et al. (2017). One problem with these datasets is that each current event in the test sets is tied to only one next event. For a fair evaluation of diversity-aware models, we extend the test sets so that each given event has multiple next events.

Experiments show that the CVAE-based seq2seq models consistently outperformed the simple seq2seq models in terms of precision (i.e., validity) without hurting recall (i.e., diversity) while forcing the simple seq2seq models to generate diverse outputs yielded low precision. The reconstruction mechanism consistently improved recall of the CVAE-based models while keeping or even increasing precision. We also confirmed that the original test sets failed to detect the clear differences between the models.

5.2 Related Work

5.2.1 Event Prediction

There is a growing body of work on learning typical event sequences (Chambers and Jurafsky, 2008; Jans et al., 2012; Pichotta and Mooney, 2014; Granroth-Wilding and Clark, 2016; Pichotta and Mooney, 2016; Hu et al., 2017; Nguyen et al., 2017). While early studies explicitly store event sequences in a symbolic manner, a recent approach to this task is to train neural network models that implicitly represent event sequence knowledge as continuous model parameters. In both cases, models are usually evaluated by how well they restore a missing portion of an event sequence. We collectively refer to this task as event prediction.

Event prediction can be categorized into two tasks: classification and generation. In the classification task, a model is to choose one from a pre-defined set of candidates for a missing event. A popular strategy is to rank candidates by similarity with the remaining part of the event sequence. Similarity measures in-

clude point-wise mutual information (Chambers and Jurafsky, 2008), conditional bi-gram probability (Jans et al., 2012), and cosine similarities based on latent semantic indexing and word embeddings (Granroth-Wilding and Clark, 2016). However, for its reliance on pre-defined candidates, the classification approach is constrained by its limited flexibility.

In the generation task, a model is to directly generate a missing event, usually in the form of a word sequence (Pichotta and Mooney, 2016; Hu et al., 2017; Nguyen et al., 2017), although one previous study adopted a predicate-argument structure-based event representation (Weber et al., 2018). Nguyen et al. (2017) worked on the task of generating a next event given a single event, which we follow in this paper. They adopted the seq2seq framework (Sutskever et al., 2014) and investigated how recurrent neural network (RNN) variants, the number of RNN layers, and the presence or absence of the attention mechanism (Bahdanau et al., 2014) affected the performance. Hu et al. (2017) gave a sequence of events to the model to generate the next one. Accordingly, they worked on hierarchically encoding the given event sequence using word-level and event-level RNNs.

All of these models are deterministic in nature and do not take into account the fact that there could be more than one valid next event. For example, both "get off bus" and "pay bus fare" seem to be appropriate next events of "board bus." The inherent diversity makes it difficult to train deterministic models. During testing, they can hardly generate multiple next events that are both valid and diverse.

5.2.2 Conditional Variational Autoencoders

Variational autoencoders (VAEs) are a neural network-based framework to learn probabilistic generation (Kingma and Welling, 2013; Rezende et al., 2014). The basic idea of VAEs is to reconstruct an input y via a latent representation z in a way similar to autoencoders (AEs). While AEs learn the process as deterministic transformation, VAEs adopt probabilistic generation: a VAE encodes y into the probability distribution of z, instead of a point in a low-dimensional vector space. It then reconstructs the input y from z drawn from the posterior distribution. z is assumed to have a prior distribution, for which a multivariate Gaussian distri-

bution is often used. As straightforward extensions of VAEs, conditional VAEs (CVAEs) let probabilistic distributions be conditioned on a common observed variable x (Kingma et al., 2014; Sohn et al., 2015). In our case, x is a current event while y is a next event to predict.

Bowman et al. (Bowman et al., 2016) applied VAEs to text generation. They constructed VAEs using RNNs as its components and found that VAEs with an RNN-based decoder failed to encode meaningful information to z. To alleviate this problem, they proposed simple but effective heuristics: KL cost annealing and word dropout. We also employ these techniques.

If a VAE-based text generation model is conditioned on text, it can be seen as a CVAE-based seq2seq model (Zhao et al., 2017; Serban et al., 2017; Zhang et al., 2016). Since a CVAE learns probabilistic generation, it is suitable for tasks where the output is not uniquely determined according to the input. One of the representative applications of CVAE-based text generation is dialogue response generation, or the task of generating possible replies to a human utterance (Zhao et al., 2017; Serban et al., 2017). Applying CVAEs to next event prediction is a natural choice because the task is also characterized by output diversity.

5.2.3 Diversity-Promoting Objective Functions

In dialogue response generation, seq2seq is known to suffer from the generic response problem: The model often ends up blindly generating uninformative responses such as "I don't know." A popular approach to this problem is to rerank the candidate outputs, which are usually produced by beam search, according to the mutual information with the conversational context (Li et al., 2016).

We notice that the reconstruction mechanism (Tu et al., 2017) serves the same purpose in a more straightforward manner, albeit stemming from a different motivation. The reconstruction mechanism forces the model to reconstruct the input from the hidden states of the decoder. Although it was originally proposed for machine translation to prevent over-translation and under-translation, it could counteract the event prediction model's tendency to concentrate on highly typical outputs.

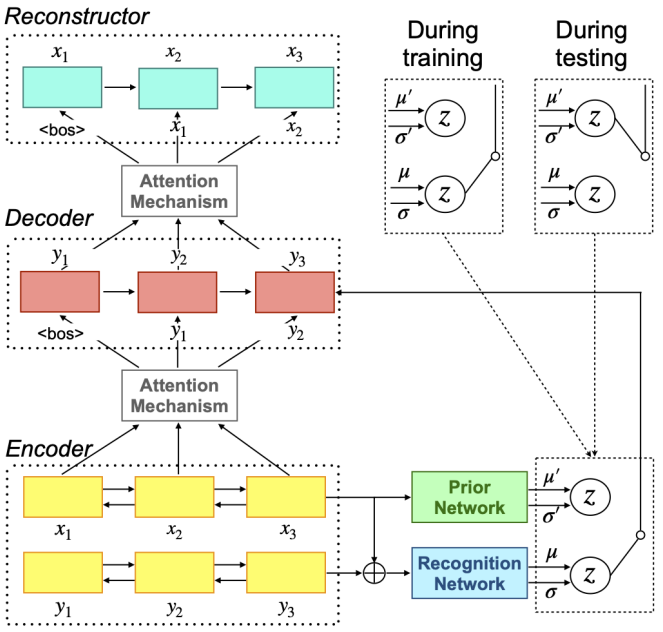

Figure 5.1: The neural network architecture of our event prediction model that uses a CVAE and a reconstruction mechanism. $\oplus$ denotes vector concatenation.

5.3 Problem Setting

Given a current event x, we are to generate a variety of events, each of which, y, often happens after x. x and y are represented by word sequences like "board bus" and "get off bus". Our goal is to learn from training data an event prediction model $p_\theta(y|x)$, where θ is the set of model parameters.

5.4 Conditional VAE with Reconstruction

Figure 5.1 illustrates an overview of our model. To capture the diversity of next events, we use a conditional variational autoencoder (CVAE) based seq2seq model. The CVAE naturally represents diverse next events as a probability distribution. Additionally, we extend the CVAE with a reconstruction mechanism (Tu et al., 2017) to alleviate the model's tendency to concentrate on a small number of highly typical events.

5.4.1 Objective Function

We introduce a probabilistic latent variable z and assume that y depends on both x and z. The conditional log likelihood of y given x is written as:

$$\log p(y|x) = \log \int_z p_\theta(y, z|x) dz \tag{5.1}$$

$$= \log \int_z p_\theta(y|z, x) p_\theta(z|x) dz. \tag{5.2}$$

We refer to $p_\theta(z|x)$ and $p_\theta(y|z, x)$ as the *prior network* and the *decoder*, respectively. Eq. 5.2 involves an intractable marginalization over the latent variable z. The CVAE circumvents this problem by maximizing the *evidence lower bound* (ELBO) of Eq. 5.2. To approximate the true posterior distribution $p_\theta(z|y, x)$, we introduce a *recognition network* $q_\phi(z|y, x)$, where ϕ is the set of model parameters. The ELBO is then written as:

$$\begin{aligned} \mathcal{L}_{\text{CVAE}}(\theta, \phi; y, x) &= -\text{KL}(q_\phi(z|y, x) \parallel p_\theta(z|x)) \\ &\quad + \mathbb{E}_{q_\phi(z|y,x)}[\log p_\theta(y|z, x)] \qquad (5.3) \\ &\leq \log p(y|x), \qquad (5.4) \end{aligned}$$

where KL indicates the KL divergence. We extend the CVAE with a reconstruction mechanism $p_\psi(x|y)$, where ψ is the set of model parameters. During training, it forces the model to reconstruct x from y drawn from the posterior distribution. Adding the corresponding term, we obtain the following objective function:

$$\begin{aligned} \mathcal{L}(\theta,\phi, \psi; y, x) &= \mathcal{L}_{\text{CVAE}}(\theta, \phi; y, x) \\ &\quad + \lambda \mathbb{E}_{q_\phi(z|y,x)}[\log p_\psi(x|y) p_\theta(y|z, x)], \qquad (5.5) \end{aligned}$$

where λ is the weight for the reconstruction term. Because outputting highly typical next events makes the reconstruction more difficult, the reconstruction mechanism counteracts the model's tendency to do so.

5.4.2 Neural Network Architecture

We first assign distributed representations to words in x and y using the same encoder. The encoder is a bi-directional LSTM (Hochreiter and Schmidhuber,

1997) with two layers. We concatenate the representations of the first and last words to obtain $\boldsymbol{h}^x$ and $\boldsymbol{h}^y$, the representations of x and y, respectively.

We assume that $\boldsymbol{z}$ is distributed according to a multivariate Gaussian distribution with a diagonal covariance matrix. During training, the recognition network provides the posterior distribution $q_\phi(z|y,x) \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma}^2\boldsymbol{I})$:

$$\begin{bmatrix} \boldsymbol{\mu} \\ \log(\boldsymbol{\sigma}^2) \end{bmatrix} = \boldsymbol{W}_1 \begin{bmatrix} \boldsymbol{h}^y \\ \boldsymbol{h}^x \end{bmatrix} + \boldsymbol{b}_1. \tag{5.6}$$

During testing, the prior network gives the prior distribution $p_\theta(z|x) \sim \mathcal{N}(\boldsymbol{\mu'}, \boldsymbol{\sigma'}^2\boldsymbol{I})$:

$$\begin{bmatrix} \boldsymbol{\mu'} \\ \log(\boldsymbol{\sigma'}^2) \end{bmatrix} = \boldsymbol{W}_2\boldsymbol{h}^x + \boldsymbol{b}_2. \tag{5.7}$$

We employ the reparametrization trick (Kingma and Welling, 2013) to sample $\boldsymbol{z}$ from the posterior distribution so that the error signal can propagate to the earlier part of the networks.

We use a single-layer LSTM as the decoder. When the decoder predicts y_i, the i-th word of y, it receives its previous hidden state, the word embedding of y_{i-1}, the latent variable $\boldsymbol{z}$, and the context representation calculated by an attention mechanism (Bahdanau et al., 2014).

We use a single-layer LSTM as the reconstructor. When the reconstructor predicts x_j, the j-th word of x, the inputs are its previous hidden state, the word embedding of x_{j-1}, and the context representation calculated by an attention mechanism. The parameters of the reconstructor's attention mechanism are different from those used in the decoder.

As indicated by Eqs. 5.3 and 5.5, we sum up three terms to get the loss: the cross entropy loss of the decoder, the cross entropy loss of the reconstructor, and the KL divergence between the posterior and prior. Since these loss terms are differentiable with respect to the model parameters θ, ϕ and ψ, we can optimize them in an end-to-end fashion.

5.4.3 Optimization Techniques

Encoding meaningful information in $\boldsymbol{z}$ using CVAEs with an RNN decoder is known to be hard (Bowman et al., 2016). We employ two common techniques to

alleviate the issue: (1) KL cost annealing (gradually increasing the weight of the KL term) and (2) word dropout (replacing target words with unknown words with a certain probability). For KL cost annealing, we increase the weight of the KL term using the sigmoid function. For word dropout, we start with no dropout, and gradually increase the dropout rate by 0.05 every epoch until it reaches a predefined value.

5.5 Datasets

We used the following two datasets provided by Nguyen et al. (2017).

Wikihow: Wikihow[1] organizes on a large scale descriptions of how to accomplish tasks. Each task is described by sub-tasks with detailed descriptions. Nguyen et al. (2017) created an event pair dataset by extracting adjacent sub-task descriptions.

Descript: The original DESCRIPT corpus is a collection of event sequence descriptions created through crowdsourcing (Wanzare et al., 2017). Nguyen et al. (2017) built a new corpus of event pairs by extracting the contiguous two event descriptions in the DESCRIPT corpus. Descript is of higher quality but smaller than Wikihow.

5.5.1 Construction of New Test Sets

One problem with these datasets is that each current event in their test sets is tied to only one next event. As discussed by Nguyen et al. (2017), test sets for event prediction should have reflected the fact that there could be more than one valid next event.

Inspired by Zhao et al. (2017), we addressed this problem by extending the test sets through an information retrieval technique and crowdsourcing. Figure 5.2 illustrates the overall workflow. For each of the two test sets, we first randomly chose 200 target event pairs. Our goal was to add multiple next events to each of the current events. For each event pair, we focused on the current event and retrieved 20 similar current events in the training set. As a similarity measure, we

[1] `https://www.wikihow.com`

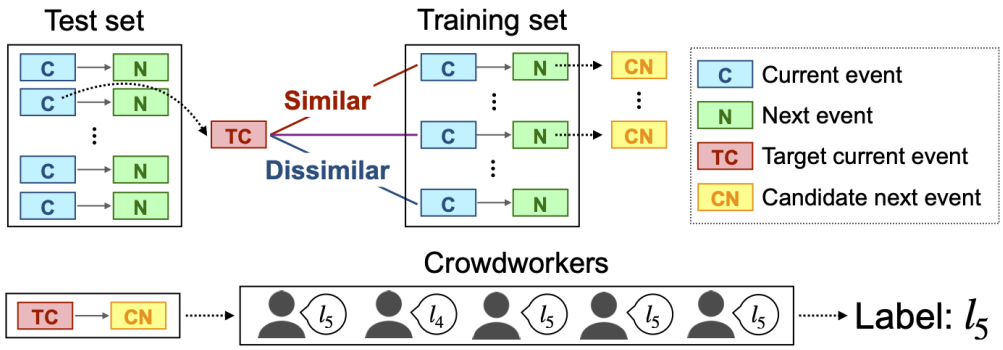

Figure 5.2: The workflow of test data construction.

used cosine similarity based on the averaged word2vec[2] embeddings of constituent words. We then used the corresponding 20 next events of the retrieved event pairs as candidates for the next events of the target current event.

We asked crowdworkers to check if a given event pair was appropriate. Note that our crowdsourcing covered not only the automatically retrieved event pairs but also the original event pairs. To remove a potential bias caused by wording, we presented a current event and a candidate next event as **A** and **B**, respectively. Each event pair was given one of the following five labels:

l_1: Strange expression.

l_2: No relation.

l_3: A and B are related, but one does not happen after the other.

l_4: A happens after B.

l_5: B happens after A.

Event pairs with label l_5 were desirable. We distributed each event pair to five workers and aggregated the five judgments by taking the majority. We used the Amazon Mechanical Turk platform and employed crowdworkers living in the US or Canada whose average work approval rates were higher than 95%. Figure 5.3

[2] `https://code.google.com/archive/p/word2vec/`

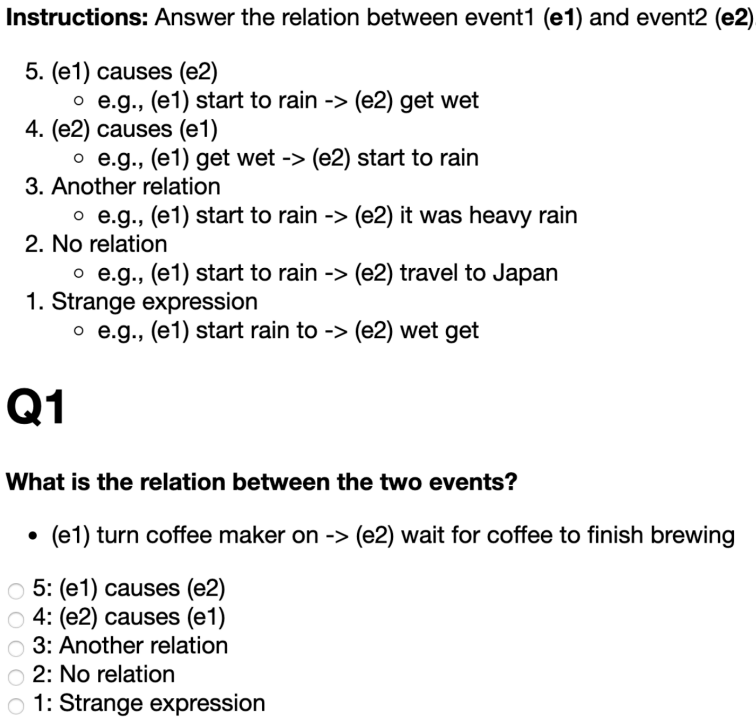

Figure 5.3: The user interface that crowdworkers used to annotate labels about the relation between events.

presents the user interface that crowdworkers used to annotate labels. The total cost was 240USD.

Table 5.1 shows the ratio of event pairs with each label. We selected event pairs with label l_5 to build new test sets. The sizes of the resultant datasets are listed in Table 5.2. One current event in Wikihow and Descript had 4.9 and 11.0 next events on average, respectively. Note that the number of unique current events in our test sets was not equal to 200 because some current events happened to have no next event with label l_5.

5.5.2 The Quality of Original Datasets

As shown in Table 5.1, only 84.1% of the original event pairs of Descript were given label l_5. Even worse, the majority of the original event pairs of Wikihow were given labels other than l_5. We had two possible explanations for this. First, be-

	l_1	l_2	l_3	l_4	l_5
Wikihow (orig.)	7.3%	20.2%	30.6%	6.5%	35.5%
Wikihow (cand.)	6.9%	37.4%	25.4%	10.0%	20.3%
Descript (orig.)	0.0%	4.5%	8.0%	3.5%	84.1%
Descript (cand.)	1.7%	19.7%	12.0%	13.3%	53.2%

Table 5.1: The result of crowdsourcing. Each number indicates the ratio of events with the corresponding label. The labels were selected by taking the majority. In no majority cases, we gave priority to the labels with smaller subscripts.

	Train	Dev	Test	New Test
Wikihow	1,287,360	26,820	26,820	858 (174)
Descript	23,320	2,915	2,915	2,203 (199)

Table 5.2: Statistics of the datasets. The training, development and test sets are the original ones provided by Nguyen et al. (2017). For each dataset, we built new test sets with multiple next events. The numbers of unique current events are in parentheses.

cause Wikihow was an open-domain dataset, it contained descriptions with which crowdworkers were not necessarily familiar (e.g., creating a website). Second, the event pairs were sometimes hard to interpret because they were extracted from adjacent descriptions out of context. The results suggest that further studies in this area should use Wikihow with caution.

5.6 Experiments

5.6.1 Model Setup

We initialized word embeddings by pre-trained word2vec embeddings. Specifically, we used the embeddings with 300 dimensions trained on the Google News corpus. The encoder, decoder, and reconstructor had hidden vectors with size

256. The prior network and the recognition network consisted of a linear map to 256-dimensional space. The latent variable z had a size of 256. We used the Adam optimizer (Kingma and Ba, 2015) for updating model parameters. The learning rate was selected from {0.0001, 0.001, 0.01}. For CVAEs, we selected the word dropout ratio from {0.0, 0.1, 0.3}. To investigate the effect of the weight parameter for the reconstruction loss, we trained and compared models with different $\lambda \in \{0.1, 0.5, 1.0\}$. Hyper-parameter tuning was done based on the *original* development sets.

5.6.2 Baselines

We compared eight seq2seq models: deterministic models (**S2S**) (Nguyen et al., 2017) and CVAE-based models (**CVAE**) with and without the attention mechanism (**A**) and the reconstruction mechanism (**R**). The hyper-parameters were the same as those reported in Section 5.6.1. The models without the attention mechanism calculated the context representation by concatenating the forward and backward last hidden states of the encoder.

To stochastically generate next events using deterministic models, we sampled words at each decoding step from the vocabulary distribution.[3] For CVAE-based models, we sampled the latent variable z and then decoded y greedily.

5.6.3 Quantitative Evaluation

Following Zhao et al. (2017), we evaluated precision and recall. For a given current event x, there were M_x reference next events r_j, $j \in [1, M_x]$. A model generated N hypothesis events h_i, $i \in [1, N]$. The precision and recall were as follows:

$$\text{precision}(x) = \frac{\sum_{i=1}^{N} \max_{j \in [1, M_x]} \text{BLEU}(r_j, h_i)}{N}$$

$$\text{recall}(x) = \frac{\sum_{j=1}^{M_x} \max_{i \in [1, N]} \text{BLEU}(r_j, h_i)}{M_x}$$

[3]We did not employ a beam search algorithm because it was not easy to compare the results with those of the probabilistic models. Beam search yields a specified number of *distinct* events while the probabilistic models can generate duplicate events.

	P@5	R@5	F@5	P@10	R@10	F@10	greedy-BLEU
S2S (Nguyen et al., 2017)	-	-	-	-	-	-	2.69±0.00
S2S+A (Nguyen et al., 2017)	-	-	-	-	-	-	2.81±0.00
S2S	2.75±0.19	**3.10±0.16**	2.91±0.17	2.69±0.12	4.22±0.16	3.28±0.14	2.62±0.23
S2S+A	2.66±0.05	**3.10±0.11**	2.86±0.08	2.74±0.08	4.15±0.11	3.30±0.06	2.64±0.07
S2S+R ($\lambda : 0.1$)	2.68±0.22	3.05±0.15	2.85±0.19	2.61±0.15	4.08±0.31	3.18±0.20	2.63±0.08
S2S+R ($\lambda : 0.5$)	2.44±0.16	2.86±0.19	2.63±0.17	2.56±0.06	4.12±0.14	3.16±0.09	2.43±0.13
S2S+R ($\lambda : 1.0$)	2.44±0.18	2.97±0.26	2.68±0.21	2.61±0.17	3.99±0.19	3.15±0.18	2.32±0.06
S2S+AR ($\lambda : 0.1$)	2.63±0.09	3.05±0.05	2.82±0.06	2.72±0.24	**4.32±0.09**	3.33±0.19	2.64±0.09
S2S+AR ($\lambda : 0.5$)	2.63±0.02	3.04±0.10	2.82±0.05	2.60±0.07	4.08±0.15	3.17±0.09	2.48±0.06
S2S+AR ($\lambda : 1.0$)	2.50±0.14	2.97±0.07	2.71±0.10	2.59±0.07	4.08±0.13	3.17±0.09	2.35±0.07
CVAE	4.94±0.11	2.07±0.08	2.92±0.10	4.92±0.08	2.09±0.07	2.93±0.08	2.62±0.03
CVAE+A	5.35±0.25	2.33±0.11	3.25±0.15	5.35±0.21	2.33±0.09	3.25±0.13	2.60±0.07
CVAE+R ($\lambda : 0.1$)	5.52±0.42	2.50±0.21	3.44±0.25	5.50±0.43	2.50±0.22	3.44±0.27	**2.79±0.11**
CVAE+R ($\lambda : 0.5$)	5.71±0.08	2.44±0.13	3.42±0.14	5.70±0.12	2.48±0.10	3.46±0.11	2.52±0.15
CVAE+R ($\lambda : 1.0$)	5.11±0.41	2.24±0.19	3.11±0.26	5.13±0.41	2.28±0.17	3.16±0.24	2.48±0.01
CVAE+AR ($\lambda : 0.1$)	**5.86±0.53**	2.40±0.10	3.40±0.02	**5.87±0.53**	2.42±0.11	3.42±0.02	2.63±0.07
CVAE+AR ($\lambda : 0.5$)	5.48±0.13	2.61±0.27	3.54±0.27	5.41±0.06	2.60±0.26	3.50±0.25	2.52±0.14
CVAE+AR ($\lambda : 1.0$)	5.32±0.28	2.86±0.28	**3.71±0.28**	5.23±0.19	3.01±0.24	**3.82±0.23**	2.48±0.04

Table 5.3: Results on Wikihow. Each model is trained three times with different random seeds. The scores are the average and standard deviation. The bold scores indicate the highest ones over models.

	P@5	R@5	F@5	P@10	R@10	F@10	greedy-BLEU
S2S (Nguyen et al., 2017)	-	-	-	-	-	-	5.42±0.00
S2S+A (Nguyen et al., 2017)	-	-	-	-	-	-	5.29±0.00
S2S	7.21±0.68	5.34±0.32	6.13±0.46	7.59±0.59	7.81±0.36	7.70±0.48	5.09±0.31
S2S+A	7.59±0.46	5.78±0.49	6.56±0.49	7.84±0.33	7.99±0.35	7.91±0.33	4.87±0.19
S2S+R ($\lambda : 0.1$)	9.04±0.42	**6.12±0.26**	7.30±0.32	8.91±0.31	**8.58±0.25**	8.74±0.28	5.49±0.22
S2S+R ($\lambda : 0.5$)	8.00±0.38	5.71±0.30	6.66±0.31	8.07±0.29	8.09±0.34	8.08±0.30	5.14±0.22
S2S+R ($\lambda : 1.0$)	6.92±0.11	5.19±0.04	5.93±0.06	6.91±0.16	7.08±0.07	6.99±0.06	4.92±0.12
S2S+AR ($\lambda : 0.1$)	8.27±0.18	5.78±0.21	6.80±0.20	8.51±0.16	8.39±0.31	8.45±0.24	5.15±0.32
S2S+AR ($\lambda : 0.5$)	8.40±0.77	6.04±0.52	7.02±0.62	8.05±0.28	7.95±0.18	8.00±0.22	**5.73±0.29**
S2S+AR ($\lambda : 1.0$)	7.58±0.49	5.58±0.23	6.43±0.31	7.35±0.20	7.51±0.27	7.43±0.23	5.34±0.16
CVAE	17.27±0.94	4.77±0.12	7.47±0.22	17.35±0.95	5.01±0.12	7.77±0.21	5.03±0.18
CVAE+A	16.13±1.91	4.51±0.20	7.04±0.42	15.99±2.21	4.75±0.33	7.32±0.61	4.65±0.33
CVAE+R ($\lambda : 0.1$)	18.19±0.69	5.40±0.24	8.33±0.36	18.44±0.33	5.89±0.17	8.92±0.22	5.50±0.24
CVAE+R ($\lambda : 0.5$)	17.33±0.61	5.10±0.42	7.87±0.48	17.35±0.57	5.67±0.40	8.55±0.47	5.34±0.09
CVAE+R ($\lambda : 1.0$)	17.20±2.05	5.03±0.26	7.78±0.52	17.10±2.41	5.42±0.33	8.23±0.63	5.24±0.11
CVAE+AR ($\lambda : 0.1$)	16.96±1.09	5.19±0.12	7.95±0.10	17.44±1.00	5.78±0.12	8.67±0.10	5.18±0.26
CVAE+AR ($\lambda : 0.5$)	**18.57±1.41**	5.45±0.36	**8.42±0.55**	**18.52±1.59**	5.91±0.34	**8.96±0.53**	5.58±0.37
CVAE+AR ($\lambda : 1.0$)	16.47±1.30	5.35±0.24	8.07±0.38	16.27±1.38	5.89±0.36	8.65±0.53	5.33±0.32

Table 5.4: Results on Descript. Each model is trained three times with different random seeds. The scores are the average and standard deviation. The bold scores indicate the highest ones over models.

where BLEU is the sentence-level variant of a well-known metric that measures the geometric mean of modified n-gram precision with the penalty of brevity (Papineni et al., 2002). The final score was averaged over the entire test set. We refer to the precision and recall as **P@N** and **R@N**, respectively. **F@N** is the harmonic mean of P@N and R@N. We report the scores with $N = 5$ and 10, in accordance with the average number of next events in our new test sets.

For comparison, we also followed the experimental procedure in Nguyen et al. (2017), where event prediction models deterministically output a single next event using greedy decoding. For CVAEs, we did this by setting z at the mean of the predicted Gaussian prior. The outputs were evaluated by BLEU. We refer to the criterion as **greedy-BLEU**. We used the original test sets for this experiment.

Table 5.3 and Table 5.4 list the evaluation results. In terms of precision (i.e., validity), the CVAE-based models consistently outperformed the deterministic models with large margins. The deterministic models achieved better recall (i.e., diversity) than the CVAE-based models, but this came with a cost of drastically low precision. The results may be somewhat surprising because our focus is on generating diverse next events. However, generating valid next events is a precondition of success, and we found that the CVAE-based models were able to satisfy the two requirements while the deterministic models were not.

For both deterministic and probabilistic models, the attention mechanism exhibited tendencies to improve precision and recall on Wikihow but to lower the scores on Descript. Our results were consistent with those of Nguyen et al. (Nguyen et al., 2017). We conjecture that Descript was so small that the attention mechanism led to overfitting.

For CVAEs, the reconstruction mechanism mostly improved recall without hurting precision, regardless of the presence or absence of the attention mechanism. Note that the best F-scores were consistently achieved by CVAEs with reconstruction. Such consistent improvements were not observed for the deterministic models. The reconstruction mechanism had evidently no effect on mitigating the difficulty of deterministic models in learning from diverse data.

In terms of greedy-BLEU, our deterministic models were competitive with the previously reported models of Nguyen et al. (Nguyen et al., 2017), though our

models were optimized based on the loss while the previous models were tuned according to greedy-BLEU. Curiously enough, greedy-BLEU indicated no big difference between the deterministic and probabilistic models, while our new test sets yielded large gaps between them in terms of precision and recall. As we will see in the next section, these differences were not spurious and did demonstrate the limitation of a single pair-based evaluation.

5.6.4 Qualitative Analysis

Table 5.5 shows next events generated by the deterministic and probabilistic models trained on Wikihow. The deterministic model generated events without any duplication, leading to a high recall. However, most of the generated events, such as "choose high speed goals", look irrelevant to the current event. This suggests that, as indicated by low precision, the deterministic model fails to generate valid next events when being forced to diversify the outputs.

The CVAE without the reconstruction mechanism appears to have generated next events that were generally valid and, at a first glance, diverse. However, a closer look reveals that they expressed a small number of highly typical events and that their semantic diversity was not large. For example, "consider the risks of psychotherapy" was semantically identical with "consider the risk factors" in this context. Compared with the vanilla CVAE, the CVAEs with reconstruction successfully generated semantically diverse next events. We would like to emphasize that the diversity was improved without sacrificing precision.

Table 5.6 shows an example from Descript. As with Wikihow, the deterministic model generated next events that were diverse but mostly invalid. The vanilla CVAE also lacked semantic diversity as with the case of Wikihow. The CVAE with reconstruction ($\lambda = 0.1$) alleviated the problem and was able to produce next events that were both valid and diverse. However, care must be taken in tuning λ, as the model with $\lambda = 1.0$ ended up concentrating on a small number of next events, which was indicated by low recall. With a too large λ, the model was strongly biased toward next events that had one-to-one correspondences with current events. Note that we could tune λ if we had new development sets with multiple next events, in addition to new test sets.

Current event: talk to mental health professional

Reference next events: [1] find support group, [2] reestablish your sense of safety, [3] spend time facing why you distrust people, [4] talk to your doctor about medication, [5] try cognitive behavioral therapy cbt, and [6] visit more than one counselor

S2S	CVAE	CVAE+AR ($\lambda : 0.1$)	CVAE+AR ($\lambda : 1.0$)
1. adjust your support system (1)	1. seek therapy (11)	1. consider the possibility of medical treatment (14)	1. get referral to therapist (8)
2. choose high speed goals (1)	2. consider psychotherapy (5)	2. ask your doctor about medications (4)	2. ask your doctor about medication (8)
3. join support group (1)	3. consider your therapist (2)	3. ask your family (2)	3. get support (4)
4. understand your parent lifestyle (1)	4. consider the risks of psychotherapy (2)	4. be aware of your depressive symptoms (2)	4. get an overview of the various topics (2)
5. listen to someone knowledgeable (1)	5. consider the risk factors (2)	5. be aware of your own mental health (2)	5. be aware of the benefits of testosterone (1)

Table 5.5: Next events generated by the deterministic and probabilistic models trained on Wikihow. We sampled 30 next events for each current event. Note that the samples can be duplicate. The numbers in parentheses indicate the frequencies.

Current event: board bus

Reference next events: [1] buy a ticket, [2] find a seat if available or stand if necessary, [3] give bus driver token or money, [4] pay driver or give prepaid card or ticket, [5] pay fare or give ticket if needed, [6] pay for the bus [7] pay the driver, [8] place your luggage overhead or beneath seat, [9] reach the destination, [10] sit down, [11] sit down and ride, [12] sit in your seat, [13] sit on the bus, and [14] take a seat in the bus

S2S	CVAE	CVAE+R ($\lambda : 0.1$)	CVAE+R ($\lambda : 1.0$)
1. pay for ticket (1)	1. get off bus (9)	1. find seat (10)	1. pay fare (29)
2. delivery driver (1)	2. pay bus fare (7)	2. pay fare (5)	2. pay the fare (1)
3. get on train (1)	3. get on bus (6)	3. get off bus (4)	3. -
4. sit down (1)	4. pay fare (4)	4. put bag in overhead compartment (2)	4. -
5. check mirrors (1)	5. pay for ticket (2)	5. wait for bus to stop (2)	5. -

Table 5.6: Next events generated by the deterministic and probabilistic models trained on Descript. We sampled 30 next events for each current event. Note that the samples can be duplicate. The numbers in parentheses indicate the frequencies.

Finally, we have to acknowledge that there is still room for improvement in the new test sets. Although we successfully collected valid and diverse next events, the data construction procedure provided no guarantee of typicality. For the reference next events of "board bus" (Table 5.6), "pay for the bus" and its variants dominate, but we are unsure if they are truly more typical than "place your luggage overhead or beneath seat." One way to take typicality into account is to ask a large number of crowdworkers to type next events given the current event, rather than to check the validity of a given event pair. Although we did not do this for the high cost and difficulty in quality control, it is worth exploring in the future.

5.7 Discussion: Using a Pre-trained Language Model

In recent years, language models pre-trained on large texts have achieved high performance on many tasks. Some of those language models are designed to be able to perform generative tasks Lewis et al. (2020); Raffel et al. (2020).

In order to investigate whether such models can effectively solve next event prediction, we fine-tune BART Lewis et al. (2020), a popular pre-trained language model that is applicable to generative tasks. BART is trained through corrupting a text with a noising function and reconstructing the original text in a seq2seq manner.

Table 5.7 shows the results. In Wikihow, BART consistently outperformed our CVAE-based method. This is probably due to the fact that Wikihow is an open-domain dataset, and BART was able to successfully transfer the knowledge gained through pre-training on open-domain text. In Descript, BART was again superior to our method in overall performance, although our method outperformed BART in precision.

Interestingly enough, on the evaluation metric of the previous study (greedy-BLEU), BART was as good as, or rather worse than, the previous study. This result demonstrates the limitation of the evaluation method which relies on a single next event, and supports the effectiveness of the proposed new dataset and evaluation method for next event prediction.

	P@5	R@5	F@5	P@10	R@10	F@10	greedy-BLEU
CVAE+AR	5.3±0.3	2.9±0.3	3.7±0.3	5.2±0.2	3.0±0.2	3.8±0.2	2.5±0.0
BART	**6.2±0.2**	**4.2±0.2**	**5.0±0.2**	**6.1±0.2**	**5.0±0.1**	**5.5±0.1**	**3.2±0.0**

(a) Results on Wikihow.

	P@5	R@5	F@5	P@10	R@10	F@10	greedy-BLEU
CVAE+AR	**18.6±1.4**	5.5±0.4	8.4±0.6	**18.5±1.6**	5.9±0.3	9.0±0.5	**5.6±0.4**
BART	16.4±0.6	**6.4±0.4**	**9.2±0.5**	16.7±0.7	**7.5±0.2**	**10.3±0.1**	4.5±0.0

(b) Results on Descript.

Table 5.7: Results of next event prediction using BART (Lewis et al., 2020). The scores of compared methods are cited from Table 5.3 and Table 5.4.

5.8 Summary of This Chapter

We tackled the task of generating next events given a current event. Aiming to capture the diversity of next events, we proposed to use a CVAE-based seq2seq model with a reconstruction mechanism. To fairly evaluate diversity-aware models, we built new test sets with multiple next events. The CVAE-based models drastically outperformed deterministic models in terms of precision and that the reconstruction mechanism improved the recall of CVAE-based models without sacrificing precision. Although we focused on event pairs in the present work, the use of longer sequence of events would be a promising direction for future work.

Chapter 6

Conclusion

6.1 Overview

In this thesis, we tackled three fundamental tasks in event-level language analysis: volitionality classification, discourse relation analysis, and next event prediction. We considered the characteristics of each task and proposed an effective method for each of them. As for volitionality classification, we devised heuristics that assigns labels to events in a raw corpus with high accuracy, and proposed a framework to learn classifiers from such heuristically labeled events effectively. As for discourse relation analysis, we proposed a self-supervised learning framework to obtain generalized event representations that are effective in recognizing discourse relations. As for next event prediction, we proposed to employ a probabilistic generative model based on CVAEs, taking into account the diversity of subsequent events.

We describe the relationship between these tasks and methods, and discuss the future prospects in the following.

6.2 The Relation between the Proposed Methods

This thesis tackled event classification, event-to-event relation analysis, and event prediction and proposed effective solutions to them. The proposed methods can be combined at an application level. To demonstrate that, we have developed a

system to analyze causality between events extracted from a huge amount of texts, called CausalityGraph (Kiyomaru et al., 2020). In CausalityGraph, discourse relation analysis is used to extract event pairs with causality. Event volitionality classification is used to categorize extracted event pairs. Event prediction has not been introduced yet, but can be used to generalize extracted event pairs and show plausible causal relations that are not explicitly written in the information source.

At the same time, we can expect a synergistic effect by combining these methods. For example, when solving discourse relation analysis, the performance could be improved by considering commonsense knowledge by learning event prediction or by explicitly considering the fundamental properties of events by learning event classification tasks, such as volitionality classification. The method of learning contextualized event representations proposed for discourse relation analysis would be effective for event classification and event prediction for events in context.

6.3 Future Prospects

6.3.1 Unified Event-level Language Analysis

It is interesting to learn event-level language analysis tasks jointly. As mentioned above, joint learning of event-level analysis tasks is expected to have a synergistic effect and improve the performance of each task. Until a few years ago, it was common to employ neural networks with different architectures for each task, which made it difficult to learn tasks with different inputs or outputs jointly. However, with the advent of general-purpose language models based on the Transformer architecture, such joint learning is becoming possible.

6.3.2 Exploratory Language Analysis by Language Modeling

In this thesis, we described that end-to-end learning is inherently inapplicable to exploratory language analysis, which motivated us to work on event-level language analysis.

However, recently, it has been shown that large language models trained on large texts can perform various tasks, even though they are not trained to solve

such tasks explicitly. The notable example is GPT-3 (Brown et al., 2020). When GPT-3 is given a context "Where was the 1994 Olympics held?," for example, it outputs the correct location "Tokyo" as the continuation. Is it possible to solve exploratory text analysis by asking GPT-3 what one wishes to know? Is event-level language analysis still necessary?

We argue that event-level language analysis is still necessary, even if models such as GPT-3 can provide plausible answers. In general, what we wish to obtain through exploratory text analysis is what we do not know and thus cannot determine whether it is correct. Unfortunately, the output of language models is not always correct. In order to verify the output, there is no choice but to look through a certain amount of information. Structural language analysis will still be useful as a technology to support the verification process.

6.3.3 Application Development

Application development using the proposed methods is important future work. Now that we have powerful tools for language understanding, it is more important than ever to think about what we need to solve using such tools. Application development is a mirror of important problems. The problems that are found through application development are those that have a clear value to be solved. While improving the performance on benchmark datasets is important, the pursuit of real-world value is essential in guiding the research in the right direction.

Appendix A

Volitionality Indicating Words

The following is a list of volitional and non-volitional adverbs that we prepared in Section 3.5. The number in parentheses indicates the number of events labeled with the adverb when we used 30M documents in CC-100 as information source.

Japanese Volitional Adverbs: *aete* (5,293), *isoide* (4,187), *jikkuri* (4,017), *shinchoni* (3,743), *nonbiri* (3,262), *wazawaza* (3,222), *sassato* (1,945), *shuchushite* (1,194), *itotekini* (920), *wazato* (880), *ishikitekini* (786), *nennirini* (766), *kirakuni* (591), *chakkari* (510), and *chuibukaku* (496).

Japanese Non-volitional Adverbs: *omowazu* (18,115), *tsui* (15,897), *jidotekini* (14,212), *futo* (12,050), *tsuitsui* (10,054), *jidode* (5,058), *kizuitara* (1,414), *ukkari* (1,333), *saiwainimo* (950), *kouunnnimo* (571), *omoigakezu* (546), *ainiku* (422), *kouunnnakotoni* (285), *fukounimo* (63), and *fukounakotoni* (32).

English Volitional Adverbs: carefully (13,594), thoroughly (12,468), actively (10,379), deliberately (3,366), intentionally (2,713), consciously (1,846), purposely (1,391), hurriedly (942), attentively (839), and proactively (388).

English Non-volitional Adverbs: unfortunately (13,070), automatically (12,824), accidentally (5,272), unexpectedly (3,106), luckily (1,894), instinctively (1,321), unconsciously (1,059), inadvertently (999), unintentionally (635), and carelessly (384).

Bibliography

[1] *ACE (Automatic Content Extraction) English Annotation Guidelines for Events*, 5.4.3 2005.07.01 edition, 2005.

[2] Shuya Abe, Kentaro Inui, and Yuji Matsumoto. Acquiring event relation knowledge by learning cooccurrence patterns and fertilizing cooccurrence samples with verbal nouns. In *Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-I*, 2008.

[3] Shuya Abe, Kentaro Inui, and Yuji Matsumoto. Two-phased event relation acquisition: Coupling the relation-oriented and argument-oriented approaches. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 1–8, Manchester, UK, August 2008. Coling 2008 Organizing Committee.

[4] Apoorv Agarwal and Owen Rambow. Automatic detection and classification of social events. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1024–1034, Cambridge, MA, October 2010. Association for Computational Linguistics.

[5] Jacqueline Aguilar, Charley Beller, Paul McNamee, Benjamin Van Durme, Stephanie Strassel, Zhiyi Song, and Joe Ellis. A comparison of the events and relations across ACE, ERE, TAC-KBP, and FrameNet annotation standards. In *Proceedings of the Second Workshop on EVENTS: Definition, Detection, Coreference, and Representation*, pages 45–53, Baltimore, Maryland, USA, June 2014. Association for Computational Linguistics.

[6] Sanjeev Arora, Yingyu Liang, and Tengyu Ma. A simple but tough-to-beat baseline for sentence embeddings. In *Proceedings of the Fifth International Conference on Learning Representations (ICLR)*, 2017.

[7] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[8] Hongxiao Bai and Hai Zhao. Deep enhanced representation for implicit discourse relation recognition. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 571–583, Santa Fe, New Mexico, USA, August 2018. Association for Computational Linguistics.

[9] Collin F. Baker, Charles J. Fillmore, and John B. Lowe. The Berkeley FrameNet project. In *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 1*, pages 86–90, Montreal, Quebec, Canada, August 1998. Association for Computational Linguistics.

[10] Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. Abstract Meaning Representation for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186, Sofia, Bulgaria, August 2013. Association for Computational Linguistics.

[11] Adam J. Berinsky, Gregory A. Huber, and Gabriel S. Lenz. Evaluating Online Labor Markets for Experimental Research: Amazon.com's Mechanical Turk. *Political Analysis*, 20:351 – 368, 2012.

[12] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[13] Tolga Bolukbasi, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, and Adam T Kalai. Man is to computer programmer as woman is to homemaker?

debiasing word embeddings. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

[14] Samuel R. Bowman and Harshit Chopra. Automatic Animacy classification. In *Proceedings of the NAACL HLT 2012 Student Research Workshop*, pages 7–10, Montréal, Canada, June 2012. Association for Computational Linguistics.

[15] Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal, September 2015. Association for Computational Linguistics.

[16] Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. Generating sentences from a continuous space. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 10–21, Berlin, Germany, August 2016. Association for Computational Linguistics.

[17] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020.

[18] Ziqiang Cao, Furu Wei, Wenjie Li, and Sujian Li. Faithful to the original: Fact aware neural abstractive summarization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[19] Lynn Carlson, Daniel Marcu, and Mary Ellen Okurowski. RST Discourse Treebank LDC2002T07. Web Download., 2002.

[20] Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Brian Strope, and Ray Kurzweil. Universal sentence encoder for English. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 169–174, Brussels, Belgium, November 2018. Association for Computational Linguistics.

[21] Nathanael Chambers and Dan Jurafsky. Unsupervised learning of narrative event chains. In *Proceedings of ACL-08: HLT*, pages 789–797, Columbus, Ohio, June 2008. Association for Computational Linguistics.

[22] Nathanael Chambers, Shan Wang, and Dan Jurafsky. Classifying temporal relations between events. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 173–176, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

[23] Jifan Chen, Qi Zhang, Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. Implicit discourse relation detection via a deep architecture with gated relevance network. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1726–1735, Berlin, Germany, August 2016. Association for Computational Linguistics.

[24] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR, 2020.

[25] Yubo Chen, Liheng Xu, Kang Liu, Daojian Zeng, and Jun Zhao. Event extraction via dynamic multi-pooling convolutional neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 167–176, Beijing, China, July 2015. Association for Computational Linguistics.

[26] Sumit Chopra, Michael Auli, and Alexander M. Rush. Abstractive sentence summarization with attentive recurrent neural networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 93–98, San Diego, California, June 2016. Association for Computational Linguistics.

[27] Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

[28] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online, July 2020. Association for Computational Linguistics.

[29] Iria da Cunha, Juan-Manuel Torres-Moreno, and Gerardo Sierra. On the development of the RST Spanish treebank. In *Proceedings of the 5th Linguistic Annotation Workshop*, pages 1–10, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.

[30] Luciano Del Corro and Rainer Gemulla. Clausie: clause-based open information extraction. In *Proceedings of the 22nd international conference on World Wide Web*, pages 355–366, 2013.

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

Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[32] Haibo Ding and Ellen Riloff. Weakly supervised induction of affective events by optimizing semantic consistency. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[33] George Doddington, Alexis Mitchell, Mark Przybocki, Lance Ramshaw, Stephanie Strassel, and Ralph Weischedel. The automatic content extraction (ACE) program – tasks, data, and evaluation. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, Lisbon, Portugal, May 2004. European Language Resources Association (ELRA).

[34] Li Du, Xiao Ding, Ting Liu, and Zhongyang Li. Modeling event background for if-then commonsense reasoning using context-aware variational autoencoder. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2682–2691, Hong Kong, China, November 2019. Association for Computational Linguistics.

[35] Xinya Du and Claire Cardie. Event extraction by answering (almost) natural questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 671–683, Online, November 2020. Association for Computational Linguistics.

[36] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, volume 37 of *Proceedings of Machine Learning Research*, pages 1180–1189, Lille, France. PMLR.

[37] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario March, and Victor Lempitsky. Domain-adversarial training of neural networks. *Journal of Machine Learning Research*, 17(59):1–35, 2016.

[38] Tianyu Gao, Xingcheng Yao, and Danqi Chen. SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[39] Daniel Gildea and Daniel Jurafsky. Automatic labeling of semantic roles. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, pages 512–520, Hong Kong, October 2000. Association for Computational Linguistics.

[40] Jesús Giménez and Lluís Màrquez. A smorgasbord of features for automatic MT evaluation. In *Proceedings of the Third Workshop on Statistical Machine Translation*, pages 195–198, Columbus, Ohio, June 2008. Association for Computational Linguistics.

[41] Andrew Gordon, Zornitsa Kozareva, and Melissa Roemmele. SemEval-2012 task 7: Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 394–398, Montréal, Canada, 7-8 June 2012. Association for Computational Linguistics.

[42] Jonathan Gordon and Benjamin Van Durme. Reporting bias and knowledge acquisition. In *Proceedings of the 2013 workshop on Automated knowledge base construction*, pages 25–30, 2013.

[43] Mark Granroth-Wilding and Stephen Clark. What happens next? event prediction using a compositional neural network model. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, pages 2727–2733. AAAI Press, 2016.

[44] Jacek Haneczok, Guillaume Jacquet, Jakub Piskorski, and Nicolas Stefanovitch. Fine-grained event classification in news-like text snippets -

shared task 2, CASE 2021. In *Proceedings of the 4th Workshop on Challenges and Applications of Automated Extraction of Socio-political Events from Text (CASE 2021)*, pages 179–192, Online, August 2021. Association for Computational Linguistics.

[45] Felix Hill, Kyunghyun Cho, and Anna Korhonen. Learning distributed representations of sentences from unlabelled data. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1367–1377, San Diego, California, June 2016. Association for Computational Linguistics.

[46] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[47] Yu Hong, Jianfeng Zhang, Bin Ma, Jianmin Yao, Guodong Zhou, and Qiaoming Zhu. Using cross-entity inference to improve event extraction. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1127–1136, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.

[48] Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. spaCy: Industrial-strength Natural Language Processing in Python, 2020.

[49] Linmei Hu, Juanzi Li, Liqiang Nie, Xiao-Li Li, and Chao Shao. What happens next? Future subevent prediction using contextual hierarchical lstm. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2017.

[50] Yichen Huang, Yizhe Zhang, Oussama Elachqar, and Yu Cheng. INSET: Sentence infilling with INter-SEntential transformer. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2502–2515, Online, July 2020. Association for Computational Linguistics.

[51] Yin Jou Huang and Sadao Kurohashi. Extractive summarization considering discourse and coreference relations based on heterogeneous graph. In

Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, pages 3046–3052, Online, April 2021. Association for Computational Linguistics.

[52] Kentaro Inui, Shuya Abe, Kazuo Hara, Hiraku Morita, Chitose Sao, Megumi Eguchi, Asuka Sumida, Koji Murakami, and Suguru Matsuyoshi. Experience mining: Building a large-scale database of personal experiences and opinions from web documents. In *Proceedings of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology-Volume 01*, pages 314–321. IEEE Computer Society, 2008.

[53] Takashi Inui, Kentaro Inui, and Yuji Matsumoto. What kinds and amounts of causal knowledge can be acquired from text by using connective markers as clues? In *International Conference on Discovery Science*, pages 180–193. Springer, 2003.

[54] Dan Iter, Kelvin Guu, Larry Lansing, and Dan Jurafsky. Pretraining with contrastive sentence objectives improves discourse performance of language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4859–4870, Online, July 2020. Association for Computational Linguistics.

[55] Bram Jans, Steven Bethard, Ivan Vulić, and Marie Francine Moens. Skip n-grams and ranking functions for predicting script events. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 336–344, Avignon, France, April 2012. Association for Computational Linguistics.

[56] Heng Ji and Ralph Grishman. Refining event extraction through cross-document inference. In *Proceedings of ACL-08: HLT*, pages 254–262, Columbus, Ohio, June 2008. Association for Computational Linguistics.

[57] Xisen Jin, Zhongyu Wei, Junyi Du, Xiangyang Xue, and Xiang Ren. Towards hierarchical importance attribution: Explaining compositional se-

mantics for neural sequence models. In *Proceedings of the Fifth International Conference on Learning Representations (ICLR)*, 2020.

[58] Shafiq Joty, Francisco Guzmán, Lluís Màrquez, and Preslav Nakov. Discourse structure in machine translation evaluation. *Computational Linguistics*, 43(4):683–722, December 2017.

[59] Nobuhiro Kaji and Masaru Kitsuregawa. Automatic construction of polarity-tagged corpus from HTML documents. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, pages 452–459, Sydney, Australia, July 2006. Association for Computational Linguistics.

[60] Daisuke Kawahara and Sadao Kurohashi. A fully-lexicalized probabilistic model for Japanese syntactic and case structure analysis. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 176–183, New York City, USA, June 2006. Association for Computational Linguistics.

[61] Daisuke Kawahara, Yuichiro Machida, Tomohide Shibata, Sadao Kurohashi, Hayato Kobayashi, and Manabu Sassano. Rapid development of a corpus with discourse annotations using two-stage crowdsourcing. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 269–278, Dublin, Ireland, August 2014. Dublin City University and Association for Computational Linguistics.

[62] Brendan Kennedy, Xisen Jin, Aida Mostafazadeh Davani, Morteza Dehghani, and Xiang Ren. Contextualizing hate speech classifiers with post-hoc explanation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5435–5442, Online, July 2020. Association for Computational Linguistics.

[63] Najoung Kim, Song Feng, Chulaka Gunasekara, and Luis Lastras. Implicit discourse relation classification: We need to talk about evaluation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5404–5414, Online, July 2020. Association for Computational Linguistics.

[64] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *Proceedings of the Third International Conference on Learning Representations (ICLR)*, 2015.

[65] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[66] Durk P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. Semi-supervised learning with deep generative models. In *Advances in Neural Information Processing Systems*, pages 3581–3589. Curran Associates, Inc., 2014.

[67] Paul Kingsbury and Martha Palmer. From TreeBank to PropBank. In *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC'02)*, Las Palmas, Canary Islands - Spain, May 2002. European Language Resources Association (ELRA).

[68] Ryan Kiros, Yukun Zhu, Russ R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Skip-thought vectors. In *Advances in Neural Information Processing Systems*, volume 28, pages 3294–3302. Curran Associates, Inc., 2015.

[69] Yudai Kishimoto, Yugo Murawaki, and Sadao Kurohashi. A knowledge-augmented neural network model for implicit discourse relation classification. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 584–595, Santa Fe, New Mexico, USA, August 2018. Association for Computational Linguistics.

[70] Yudai Kishimoto, Shinnosuke Sawada, Yugo Murawaki, Daisuke Kawahara, and Sadao Kurohashi. Improving crowdsourcing-based annotation of Japanese discourse relations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, May 2018. European Language Resources Association (ELRA).

[71] Yudai Kishimoto, Yugo Murawaki, Daisuke Kawahara, and Sadao Kurohashi. Japanese Discourse Relation Analysis: Task Definition, Connective

Detection, and Corpus Annotation. *Journal of Natural Language Processing*, 27(4):889–931, 2020. (in Japanese).

[72] Hirokazu Kiyomaru, Nobuhiro Ueda, Takashi Kodama, Yu Tanaka, Ribeka Tanaka, Daisuke Kawahara, and Sadao Kurohashi. Causalitygraph: A system to organize causes, results, and solutions of events based on structural language analysis. In *Proceedings of 26th Annual Conference of Association for Natural Language Processing*, pages 1125–1128, 2020. (in Japanese).

[73] François Lareau, Mark Dras, and Robert Dale. Detecting interesting event sequences for sports reporting. In *Proceedings of the 13th European Workshop on Natural Language Generation*, pages 200–205, Nancy, France, September 2011. Association for Computational Linguistics.

[74] Hae-Yun Lee and Sueun Jun. Constructing an ontology of coherence relations: An example of 'causal relation'. In *Proceedings of the 22nd Pacific Asia Conference on Language, Information and Computation*, pages 245–252, The University of the Philippines Visayas Cebu College, Cebu City, Philippines, November 2008. De La Salle University, Manila, Philippines.

[75] Hector J. Levesque, Ernest Davis, and Leora Morgenstern. The winograd schema challenge. In *Proceedings of the Thirteenth International Conference on Principles of Knowledge Representation and Reasoning*, KR'12, page 552–561. AAAI Press, 2012. ISBN 9781577355601.

[76] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online, July 2020. Association for Computational Linguistics.

[77] Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 Conference of the North American Chapter of the*

Association for Computational Linguistics: Human Language Technologies, pages 110–119, San Diego, California, June 2016. Association for Computational Linguistics.

[78] Junyi Jessy Li, Kapil Thadani, and Amanda Stent. The role of discourse units in near-extractive summarization. In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 137–147, Los Angeles, September 2016. Association for Computational Linguistics.

[79] Qi Li, Heng Ji, and Liang Huang. Joint event extraction via structured prediction with global features. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 73–82, Sofia, Bulgaria, August 2013. Association for Computational Linguistics.

[80] Shasha Liao and Ralph Grishman. Using document level cross-event inference to improve event extraction. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 789–797, Uppsala, Sweden, July 2010. Association for Computational Linguistics.

[81] Chao-Hong Liu, Yasufumi Moriya, Alberto Poncelas, and Declan Groves. IJCNLP-2017 task 4: Customer feedback analysis. In *Proceedings of the IJCNLP 2017, Shared Tasks*, pages 26–33, Taipei, Taiwan, December 2017. Asian Federation of Natural Language Processing.

[82] Jian Liu, Yubo Chen, Kang Liu, Wei Bi, and Xiaojiang Liu. Event extraction as machine reading comprehension. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1641–1651, Online, November 2020. Association for Computational Linguistics.

[83] Yang Liu and Mirella Lapata. Text summarization with pretrained encoders. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on*

Natural Language Processing (EMNLP-IJCNLP), pages 3730–3740, Hong Kong, China, November 2019. Association for Computational Linguistics.

[84] Yang Liu and Sujian Li. Recognizing implicit discourse relations via repeated reading: Neural networks with multi-level attention. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1224–1233, Austin, Texas, November 2016. Association for Computational Linguistics.

[85] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[86] Chi-kiu Lo, Anand Karthik Tumuluru, and Dekai Wu. Fully automatic semantic MT evaluation. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 243–252, Montréal, Canada, June 2012. Association for Computational Linguistics.

[87] Peter LoBue and Alexander Yates. Types of common-sense knowledge needed for recognizing textual entailment. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 329–334, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.

[88] Annie Louis, Aravind Joshi, and Ani Nenkova. Discourse indicators for content selection in summarization. In *Proceedings of the SIGDIAL 2010 Conference*, pages 147–156, Tokyo, Japan, September 2010. Association for Computational Linguistics.

[89] William C Mann and Sandra A Thompson. Rhetorical structure theory: Toward a functional theory of text organization. *Text-interdisciplinary Journal for the Study of Discourse*, 8(3):243–281, 1988.

[90] Thomas Meyer, Najeh Hajlaoui, and Andrei Popescu-Belis. Disambiguating discourse connectives for statistical machine translation. *IEEE/ACM*

Transactions on Audio, Speech, and Language Processing, 23(7):1184–1197, 2015.

[91] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, volume 26, pages 3111–3119. Curran Associates, Inc., 2013.

[92] Paramita Mirza, Rachele Sprugnoli, Sara Tonelli, and Manuela Speranza. Annotating causality in the TempEval-3 corpus. In *Proceedings of the EACL 2014 Workshop on Computational Approaches to Causality in Language (CAtoCL)*, pages 10–19, Gothenburg, Sweden, April 2014. Association for Computational Linguistics.

[93] Jiaqi Mu, Suma Bhat, and Pramod Viswanath. All-but-the-top: Simple and effective postprocessing for word representations. In *Proceedings of the Third International Conference on Learning Representations (ICLR)*, 2018.

[94] Martina Naughton, Nicola Stokes, and Joe Carthy. Investigating statistical techniques for sentence-level event classification. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 617–624, Manchester, UK, August 2008. Coling 2008 Organizing Committee.

[95] Dai Quoc Nguyen, Dat Quoc Nguyen, Cuong Xuan Chu, Stefan Thater, and Manfred Pinkal. Sequence to sequence learning for event prediction. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 37–42, Taipei, Taiwan, November 2017. Asian Federation of Natural Language Processing.

[96] Thien Huu Nguyen, Kyunghyun Cho, and Ralph Grishman. Joint event extraction via recurrent neural networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 300–309, San Diego, California, June 2016. Association for Computational Linguistics.

[97] Kazumasa Omura, Daisuke Kawahara, and Sadao Kurohashi. A method for building a commonsense inference dataset based on basic events. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2450–2460, Online, November 2020. Association for Computational Linguistics.

[98] Ray Oshikawa, Jing Qian, and William Yang Wang. A survey on natural language processing for fake news detection. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 6086–6093, Marseille, France, May 2020. European Language Resources Association. ISBN 979-10-95546-34-4.

[99] Simon Ostermann, Ashutosh Modi, Michael Roth, Stefan Thater, and Manfred Pinkal. MCScript: A novel dataset for assessing machine comprehension using script knowledge. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, May 2018. European Language Resources Association (ELRA).

[100] Simon Ostermann, Michael Roth, and Manfred Pinkal. MCScript2.0: A machine comprehension corpus focused on script events and participants. In *Proceedings of the Eighth Joint Conference on Lexical and Computational Semantics (*SEM 2019)*, pages 103–117, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[101] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics, 2002.

[102] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. 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 2227–2237, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[103] Karl Pichotta and Raymond Mooney. Statistical script learning with multi-argument events. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 220–229, Gothenburg, Sweden, April 2014. Association for Computational Linguistics.

[104] Karl Pichotta and Raymond J. Mooney. Using sentence-level LSTM language models for script inference. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 279–289, Berlin, Germany, August 2016. Association for Computational Linguistics.

[105] Rashmi Prasad, Nikhil Dinesh, Alan Lee, Eleni Miltsakaki, Livio Robaldo, Aravind Joshi, and Bonnie Webber. The Penn Discourse TreeBank 2.0. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco, May 2008. European Language Resources Association (ELRA).

[106] Rashmi Prasad, Bonnie Webber, and Alan Lee. Discourse annotation in the PDTB: The next generation. In *Proceedings 14th Joint ACL - ISO Workshop on Interoperable Semantic Annotation*, pages 87–97, Santa Fe, New Mexico, USA, August 2018. Association for Computational Linguistics.

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

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

[109] Lianhui Qin, Zhisong Zhang, Hai Zhao, Zhiting Hu, and Eric Xing. Adversarial connective-exploiting networks for implicit discourse relation classification. In *Proceedings of the 55th Annual Meeting of the Association*

for Computational Linguistics (Volume 1: Long Papers), pages 1006–1017, Vancouver, Canada, July 2017. Association for Computational Linguistics.

[110] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020.

[111] Alan Ramponi and Barbara Plank. Neural unsupervised domain adaptation in NLP—A survey. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6838–6855, Barcelona, Spain (Online), December 2020. International Committee on Computational Linguistics.

[112] Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China, November 2019. Association for Computational Linguistics.

[113] Drew Reisinger, Rachel Rudinger, Francis Ferraro, Craig Harman, Kyle Rawlins, and Benjamin Van Durme. Semantic proto-roles. *Transactions of the Association for Computational Linguistics*, 3:475–488, 2015.

[114] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 1278–1286, Bejing, China, 2014. PMLR.

[115] Melissa Roemmele, Cosmin Adrian Bejan, and Andrew S Gordon. Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In *2011 AAAI Spring Symposium Series*, 2011.

[116] Alexander M. Rush, Sumit Chopra, and Jason Weston. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015*

Conference on Empirical Methods in Natural Language Processing, pages 379–389, Lisbon, Portugal, September 2015. Association for Computational Linguistics.

[117] Attapol Rutherford, Vera Demberg, and Nianwen Xue. A systematic study of neural discourse models for implicit discourse relation. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 281–291, Valencia, Spain, April 2017. Association for Computational Linguistics.

[118] Jun Saito, Yugo Murawaki, and Sadao Kurohashi. Minimally supervised learning of affective events using discourse relations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5758–5765, Hong Kong, China, November 2019. Association for Computational Linguistics.

[119] Maarten Sap, Ronan Le Bras, Emily Allaway, Chandra Bhagavatula, Nicholas Lourie, Hannah Rashkin, Brendan Roof, Noah A. Smith, and Yejin Choi. Atomic: An atlas of machine commonsense for if-then reasoning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33 (01):3027–3035, 2019.

[120] Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. Social IQa: Commonsense reasoning about social interactions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4463–4473, Hong Kong, China, November 2019. Association for Computational Linguistics.

[121] Roger C Schank and Robert P Abelson. Scripts, plans, and knowledge. In *Advance Papers of the Fourth International Joint Conference on Artificial Intelligence*, volume 75, pages 151–157, 1975.

[122] Iulian V. Serban, Alessandro Sordoni, Yoshua Bengio, Aaron Courville, and Joelle Pineau. Building end-to-end dialogue systems using generative hier-

archical neural network models. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, pages 3776–3783. AAAI Press, 2016.

[123] Iulian Vlad Serban, Alessandro Sordoni, Ryan Lowe, Laurent Charlin, Joelle Pineau, Aaron Courville, and Yoshua Bengio. A hierarchical latent variable encoder-decoder model for generating dialogues. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, pages 3295–3301. AAAI Press, 2017.

[124] Darsh Shah, Tao Lei, Alessandro Moschitti, Salvatore Romeo, and Preslav Nakov. Adversarial domain adaptation for duplicate question detection. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1056–1063, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.

[125] Jian Shen, Yanru Qu, Weinan Zhang, and Yong Yu. Wasserstein distance guided representation learning for domain adaptation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), 2018.

[126] Tomohide Shibata, Shotaro Kohama, and Sadao Kurohashi. A large scale database of strongly-related events in Japanese. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 3283–3288, Reykjavik, Iceland, May 2014. European Language Resources Association (ELRA).

[127] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA, October 2013. Association for Computational Linguistics.

[128] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. In *Advances*

in Neural Information Processing Systems, pages 3483–3491. Curran Associates, Inc., 2015.

[129] Swapna Somasundaran, Galileo Namata, Janyce Wiebe, and Lise Getoor. Supervised and unsupervised methods in employing discourse relations for improving opinion polarity classification. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 170–179, Singapore, August 2009. Association for Computational Linguistics.

[130] Alessandro Sordoni, Michel Galley, Michael Auli, Chris Brockett, Yangfeng Ji, Margaret Mitchell, Jian-Yun Nie, Jianfeng Gao, and Bill Dolan. A neural network approach to context-sensitive generation of conversational responses. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 196–205, Denver, Colorado, May–June 2015. Association for Computational Linguistics.

[131] Robyn Speer, Joshua Chin, and Catherine Havasi. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, page 4444–4451. AAAI Press, 2017.

[132] Manfred Stede and Arne Neumann. Potsdam commentary corpus 2.0: Annotation for discourse research. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 925–929, Reykjavik, Iceland, May 2014. European Language Resources Association (ELRA).

[133] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*, pages 3104–3112. Curran Associates, Inc., 2014.

[134] Adam Teichert, Adam Poliak, Benjamin Van Durme, and Matthew Gormley. Semantic proto-role labeling. *Proceedings of the AAAI Conference on Artificial Intelligence*, 31(1), 2017.

[135] Zhaopeng Tu, Yang Liu, Lifeng Shang, Xiaohua Liu, and Hang Li. Neural machine translation with reconstruction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.

[136] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2019.

[137] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[138] Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. Superglue: A stickier benchmark for general-purpose language understanding systems. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[139] Lilian Wanzare, Alessandra Zarcone, Stefan Thater, and Manfred Pinkal. Inducing script structure from crowdsourced event descriptions via semi-supervised clustering. In *Proceedings of the 2nd Workshop on Linking Models of Lexical, Sentential and Discourse-level Semantics*, pages 1–11, Valencia, Spain, April 2017. Association for Computational Linguistics.

[140] Lilian D. A. Wanzare, Alessandra Zarcone, Stefan Thater, and Manfred Pinkal. A crowdsourced database of event sequence descriptions for the acquisition of high-quality script knowledge. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3494–3501, Portorož, Slovenia, May 2016. European Language Resources Association (ELRA).

[141] Noah Weber, Leena Shekhar, Niranjan Balasubramanian, and Nathanael Chambers. Hierarchical quantized representations for script generation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3783–3792, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.

[142] Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzmán, Armand Joulin, and Edouard Grave. CCNet: Extracting high quality monolingual datasets from web crawl data. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4003–4012, Marseille, France, May 2020. European Language Resources Association. ISBN 979-10-95546-34-4.

[143] Aaron Steven White, Drew Reisinger, Keisuke Sakaguchi, Tim Vieira, Sheng Zhang, Rachel Rudinger, Kyle Rawlins, and Benjamin Van Durme. Universal decompositional semantics on Universal Dependencies. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1713–1723, Austin, Texas, November 2016. Association for Computational Linguistics.

[144] John Wieting and Kevin Gimpel. ParaNMT-50M: Pushing the limits of paraphrastic sentence embeddings with millions of machine translations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 451–462, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[145] John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. Towards universal paraphrastic sentence embeddings. In *Proceedings of the Third International Conference on Learning Representations (ICLR)*, 2016.

[146] Adina Williams, Nikita Nangia, and Samuel Bowman. A broad-coverage challenge corpus for sentence understanding through inference. 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 1112–1122, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[147] Jiacheng Xu, Zhe Gan, Yu Cheng, and Jingjing Liu. Discourse-aware neural extractive text summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5021–5031, Online, July 2020. Association for Computational Linguistics.

[148] Yuanmeng Yan, Rumei Li, Sirui Wang, Fuzheng Zhang, Wei Wu, and Weiran Xu. ConSERT: A contrastive framework for self-supervised sentence representation transfer. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5065–5075, Online, August 2021. Association for Computational Linguistics.

[149] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. XLNet: Generalized Autoregressive Pretraining for Language Understanding. In *Advances in Neural Information Processing Systems*, volume 32, pages 5753–5763. Curran Associates, Inc., 2019.

[150] Deniz Zeyrek, Amália Mendes, and Murathan Kurfalı. Multilingual extension of PDTB-style annotation: The case of TED multilingual discourse bank. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, May 2018. European Language Resources Association (ELRA).

[151] Biao Zhang, Deyi Xiong, jinsong su, Hong Duan, and Min Zhang. Variational neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 521–530, Austin, Texas, 2016. Association for Computational Linguistics.

[152] Xingxing Zhang, Furu Wei, and Ming Zhou. HIBERT: Document level pre-training of hierarchical bidirectional transformers for document summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5059–5069, Florence, Italy, July 2019. Association for Computational Linguistics.

[153] Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D. Manning. Position-aware attention and supervised data improve slot filling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP 2017)*, pages 35–45, 2017.

[154] Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. Men also like shopping: Reducing gender bias amplification using corpus-level constraints. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2979–2989, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

[155] Jieyu Zhao, Tianlu Wang, Mark Yatskar, Ryan Cotterell, Vicente Ordonez, and Kai-Wei Chang. Gender bias in contextualized word embeddings. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 629–634, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[156] Tiancheng Zhao, Ran Zhao, and Maxine Eskenazi. Learning discourse-level diversity for neural dialog models using conditional variational autoencoders. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 654–664, Vancouver, Canada, July 2017. Association for Computational Linguistics.

[157] Yuping Zhou and Nianwen Xue. PDTB-style discourse annotation of Chinese text. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 69–77, Jeju Island, Korea, July 2012. Association for Computational Linguistics.

[158] Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *The IEEE International Conference on Computer Vision (ICCV)*, 2015.

[159] Yuan Zhuang, Tianyu Jiang, and Ellen Riloff. Affective event classification with discourse-enhanced self-training. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5608–5617, Online, November 2020. Association for Computational Linguistics.

List of Major Publications

[1] Hirokazu Kiyomaru and Sadao Kurohashi. Minimally-Supervised Joint Learning of Event Volitionality and Subject Animacy Classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022.

[2] Hirokazu Kiyomaru and Sadao Kurohashi. Contextualized and Generalized Sentence Representations by Contrastive Self-Supervised Learning: A Case Study on Discourse Relation Analysis. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5578–5584, 2021.

[3] Hirokazu Kiyomaru, Kazumasa Omura, Yugo Murawaki, Daisuke Kawahara, and Sadao Kurohashi. Diversity-aware Event Prediction based on a Conditional Variational Autoencoder with Reconstruction In *Proceedings of the First Workshop on Commonsense Inference in Natural Language Processing*, pages 113–122, 2019.

List of Other Publications

[1] Akiko Aizawa, Frederic Bergeron, Junjie Chen, Fei Cheng, Katsuhiko Hayashi, Kentaro Inui, Hiroyoshi Ito, Daisuke Kawahara, Masaru Kitsuregawa, Hirokazu Kiyomaru, Masaki Kobayashi, Takashi Kodama, Sadao Kurohashi, Qianying Liu, Masaki Matsubara, Yusuke Miyao, Atsuyuki Morishima, Yugo Murawaki, Kazumasa Omura, Haiyue Song, Eiichiro Sumita, Shinji Suzuki, Ribeka Tanaka, Yu Tanaka, Masashi Toyoda, Nobuhiro Ueda, Honai Ueoka, Masao Utiyama, and Ying Zhong. A System for Worldwide COVID-19 Information Aggregation. In *Proceedings of the 1st Workshop on NLP for COVID-19 (Part 2) at EMNLP 2020*, 2020.

[2] Hirokazu Kiyomaru, Nobuhiro Ueda, Takashi Kodama, Yu Tanaka, Ribeka Tanaka, Daisuke Kawahara, and Sadao Kurohashi. CausalityGraph: A System to Organize Causes, Results, and Solutions of Events based on Structural

Language Analysis. In *Proceedings of 26th Annual Conference of Association for Natural Language Processing*, pages 1125–1128, 2020. (in Japanese).

[3] Hirokazu Kiyomaru, Kazumasa Omura, Yugo Murawaki, Daisuke Kawahara, and Sadao Kurohashi. Diversity-aware Event Prediction based on a Conditional Variational Autoencoder In *Proceedings of 25th Annual Conference of Association for Natural Language Processing*, pages 1531–1534, 2019. (in Japanese).

[4] Katsuyoshi Yamagami, Hirokazu Kiyomaru, and Sadao Kurohashi. Knowledge-based Dialog Approach for Exploring User's Intention. In *Proceedings of FAIM/ISCA Workshop on Artificial Intelligence for Multimodal Human Robot Interaction*, pages 53–56, 2018.

[5] Naoki Otani, Hirokazu Kiyomaru, Daisuke Kawahara, and Sadao Kurohashi. Cross-lingual Knowledge Projection Using Machine Translation and Target-side Knowledge Base Completion. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1508–1520, 2018.

[6] Katsuyoshi Yamagami, Mitsuru Endoh, Hirokazu Kiyomaru, and Sadao Kurohashi. Associative Dialog System for Recipe Recommendation Using Food Knowledge. In *Proceedings of The 32nd Annual Conference of the Japanese Society for Artificial Intelligence*, 2018. (in Japanese).

[7] Hirokazu Kiyomaru, Katsuyoshi Yamagami, and Sadao Kurohashi. Building a Basic Culinary Knowledge Base Based on Recipe Corpus and Crowdsourcing. In *Proceedings of 24th Annual Conference of Association for Natural Language Processing*, pages 662–665, 2018. (in Japanese).

[8] Kiyona Oto, Takuma Seno, Hirokazu Kiyomaru, Kunimasa Kawasaki, Masahiko Osawa, Shigemi Nagata, and Michita Imai. The Relation between Predictive Cognition and Silence Time: the language game with a robot which can't use natural language. In *Proceedings of the Human-Agent Interaction Symposium 2017*, 2017. (in Japanese).

[9] Hirokazu Kiyomaru, Masahiko Osawa, and Michita Imai. Predictive Recognition-based Linguistic Communication Without Using Natural Language. In *Proceedings of the 6th Meeting of Special Interest Group on Artificial General Intelligence*, 2017. (in Japanese).

[10] Hirokazu Kiyomaru, Masahiko Osawa, and Hiroshi Yamakawa. BiCAmon: Activity monitoring tool on 3D connecome structures for various cognitive architectures. In *Proceedings of Neuroinformatics 2016*, pages 49–53, 2016.