NAIST-IS-DD0561002

Doctoral Dissertation

Event Relation Acquisition from Large Text Corpora

Shuya Abe

February 20, 2010

Department of Information Processing
Graduate School of Information Science
Nara Institute of Science and Technology

A Doctoral Dissertation
submitted to Graduate School of Information Science,
Nara Institute of Science and Technology
in partial fulfillment of the requirements for the degree of
Doctor of ENGINEERING

Shuya Abe

Thesis Committee:

Professor Yuji Matsumoto	(Supervisor)
Professor Kiyohiro Shikano	(Co-supervisor)
Associate Professor Kentaro Inui	(Co-supervisor)

Event Relation Acquisition from Large Text Corpora*

Shuya Abe

Abstract

The growing interest in practical NLP applications such as question answering, information extraction and multi-document summarization places increasing demands on the processing of relations between textual fragments such as entailment and causal relations. Such applications often need to rely on a large amount of lexical semantic knowledge. For example, a causal (and entailment) relation holds between the verb phrases *wash something* and *something is clean*, which reflects the commonsense notion that if someone has washed something, this object is clean as a result of the washing event. A crucial issue is how to obtain and maintain a potentially huge collection of such event relation instances. This thesis addresses the issue of how to automatically acquire such instances of relations between events from a large-scale text collection.

Addressing the task, we propose two approaches. First, We propose *Extended Espresso* that is several extensions to a state-of-the-art method originally designed for entity relation extraction, reporting on the present results of our experiments on a Japanese Web corpus. The results show that (a) there are indeed specific cooccurrence patterns useful for event relation acquisition, (b) the use of cooccurrence samples involving verbal nouns has positive impacts on both recall and precision, and (c) over five thousand relation instances are acquired from a 500M-sentence Web corpus with a precision of about 66% for *action-effect* relations.

Second, we argue the complementarity between the pattern-based relation-oriented approach and the anchor-based argument-oriented approach. We then propose a *two phase approach*, which first uses lexico-syntactic patterns to acquire predicate pairs and then uses two types of anchors to identify shared arguments. The present results of our empirical evaluation on a large-scale Japanese Web corpus have shown that (a) the

*Doctoral Dissertation, Department of Information Processing, Graduate School of Information Science, Nara Institute of Science and Technology, NAIST-IS-DD0561002, February 20, 2010.

anchor-based filtering extensively improves the accuracy of predicate pair acquisition, (b) the two types of anchors are almost equally contributive and combining them improves recall without losing accuracy, and (c) the anchor-based method also achieves high accuracy in shared argument identification.

Keywords:

computational linguistics, knowledge acquisition, event knowledge, relation knowledge, large text

大規模テキストからの事態間関係知識の獲得*

阿部 修也

内容梗概

テキスト中の含意関係や因果関係を理解することが，質問応答，情報抽出，複数文章要約などの自然言語処理の応用に役立つと知られている．これを実現するためには，例えば，動詞「洗う」と動詞句「きれいになる」が，何かを洗うという行為の結果としてその何かがきれいになるという因果関係である，といったような知識が必要である．本論文では，事態と事態の間にある関係を大規模にかつ機械的に獲得するために Extended Espresso と Two-phase method を提案し，この手法を用いた実験結果を示す．

Extended Espresso は，大規模コーパスから事態表現間の意味的関係の知識の獲得を目的として，実体間関係獲得手法として提案された Espresso を事態間関係に適用できるように拡張した手法である．この拡張は主に 2 つの点からなり，(1) 知識獲得のために事態表現を定義し，(2) 事態間関係に適合するように共起パターンのテンプレートを拡張した．日本語 Web コーパスを用いて実験したところ，(a) 事態間関係獲得に有用な共起パターンが多数存在し，パターンの学習が有効であることがわかった．また行為－効果関係については 5 億文 Web コーパスから少なくとも 5000 種類の事態対を約 66% の精度で獲得することができた．

Two-phase method は，行為－効果関係，行為－手段関係のような事態間の関係を大規模コーパスから自動的に獲得するための手法であり，共起パターンを利用する手法で起こる事態を表現する述語間で共有される項を認識することが難しいという問題に対処するために，述語間で共有される名詞（アンカー）を用いて共有項を獲得し，共起パターンを用いて獲得した所与の関係を満たす述語対と共有項を組み合わせ，共有項と共に事態間関係を獲得する．このとき 2 種類の異なるアンカーを用いることで，精度を保ったまま再現率を向上できることを確認した．

*奈良先端科学技術大学院大学 情報科学研究科 情報処理学専攻 博士論文, NAIST-IS-DD0561002, 2010 年 2 月 20 日.

キーワード

自然言語処理，知識獲得，事態知識，関係知識，大規模テキスト

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1. Background

In the early days, computers there were used mainly in government, military and industry. In recent years, computers were used also for personal situations. The environment around computers changed, and role of the computers changed with it. Because of the change, computers need to adapt to personal situations. For example, the “manner mode” of a mobile phone is a function to adapt to the needs of its user for silent operation in certain social situations. To adapt to personal situations, computers need to understand a person’s needs. To understand a person is to have common sense. Some projects tried to build common sense knowledge base (KB) for computers[13, 26, 16, 18, 25]. However, those KBs were not enough for personal adaptation of computers because of they do not cover all common sense.

The growing interest in practical Natural Language Processing (NLP) applications such as Question Answering, Information Extraction and Multi-Document Summarization has greatly increased for identification of relations between textual fragments such as entailment and causal relations. Such applications often need to rely on a large amount of lexical semantic knowledge. For example, a causal (and entailment) relation holds between the verb phrases *wash something* and *something is clean*, which reflects the commonsense notion that if someone has washed something, this object is clean as a result of the washing event.

$$\textit{wash something} \longrightarrow_{\textit{causal relation}} \textit{something is clean}$$

2. Event Relation Acquisition

Some applications need to rely on a large amount of event relation instances. We can easily estimate a number of event relation instances at following.

$$\textit{(Number of event relation instances)} =$$
$$\textit{(Number of event instances)}^2 \times \textit{(Number of relation types)}$$

The *Number of event instances* is a large amount, and the *Number of event relation instances* is consequently also a large amount. For example, *Number of event instances* include *wash something*, *something is clean*, and so on. *Number of relation types* include *causal relation*, *entailment relation* and so on.

On the other hand, the *Number of relation types* is probably not a large amount. However, different applications demand different relation types, and thus, the *Number of relation types* is not known. Event relation instances have to be built with relation types specific to each different application. To build event relation instances for various applications, we develop a method that is capable of building event relation instances from various relations.

Some applications need to rely on a large amount of event instances. However, it is high cost to build event relation instances by hand. Motivated by this problem, several research groups have reported on experiment on automatic acquisition of causal, temporal and entailment relations between event mentions (typically verbs or verb phrases). A crucial issue is how to obtain and maintain a potentially huge collection of event relation instances. This thesis addresses the problem of how to automatically acquire such instances of relations between events from a large-scale text collection, and to acquire various relation types between events. We avoid methods capable of acquiring only a specific relation types of event relation instances.

An important aspect to consider in event relation acquisition is that each event has arguments. For example, the causal relation between *wash something* and *something is clean* can be represented naturally as:

$$\textit{wash}(\text{obj}:X) \rightarrow_{cause} \textit{is\_clean}(\text{subj}:X)$$

where X is a logical variable denoting that the filler of the object slot of the *wash* event should be shared (i.e. identical) with the filler of the subject slot of the *is_clean* event.

To be more general, an instance of a given relation R can be represented as:

$$predicate_1(\mathrm{arg}_1{:}X) \rightarrow_R predicate_2(\mathrm{arg}_2{:}X)$$

where $predicate_i$ is a natural language predicate, typically a verb or adjective, and X is a logical variable denoting which argument of one predicate and which argument of the other are shared.

In other words, our purpose is to acquire event relation instances. An event relation instance satisfies following three features.

- A predicate pair
- A shared argument between a predicate pair
- A relation type between a predicate pair

3. Proposed Methods

The goal we pursue in this thesis is to acquire event relation instances. Existing methods for event relation acquisition can be classified into two approaches, which we refer to as the *pattern-based approach* and *anchor-based approach*.

We propose the following two methods.

- Extended Espresso
- Two-phase method

In this section, we outline these two methods.

3.1 Extended Espresso

Several research groups have reported their experiments on automatic acquisition of causal, temporal and entailment relations between event mentions (typically verbs or verb phrases) [14, 9, 4, 28, 21, 29, 3]. The common idea behind them is to use a small number of manually selected generic lexico-syntactic co-occurrence patterns (LSPs or simply patterns). *to Verb-X and then Verb-Y*, for example, is used to obtain temporal relations such as *marry* and *divorce* [4]. The use of such generic patterns, however, tends to have high recall and low precision, which requires an additional component for pruning extracted relations. Approaches to pruning bad relations can be broadly classified into two groups: either by devising heuristic scores [4, 28, 29] or by training heavily-supervised classifiers for disambiguation [9].

We explore a third way for enhancing present LSP-based methods for event relation acquisition. The basic idea is inspired by the following recent findings in relation extraction [23, 20], which aims at extracting semantic relations between *entities* (as opposed to *events*) from texts.

- (a) The use of generic patterns tends to be high recall but low precision, which requires an additional component for pruning.
- (b) On the other hand, there are specific patterns that are highly reliable but they are much less frequent than generic patterns and each makes only a small contribution to recall.
- (c) Combining a few generic patters with a much larger collection of reliable specific patterns boosts both precision and recall. Such specific patterns can be acquired from a very large corpus with seeds.

Given these insights, an interesting question is whether the same story applies to event relation acquisition as well or not. In this thesis, we explore this problem through the following steps. First, while previous methods use only verb-verb co-occurrences, we use co-occurrences between verbal nouns and verbs such as *cannot ⟨find out (something)⟩ due to the lack of ⟨investigation⟩* as well as verb-verb co-occurrences. This extension dramatically enlarge the pool of potential candidate LSPs. Second, we extend Pantel and Pennacchiotti [20]'s Espresso algorithm, which induces specific reliable LSPs in a bootstrapping manner for entity relation extraction, so that the extended algorithm can apply to event relations. Third, we report on the present results of our empirical experiments, where the extended algorithm is applied to a 500M-sentence Japanese Web corpus to acquire event relations.

3.2 Two-phrase Method

The method attends to consider in event relation acquisition is that each event has arguments. For example, the causal relation between *wash something* and *something is clean* can be represented naturally as:

$$wash(\text{obj}:X) \rightarrow_{cause} is\_clean(\text{subj}:X)$$

where X is a logical variable denoting that the filler of the object slot of the *wash* event should be shared (i.e. identical) with the filler of the subject slot of the *is_clean* event.

The goal we pursue in this method is therefore not only (a) to find predicate pairs that are of a given relation type, but also (b) to identify the arguments shared between the predicates if any. We call the former subtask *predicate pair acquisition* and the latter *shared argument identification*. However, existing state-of-the-art methods for event relation acquisition are designed to achieve only either of these two subtasks but *not both*. In this thesis, we propose a two-phased method, which first uses lexico-syntactic patterns to acquire predicate pairs for a given relation type and then uses two kinds of anchors to identify shared arguments.

Existing methods for event relation acquisition can be classified into two approaches, which we call the *pattern-based approach* and *anchor-based approach* in this thesis.

The common idea behind the *pattern-based approach* is to use a small number of manually selected generic lexico-syntactic co-occurrence patterns (LSPs or simply patterns). Perhaps the simplest way of using LSPs for event relation acquisition can be seen in the method Chklovski and Pantel [4] employ to develop their knowledge resource called *VerbOcean*. Their method uses a small number of manually selected generic LSPs such as *to* ⟨Verb-X⟩ *and then* ⟨Verb-Y⟩ [1] to obtain six types of semantic relations including *strength* (e.g. *taint – poison*) and *happens-before* (e.g. *marry – divorce*). The use of such generic patterns, however, tends to be high recall but low precision. Chklovski and Pantel [4], for example, report that their method obtains about 29,000 verb pairs with 65.5% precision.

The *anchor-based approach*, on the other hand, has emerged mainly in the context of paraphrase and entailment acquisition. This approach uses information of argument fillers (i.e. anchors) of each event expression as a useful clue for identifying event relations.

It is by now clear that the above two approaches, which apparently have emerged somewhat independently, could play a complementary role with each other. Pattern-based methods, on the one hand, are designed to be capable of discriminating relatively

[1] A ⟨⟩ included in an LSP denotes, throughout this thesis, a variable slot to be filled with an event expression. The filler of ⟨⟩ denotes either the lexical or syntactic constraints on the slot or an example that is to fill the slot.

fine-grained relation types. However, these methods are severely limited for the purpose of shared argument identification because lexico-syntactic patterns are not a good indication of argument-shared structure in general. The anchor-based approach, on the other hand, works well for identifying shared arguments simply because it relies on argument information in identifying synonymous or entailment verb pairs. However, it has no direct means to discriminate more fine-grained specific relations such as causality and backward presupposition. To sum up, the pattern-based approach tends to be rather *relation-oriented* while the anchor-based approach tends to be *argument-oriented*.

The complementarity between the pattern-based relation-oriented approach and the anchor-based argument-oriented approach as discussed above naturally leads us to consider combining them.

Chapter 4 introduces detail of this method and empirical evaluation.

4. Contributions

We have two main contributions.

Contribution 1 Several research groups have reported automatic event relation acquisition methods. The methods use manually selected generic lexico-syntactic co-occurrence patterns. However, the methods tend to be high recall but low precision. On the other hand, another research group has reported an automatic entity relation acquisition method. The method automatically acquires generic lexico-syntactic cooccurrence patterns, and also automatically acquire specific lexico-syntactic co-occurrence patterns. The method boosts precision and recall. However, the method is for entity relation acquisition, a similar method for event relation acquisition does not exist as well. We propose a method to automatically acquire generic lexico-syntactic cooccurrence patterns, and also automatically acquire specific lexico-syntactic cooccurrence patterns for event relation acquisition. In Chapter 3, we propose *Extended Espresso*. It is our first main contribution.

Contribution 2 The goal we pursue in this thesis is therefore not only (a) to find predicate pairs that are of a given relation type, but also (b) to identify the arguments shared between the predicates if any. We call the former subtask *predicate pair acquisition* and the latter *shared argument identification*. However, existing state-of-the-art methods for event relation acquisition are designed to achieve only either of these two subtasks but *not both*. We propose *Two-phase method*, which first uses lexico-syntactic patterns to acquire predicate pairs for a given relation type and then uses two kinds of anchors to identify shared arguments. Our second main contribution is to propose the method. We describe the method in Chapter 4.

5. Organization of This Dissertation

The organization of this thesis is as follows. Chapter 2 shows related work. In Chapter 3, we describe a method of event relation acquisition. We call the method Extended Espresso that is based on Pantel and Pennacchiotti [20]'s Espresso algorithm (Section 3 of Chapter 2). In Chapter 4, we describe a revised event relation acquisition method. The method solves problems with Extended Espresso. Chapter 5 summarizes our research and describe future work.

Chapter 2

Related Work

1. Pattern-based Approach

Several research groups have reported their experiments on automatic acquisition of causal, temporal and entailment relations between event mentions (typically verbs or verb phrases) [14, 9, 4, 28, 22, 29, 2, 3, 23, 19].

The common idea behind the *pattern-based approach* is to use a small number of manually selected generic lexico-syntactic co-occurrence patterns (LSPs or simply patterns). Perhaps the simplest way of using LSPs for event relation acquisition can be seen in the method Chklovski and Pantel [4] employ to develop their knowledge resource called *VerbOcean*. Their method uses a small number of manually selected generic LSPs such as *to* ⟨Verb-X⟩ *and then* ⟨Verb-Y⟩ to obtain six types of semantic relations including *strength* (e.g. *taint – poison*) and *happens-before* (e.g. *marry – divorce*). The use of such generic patterns, however, tends to be high recall but low precision. Chklovski and Pantel [4], for example, report that their method obtains about 29,000 verb pairs with 65.5% precision.

This low-precision problem requires an additional component for pruning extracted relations. This issue has been addressed from a variety of angles. For example, some devise heuristic statistical scores and report their impact on precision [4, 28, 29]. Another way is to incorporate a classifier trained with supervision. Inui et al. [9], for example, use a Japanese generic causal connective marker *tame* (because) and a supervised classifier learner to separately obtain four types of causal relations: *cause*, *precondition*, *effect* and *means*.

More recently, Abe et al. [2] propose to extend Pantel and Pennacchiotti [20]'s Espresso algorithm, which induces specific reliable LSPs in a bootstrapping manner for entity-entity relation extraction, so that the extended algorithm can apply to event relations. Their method learns a large number of relatively specific patterns such as *cannot* ⟨find out (something)⟩ *due to the lack of* ⟨investigation⟩ in a boot-strapping fashion, which produces a remarkable improvement on precision.

On the other hand, several research groups aim at extracting semantic relations between *entities* (as opposed to *events*) from texts[7, 24, 17, 20, 23, 5].

2. Anchor-based Approach

The *anchor-based approach*, on the other hand, has emerged mainly in the context of paraphrase and entailment acquisition. This approach uses information of argument fillers (i.e. anchors) of each event expression as a useful clue for identifying event relations. A popular way of using such argument information relies on the distributional hypothesis [6] and identifies synonymous event expressions by seeking a set of event expressions whose argument fillers have a similar distribution. Such algorithms as DIRT [14] and TE/ASE [27] represent this line of research. For example, Figure 2.1 shows an example of DIRT.

Another way of using argument information is proposed by Pekar [22], which identifies candidate verb pairs for the entailment relation by imposing criteria.

- (a) The two verbs must appear in the same local discourse-related context.
- (b) Their arguments need to refer to the same participant i.e. anchor.

For example, if a pair of clauses *Mary bought a house.* and *The house belongs to Mary.* appear in a single local discourse-related context, two pairs of verbs, *buy*(obj:*X*) – *belong*(subj:*X*) and *buy*(subj:*X*) – *belong*(to:*X*) are identified as candidate entailment pairs. Figure 2.2 shows this example.

It is by now clear that the above two approaches, which apparently have emerged somewhat independently, could play a complementary role with each other. Pattern-based methods, on the one hand, are designed to be capable of discriminating relatively fine-grained relation types. For example, the patterns used by Chklovski and Pantel [4] identify six relation types, while Abe et al. [2] identify two of the four causal relation

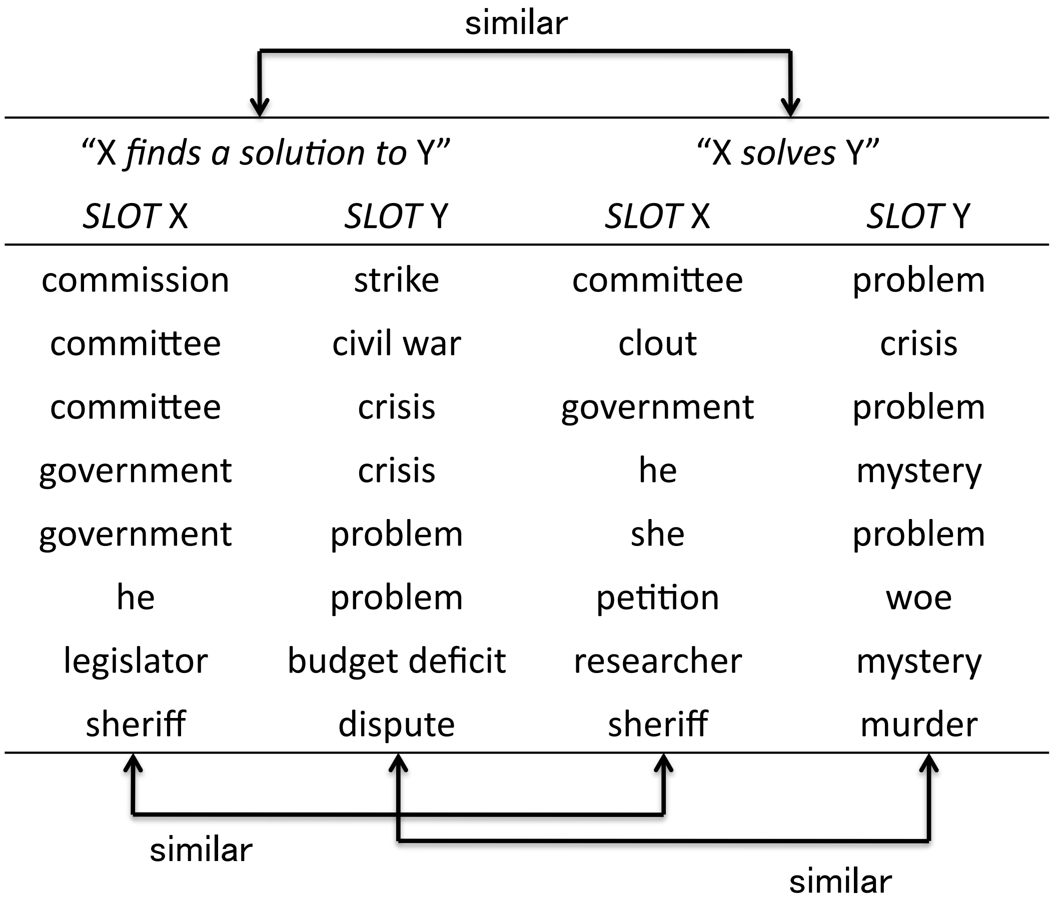

Figure 2.1. DIRT

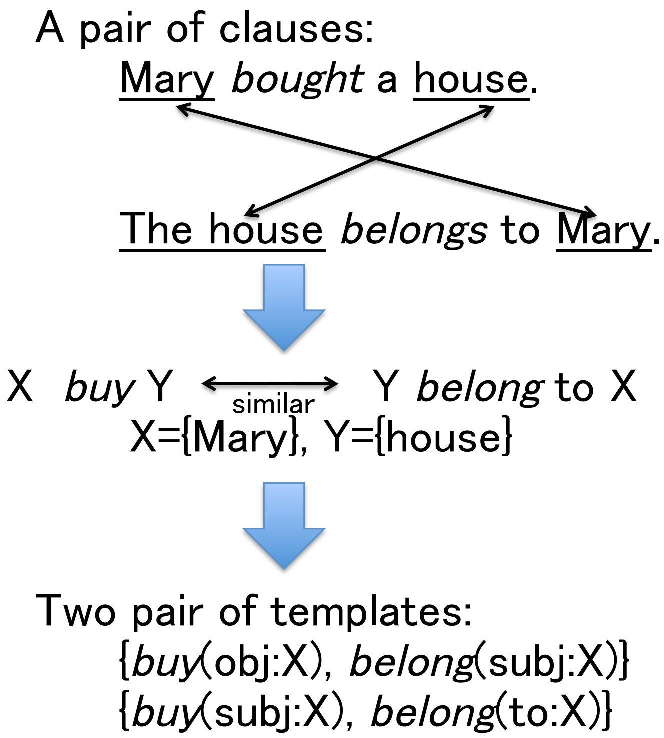

Figure 2.2. Pekar's method

types defined by Inui et al. [9]. However, these methods are severely limited for the purpose of shared argument identification because lexico-syntactic patterns are not a good indication of argument-shared structure in general.

In spite of this complementarity, however, to our best knowledge, the issue of how to benefit from both approaches has never been paid enough attention. An interesting exception could be found in Torisawa [28]'s method of combining verb pairs extracted with a highly generic connective pattern ⟨*Verb-X*⟩ *and* ⟨*Verb-Y*⟩ together with the co-occurrence statistics between verbs and their arguments. While the reported results for inference rules with temporal ordering look promising, it is not clear yet, however, whether the method applies to other types of relations because it relies on relation-specific heuristics.

3. Espresso

3.1 Introduction

In chapijcnlp, we extend Pantel and Pennacchiotti [20]'s Espresso algorithm, which induces specific reliable LSPs in a bootstrapping manner for entity relation extraction, so that the extended algorithm can apply to event relations. This section overviews Espresso algorithm. Espresso takes as input a small number of seed instances of a given target relation and iteratively learns co-occurrence patterns and relation instances in a bootstrapping manner. Figure 2.3 illustrates Espresso algorithm.

3.2 Ranking Co-occurrence Patterns

For each given relation instance $\{x, y\}$, Espresso retrieves the sentences including both x and y from a corpus and extracts from them co-occurrence samples. For example, given an instance of the *is-a* relation such as ⟨*Italy,country*⟩, Espresso may find co-occurrence samples such as *countries such as Italy* and extract such a pattern as *Y such as X*. Espresso defines the reliability $r_\pi(p)$ of pattern p as the average strength of its association with each relation instance i in the current instance set I, where each instance i is weighted by its reliability $r_\iota(i)$:

$$r_\pi(p) = \frac{1}{|I|} \sum_{i \in I} \frac{pmi(i, p)}{max_{pmi}} \times r_\iota(i) \tag{2.1}$$

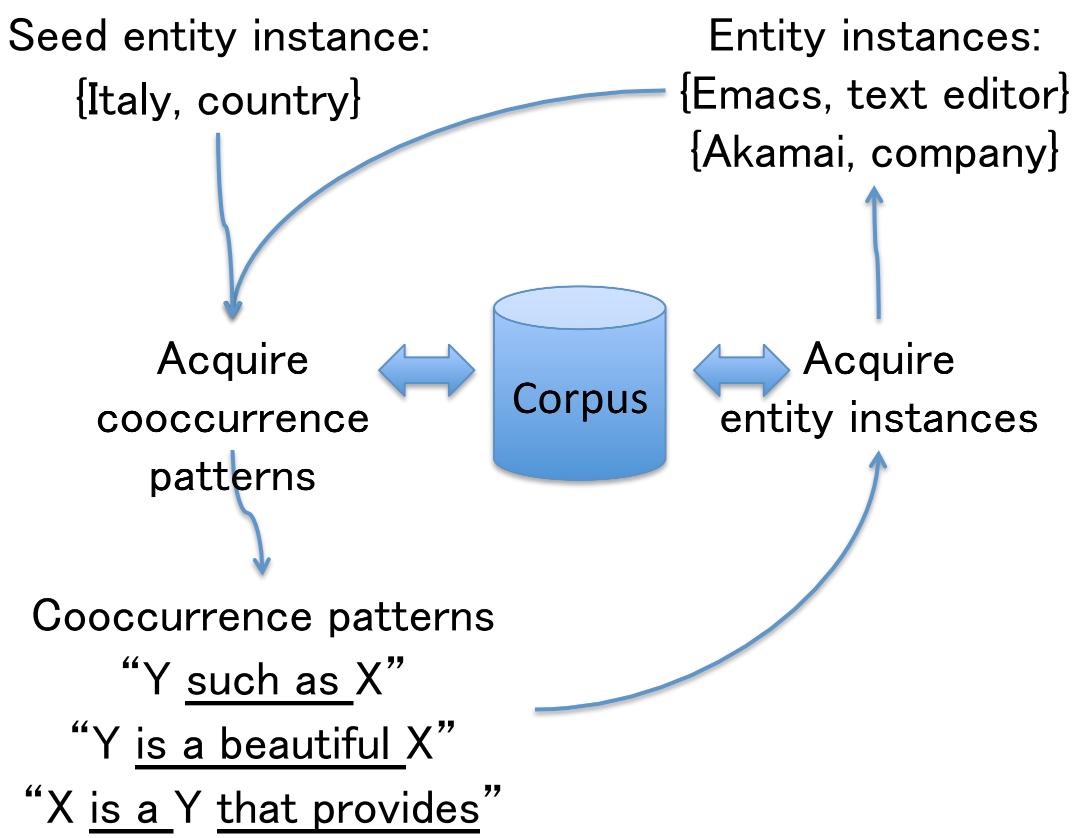

Figure 2.3. Espresso

where $pmi(i, p)$ is the pointwise mutual information between i and p, and max_{pmi} is the maximum PMI between all patterns and all instances.

$$pmi(x, y) = \log \frac{P(x, y)}{P(x)P(y)} \tag{2.2}$$

When less frequently, PMI has been known to be exorbitant high value. For reducting the exorbitant value, Espresso substitute (2.3) citepantel2004 for (2.2)

$$pmi(x, y) = \log \frac{P(x, y)}{P(x)P(y)} * \frac{C_{xy}}{C_{xy} + 1} * \frac{min(\sum_{i=1}^{n} C_{x_i}, \sum_{j=1}^{m} C_{y_j})}{min(\sum_{i=1}^{n} C_{x_i}, \sum_{j=1}^{m} C_{y_j}) + 1} \tag{2.3}$$

where C_{xy} is a frequency of co-occur x_i and y_j. C_{x_i} is a frequency of x_i. C_{y_j} is a frequency of y_j. n is a type number of x. m is a type number of y.

3.3 Ranking Relation Instances

Intuitively, a reliable relation instance is one that is highly associated with multiple reliable patterns. Hence, analogously to the above pattern reliability measure, Espresso defines the reliability $r_\iota(i)$ of instance i as:

$$r_\iota(i) = \frac{1}{|P|} \sum_{p \in P} \frac{pmi(i, p)}{max_{pmi}} \times r_\pi(p) \tag{2.4}$$

where $r_\pi(p)$ is the reliability of pattern p, defined above in (2.1), and max_{pmi} is as before. $r_\iota(i)$ and $r_\pi(p)$ are recursively defined, where $r_\iota(i) = 1$ for each manually supplied seed instance i[1].

[1]For our extension, $r_\iota(i) = -1$ for each manually supplied negative instance.

Chapter 3

Extended Espresso

1. Introduction

The growing interest in practical Natural Language Processing (NLP) applications such as Question Answering, Information Extraction and Multi-Document Summarization has greatly increased for identification of relations between textual fragments such as entailment and causal relations. Such applications often need to rely on a large amount of lexical semantic knowledge. For example, a causal (and entailment) relation holds between the verb phrases *wash something* and *something is clean*, which reflects the commonsense notion that if someone has washed something, this object is clean as a result of the washing event. A crucial issue is how to obtain and maintain a potentially huge collection of event relation instances.

Motivated by this problem, several research groups have reported on experiment on automatic acquisition of causal, temporal and entailment relations between event mentions (typically verbs or verb phrases) [14, 9, 4, 28, 21, 29]. The common idea behind them is to use a small number of manually selected generic lexico-syntactic co-occurrence patterns (LSPs or simply patterns). *to Verb-X and then Verb-Y*, for example, is used to obtain temporal relations such as *marry* and *divorce* [4]. The use of such generic patterns, however, tends to have high recall and low precision, which requires an additional component for pruning extracted relations. Approaches to pruning bad relations can be broadly classified into two groups: either by devising heuristic scores [4, 28, 29] or by training heavily-supervised classifiers for disambiguation [9].

This chapter explores a third way for enhancing present LSP-based methods for event relation acquisition. The basic idea is inspired by the following recent findings in relation extraction [23, 20], which aims at extracting semantic relations between *entities* (as opposed to *events*) from texts.

(a) The use of generic patterns tends to be high recall but low precision, which requires an additional component for pruning.

(b) On the other hand, there are specific patterns that are highly reliable but they are much less frequent than generic patterns and each makes only a small contribution to recall.

(c) Combining a few generic patters with a much larger collection of reliable specific patterns boosts both precision and recall. Such specific patterns can be acquired from a very large corpus with seeds.

Given these insights, an intriguing question is whether the same story applies to event relation acquisition as well or not. In this thesis, we explore this issue through the following steps. First, while previous methods use only verb-verb co-occurrences, we use co-occurrences between verbal nouns and verbs such as *cannot ⟨find out (something)⟩ due to the lack of ⟨investigation⟩* as well as verb-verb co-occurrences.

This extension dramatically enlarges the pool of potential candidate LSPs (Section Section 3.8). Second, we extend Pantel and Pennacchiotti [20]'s Espresso algorithm, which induces specific reliable LSPs in a bootstrapping manner for entity relation extraction, so that the extended algorithm can apply to event relations (Sections Section 3.3 to Section 3.4). Third, we report on the present results of our empirical experiments, where the extended algorithm is applied to a Japanese 500M-sentence Web corpus to acquire two types of event relations, *action-effect* and *action-means* relations (Section Section 4)

2. Contribution

Several research groups have reported automatic event relation acquisition methods. The methods use manually selected generic lexico-syntactic co-occurrence patterns. However, the methods tend to be high recall but low precision. On the other hand, another research group has reported an automatic entity relation acquisition method. The

Table 3.1. Categories of pattern-based methods

	How to build cooccurrence patterns?	
Type of instances	By hand (only generic patterns)	By machine (genric and specific patterns)
Entity	Methods exist.	Methods exist.
Event	Methods exist.	We propose Extended Espresso.

method automatically acquires generic lexico-syntactic cooccurrence patterns, and also automatically acquire specific lexico-syntactic co-occurrence patterns. The method boosts precision and recall. However, the method is for entity relation acquisition, a similar method for event relation acquisition does not exist as well. We propose a method to automatically acquire generic lexico-syntactic cooccurrence patterns, and also automatically acquire specific lexico-syntactic cooccurrence patterns for event relation acquisition. Table 3.1 shows a difference of some pattern based methods.

3. Method

3.1 Introduction

Espresso has following advantages.

- Semi-automatic knowledge acquisition
- Acquiring a large amount of knowledge from large text
- Reasonable precision using bootstrapping

Espresso has many advantages, Espresso is a desirable pattern-based method for automatic noun relation acquisition at present time. We therefore extend Espresso for automatic event relation acquisition as a matter of course. In this section, we describe *Extended Espresso* that extend Espresso for event relation acquisition. Figure 3.1 shows Extended Espresso.

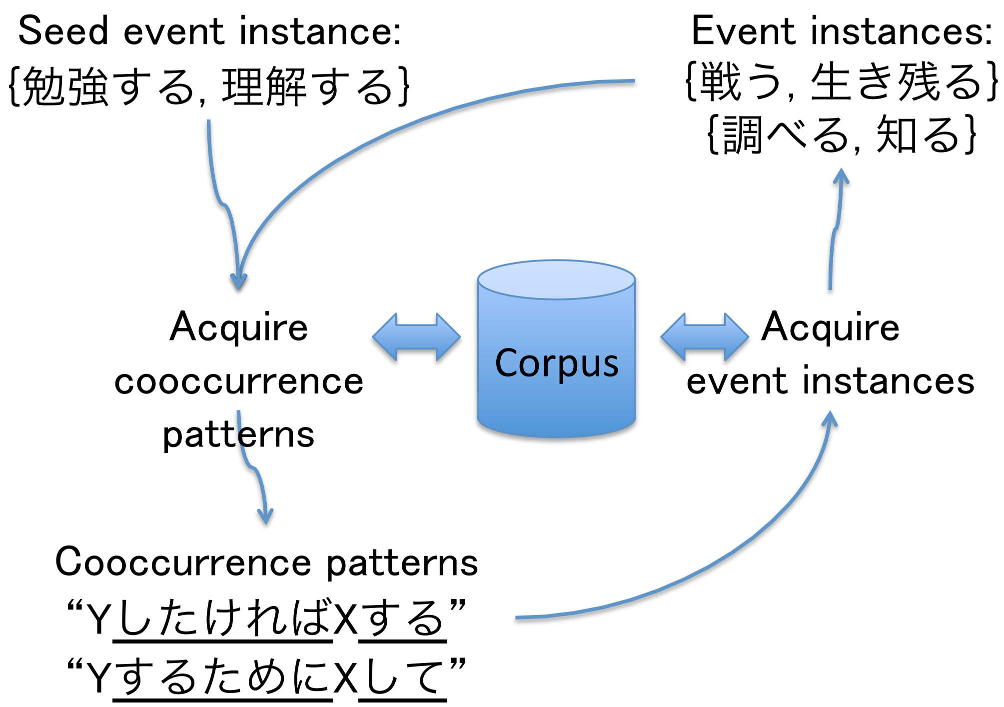

Figure 3.1. Extended Espresso

3.2 Event expressions

Verb-argument Structure

We decide to represent an event by verb-argument structure to generalize variable expressions. For example, "『坊ちゃん』は夏目漱石によって発表された" (Bocchan-ha Natsume-Souseki-ni-yotte happyou-sareta; Bocchan was published by Souseki Natsume.) and "夏目漱石が『坊ちゃん』を発表する" (Natsume-Souseki-ga Bocchan-wo happyou-suru; Souseki Natsume publishes Bocchan.) are different syntactic structures. However, we can generalize the phrases to a verb-argument structure: "夏目漱石ガ『坊ちゃん』ヲ発表する" (Natsume-Souseki-GA / Bocchan-WO / happyou-suru; Souseki Natsume).

Form of Morphemes

We regard a form of morphemes of an event expression as an original form because of generalizing an event expression. For example, we regard "走った" as "走る".

However, if a form is followed by a specific expression, we do not regard exceptionally a form as an original form. The expression is following.

- Passive expression ("〜される")
- Causative expression ("〜させる")
- Possible expression ("〜られる")
- Desire expression ("〜したい")

For example, we do not regard "走りたい" (hashiritai; want to run) as "走る" (hashiru; run), we regard "走りたい" as "走りたい". In addition, we regard "走りたかった" (hashiri-takatta; wanted to run) as "走りたい".

We also apply the rule to a verbal noun following verb "する" (suru; do). For example, we regard "研究したかった" (kenkyuu-shitakatta; wanted to research) as "研究したい" (kenkyuu-shitai; want to research).

Meaningless / Ambiguously Words

We do not regard a meaningless word (e.g. "ある" (aru), "なる" (naru), "する" (suru)) and a very ambiguously word as an event expression. The reason is that an event relation between meaningless / ambiguously words is very noisy. However, we do not exclude the words. We merge the words with a case of immediately before the words[1], we regard the merged word as an event expression. For example, we do not regard "付く" (tsuku; stick) of "焦げ目が付く" (kogeme-ga-tsuku; stick burn / burn) as event expression, however we regard "焦げ目が付く" as an event expression. In addition, if the words is "する" and a case of immediately before the words is Wo-case, we omit Wo-case. For example, we regard "研究をする" as "研究する"[2]. In our experiments, we regard following words as meaningless or very ambiguously.

> 「ある」「いく」「いる」「おこなう」「かる」「する」「ちる」「できる」「とめる」「なる」「みる」「やる」「付く」「伝える」「似る」「作る」「使う」「保つ」「入る」「入れる」「出す」「出る」「分かる」「加える」「取り戻す」「取る」「向ける」「含む」「呼ぶ」「因る」「増える」「変わる」「寄る」「居る」「建つ」「引く」「弱る」「得る」「思う」「救う」「断つ」「書く」「止める」「残る」「減る」「生じる」「知る」「立つ」「終る」「終わる」「終了」「経つ」「経る」「続ける」「考え」「考える」「聞く」「行う」「見える」「見せる」「見る」「見失う」「言う」「言える」「話す」「語る」「読む」「踏み切る」「込める」「通る」「進む」「進める」「変化（する）」「対処（する）」「影響（する）」「拡大（する）」「決定（する）」「縮小（する）」「進展（する）」「開始（する）」「関係（する）」

The words are meaningless words as "付く" and very ambiguously words as "開始する" (kaishi-suru; begin) or "影響する" (eikyou-suru; effect).

3.3 Selection of Arguments

One major step from the extraction of entity relations to the extraction of event relations is how to address the issue of *generalization*. In entity relation extraction, relations are typically assumed to hold between chunks like named entities or simply between one-word terms, where the issue of determining the appropriate level of the generality of extracted relations has not been salient. In event relation extraction, on the other hand, this issue immediately arises. For example, the co-occurrence sample in (1) suggests

[1] The words are verb, the words therefore have some cases.

[2] "研究する" is event expression in our experiments

the *action-effect* relation between *niku-o yaku* (grill the meat) and *(niku-ni) kogeme-ga tsuku* ((the meat) gets brown)[3].

(1) *(kogeme-ga tsuku) -kurai niku-o yaku*
a burn-NOM get -so that meat-ACC grill
grill the meat so that it gets brown
(grill the meat to a deep brown)

In this relation, the argument *niku* (meat) of the verb *yaku* (grill) can be dropped and generalized to *something to grill*; namely the *action-effect* relation still holds between *X-o yaku* (grill X) and *X-ni kogeme-ga tsuku* (X gets brown). On the other hand, however, the argument *kogeme* (a burn) of the verb *tsuku* (get) cannot be dropped; otherwise, the relation would no longer hold.

One straightforward way to address this problem is to expand each co-occurrence sample to those corresponding to different degrees of generalization and feed them to the relation extraction model so that its scoring function can select appropriate event pairs from expanded samples. For example, co-occurrence sample (1) is expanded to those as in (2):

(2) a. *(kogeme-ga tsuku) -kurai niku-o yaku*
a burn-NOM get -so that meat-ACC grill

b. *(tsuku) -kurai niku-o yaku*
get -so that meat-ACC grill

c. *(kogeme-ga tsuku) -kurai yaku*
a burn-NOM get -so that grill

d. *(tsuku) -kurai yaku*
get -so that grill

In practice, in our experiments (Section 4), we restrict the number of arguments for each event up to one to avoid the explosion of the types of infrequent candidate relation instances.

[3]The parenthesis in the first row of (1) indicates a subordinate clause.

3.4 Dependency-based Co-occurrence Patterns

Introduction

The original Espresso encodes patterns simply as a word sequence because entity mentions in the relations it scopes tend to co-occur locally in a single phrase or clause. For example, a pattern "*x* such as *y*" represents *is-a* relation. The pattern can acquire *is-a* relation instances such as "Italy", "country" from "countries such as Italy".

In event relation extraction, however, co-occurrence patterns of event mentions in the relations we consider (causal relations, temporal relations, etc.) can be captured better as a path on a syntactic dependency tree because (i) such mention pairs tend to co-occur in a longer dependency path and (ii) as discussed in Section 3.3. We want to exclude the arguments of event mentions from co-occurrence patterns, which would be difficult with word sequence-based representations of patterns.

A Japanese sentence can be analyzed as a sequence of base phrase (BP) chunks called *bunsetsu* chunks, each which typically consists of one content (multi-)word followed by functional words. We assume each sentence of our corpus is given a dependency parse tree over its BP chunks. Let us call a BP chunk containing a verb or verbal noun an *event chunk*. We create a co-occurrence sample from any pair of event chunks that co-occur if either[4] (Figure 3.2):

(a) One event chunk depends directly on the other.

(b) One event chunk depends indirectly on the other via one intermediate chunk.

Additionally, we apply the Japanese functional expressions dictionary [15] to a co-occurrence pattern for generalization.

In Figure 3.3, for example, the two event chunks, "退職後に" (taishoku-go-ni; after retirement) and "始める" (hajimeru; begin), meet the condition (b) above and the dependency path designated by underline is identified as a candidate co-occurrence pattern. The argument "PC を" (PC-o; PC-ACC) of the verb "始める" (hajimeru; begin) is excluded from the path.

[4]We performed preliminary experiments to decide dependency patterns for extracting co-occurrence samples. As a result of the experiments we decide to use the above two dependency patterns. On the other hand, we decided to omit another dependency pattern that the two event chunk depends directly on the same arbitrary chunk because co-occurrence samples by the pattern are much noisy.

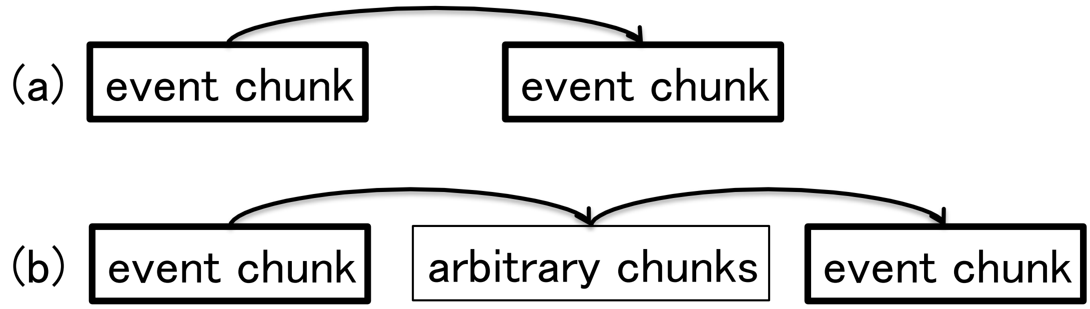

Figure 3.2. Dependency patterns

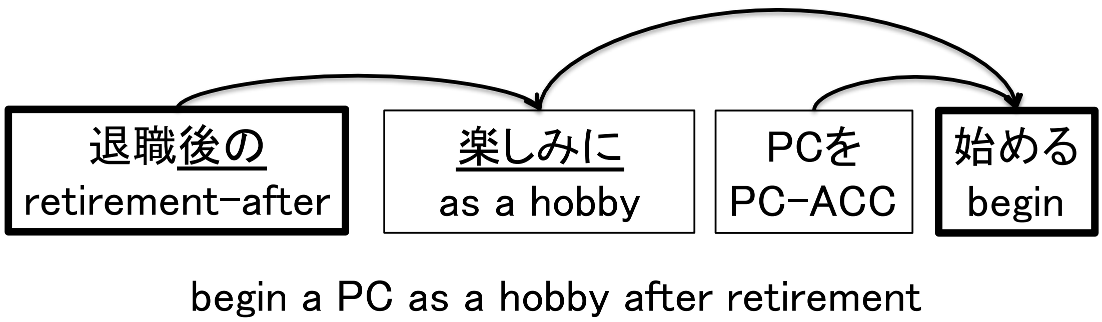

Figure 3.3. An example of co-occurrence pattern

Section 3.4 shows a detail of rules of co-occurrence pattern. To design co-occurrence pattern for event relation acquisition, we should consider following conditions.

- (a) A pattern sufficiently represent a relation between events.
- (b) A pattern and related events are sufficiently co-occurrence in corpus.

The above condition (a) and (b) are incompatible, it is importance to balance the conditions. We therefore decide the rules of co-occurrence pattern by preliminary experiments.

Detail of Dependency-based Co-occurrence Patterns

The co-occurrence is constructed words between event expressions, a word after an event, POS (Part Of Speech) of events, volitionality of events. We show details.

If event chunks satisfy a dependency pattern, we build a co-occurrence pattern. The co-occurrence pattern is structured by following elements.

(a) A string of functional words after content words representing an event in a forward event bunsetsu chunk.

(b) A string of words between event bunsetsu chunks in dependency tree.

(c) A string in a backward event bunsetsu chunk or a bunsetsu chunk depended by the backward event bunsetsu chunk,

  (c1) A string "ない" ("not") if a negative expression is included in the bunsetsu chunk (e.g. "〜ない", "〜ません", "〜せず", "〜ぬ").

  (c2) A string "できる" ("can") if a possible expression is included in the bunsetsu chunk (e.g. "〜できる", "〜出来る", "〜することができる", "〜することが出来る", "〜することが可能だ").

  (c3) A string "できない" ("can not") if a negative positive expression is included in the bunsetsu chunk.

(d) Strings of POS of event expressions.

(e) Strings of volitionality of event expressions.

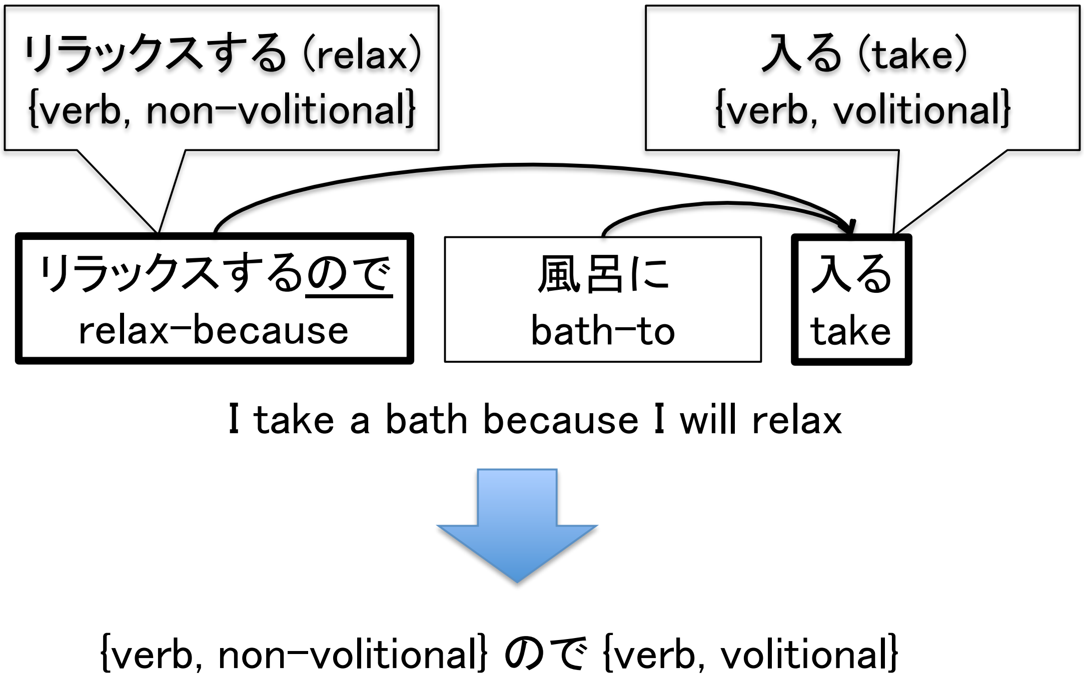

Figure 3.4. An example of co-occurrence pattern (dependency pattern 1)

Examples of Co-occurrence Pattern

We show examples of co-occurrence patterns.

Example 1 Figure 3.4 has two event bunsetsu chunks: "リラックスするので" and "入る", the chunks satisfy a dependency pattern. We therefore can extract a co-occurrence pattern and a pair of event instances. The co-occurrence pattern is structured following elements.

(a) "ので": A string of functional words of the forward event bunsetsu chunk

(d) ⟨verb, non-volitional⟩: POS and volitional of the forward event bunsetsu chunk

(e) ⟨verb, volitional⟩: POS and volitional of the backward event bunsetsu chunk

A co-occurrence pattern from the example is "⟨verb, non-volitional⟩ので⟨verb, volitional⟩".

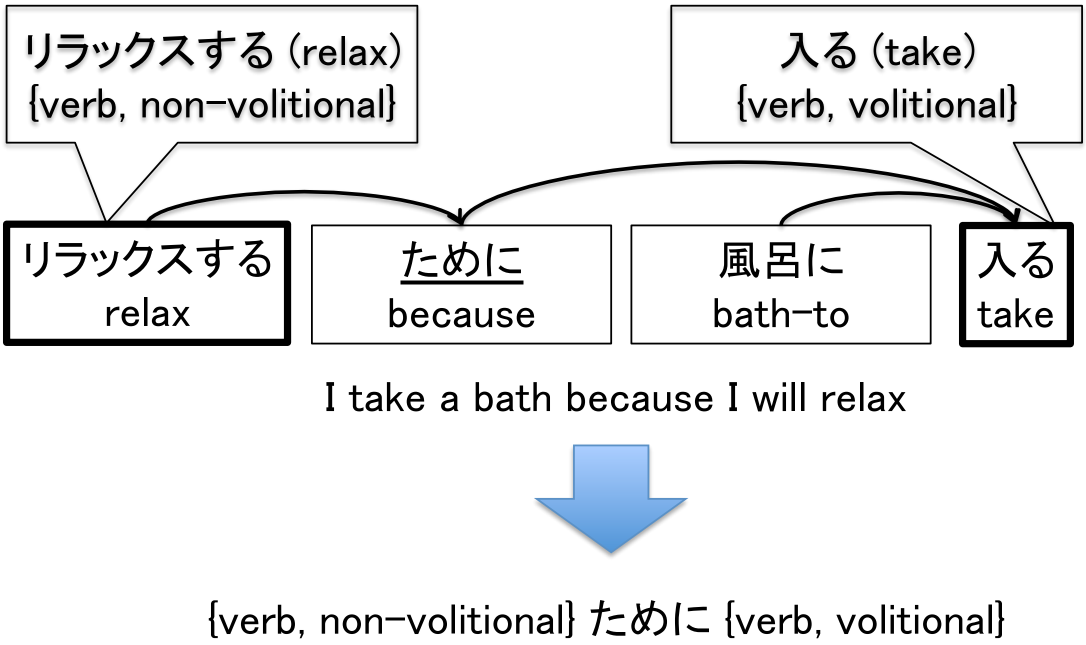

Figure 3.5. An example of co-occurrence pattern (dependency pattern 2)

Example 2 Figure 3.5 has two event bunsetsu chunks: "リラックスする" and "入る", the chunks satisfy a dependency pattern. We therefore can extract a co-occurrence pattern and a pair of event instances. The co-occurrence pattern is structured following elements.

(b) "ために": A string of words between event bunsetsu chunks in dependency tree.

(d) ⟨verb, non-volitional⟩: POS and volitional of the forward event bunsetsu chunk

(e) ⟨verb, volitional⟩: POS and volitional of the backward event bunsetsu chunk

A co-occurrence pattern from the example is "⟨verb, non-volitional⟩ために⟨verb, volitional⟩".

Example 3 Figure 3.5 has two event bunsetsu chunks: "退職後の" and "始める", the chunks satisfy a dependency pattern. We therefore can extract a co-occurrence pattern and a pair of event instances. The co-occurrence pattern is structured following elements.

(a) "後の": A string of functional words of the forward event bunsetsu chunk

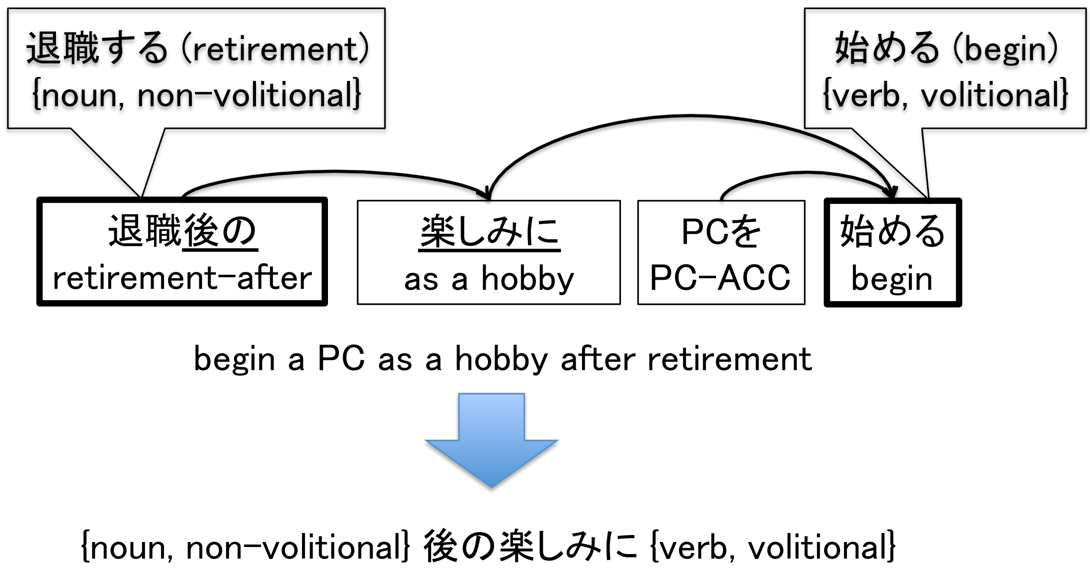

Figure 3.6. An example of co-occurrence pattern (verbal noun)

(b) "楽しみに": A string of words between event bunsetsu chunks in dependency tree.

(d) ⟨noun, non-volitional⟩: POS and volitional of the forward event bunsetsu chunk

(e) ⟨verb, volitional⟩: POS and volitional of the backward event bunsetsu chunk

A co-occurrence pattern from the example is "⟨noun, non-volitional⟩後の楽しみに ⟨verb, volitional⟩".

3.5 Generalization of Co-occurrence Patterns

We generalize co-occurrence patterns. In this sub-section, we describe generalization rules. The original Espresso use also similar rules for a co-occurrence pattern.

Generalization of Functional Expressions

We employ Japanese Functional Expressions Dictionary [15] to generalize functional words in a co-occurrence pattern. To achieve it, we use a "Level" of Japanese Functional Expressions Dictionary, and replace a Level-9 word with a Level-3 word from words of a co-occurrence pattern.

In addition, we remove a functional expression “ます” to replace “〜します” with “〜する” from a co-occurrence pattern. Similarly, we remove also a functional expression “と思う” to replace “〜すると思う” with “〜する” from a co-occurrence pattern.

Moreover, we remove punctuation marks, symbols and some suffix words (“達”, “等”) from a co-occurrence pattern.

Generalization of Named Entity

To generalize a co-occurrence pattern, we employ a named entity recognition and a POS (part of speech) analyzation.

We replace a string of a named entity expression with a string of a class name of the named entity expression from a co-occurrence pattern. For example, we consider to acquire a pair of event instances between “待つ” and “来る” from Figure 3.7. The co-occurrence pattern of the example is “こと３０分で”[5]. We consider a situation to acquire event instances using the pattern, it is better to replace sub-string “３０分” (thirty minutes) of the pattern with an abstract word of meaning various time. Using the abstract word, the pattern matches not only “３０分” (thirty minutes), but also various situations (e.g. “４０分”(forty minutes), “１時間” (one hour)). We therefor replace a sub-string “３０分” with a abstract string “TIME”. The “TIME” is a named entity class. Finally, the pattern becomes “こと **TIME** で”.

In our experiments, we employ CaboCha [12] for named entity recognition, we also employ the named entity class defined by IREX[6]: ARTIFACT, DATE, LOCATION, MONEY, OPTIONAL, ORGANIZATION, PERCENT, PERSON, TIME.

In addition, we employ POS tagger for a word that Cabocha fails to recognize named entity. In our experiments, we employ CaboCha for POS tagging[7]. We also regard following POSs as named entity expression.

- 名詞-接尾-人名 (Noun-suffix-person’s name)
- 名詞-接尾-地域 (Noun-suffix-location)
- 名詞-接尾-助数詞 (Noun-suffix-counter)

[5]We omit information of event expressions form the pattern for explain.
[6]Cabocha employ IREX defined named entity class.
[7]CaboCha employ IPA POS definition.

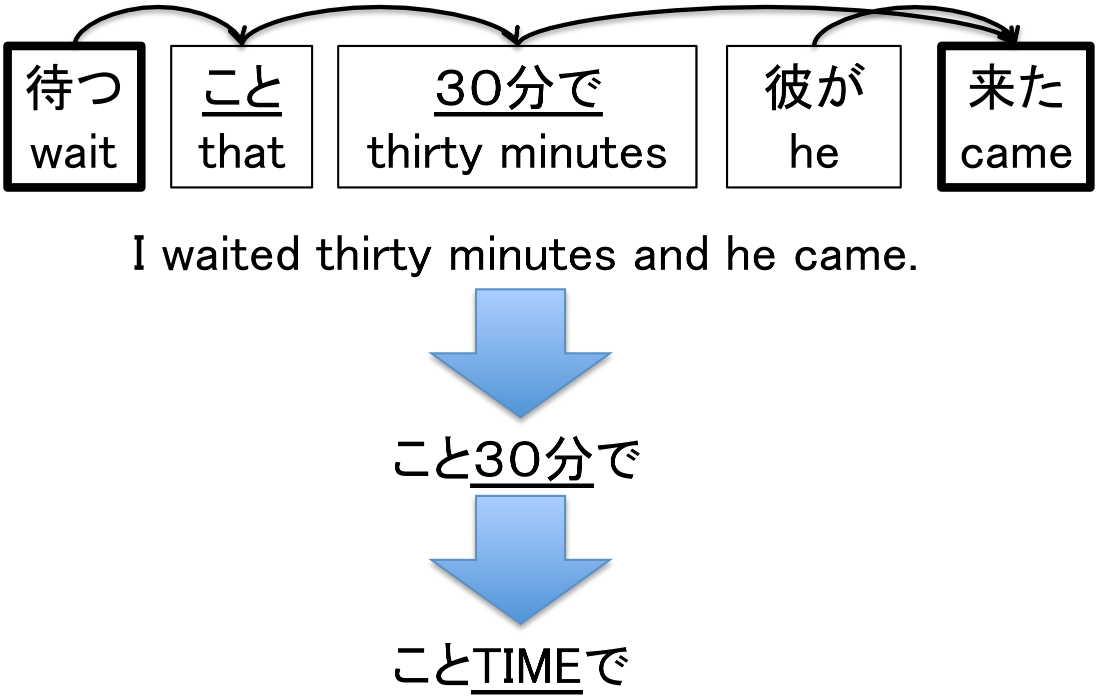

Figure 3.7. An example of generalization of a named entity expression

- 名詞-数 (Noun-numeral)
- 名詞-固有名詞-一般 (Noun-named entity-general)
- 名詞-固有名詞-人名 (Noun-named entity-person's name)
- 名詞-固有名詞-組織 (Noun-named entity-organization)
- 名詞-固有名詞-地域 (Noun-named entity-location)

3.6 Volitionality of Events

Inui et al. [9] discuss how causal relations between events should be typologized for the purpose of semantic inference and classify causal relations basically into four types — Effect, Means, Precondition and Cause relations — based primarily on the volitionality of involved events. For example, Effect relations hold between volitional actions and their resultative non-volitional states/happenings/experiences, while Cause relations hold between only non-volitional states/happenings/experiences.

Following this typology, we are concerned with the volitionality of each event mention. For our experiments, we manually built a lexicon of over 12,000 verbs (including verbal nouns) with volitionality labels, obtaining 8,968 volitional verbs, 3,597 non-volitional and 547 ambiguous. Volitional verbs include *taberu* (eat) and *kenkyu-suru* (research), while non-volitional verbs include *atatamaru* (get warm), *kowareru* (to break-*vi*) and *kanashimu* (be sad). We discarded the ambiguous verbs in the experiments.

3.7 Modification of Reliability Equations

The results of preliminary experiments, we change (2.1) into (3.1), change (2.4) into (3.2). In addition, we normalize reliability between -1 and 1 at each stage of the bootstrap.

$$r'_\pi(p) = \sum_{i \in I} pmi(i,p) \times r'_\iota(i) \tag{3.1}$$

$$r'_\iota(i) = \sum_{p \in P} pmi(i,p) \times r'_\pi(p) \tag{3.2}$$

The reason to remove $|I|$ from (2.1) and remove $|P|$ from (2.4) is to emphasize the assumption of Espresso. The assumption is that reliability of an instance associated by highly reliability patterns is high and reliability of a pattern assosiated by highly reliability instances is also high.

The reason to remove max_{pmi} from (2.1) and (2.4) is to normalize a reliability between -1 and +1. A reliability of the original expressions is not in between -1 and +1. In the changed expressions, we divide a reliability by a maximum reliability each bootstrap stages instead of dividing by max_{pmi}.

3.8 Co-occurrences with Verbal Nouns

Most previous methods for event relation acquisition rely on verb-verb co-occurrences because verbs (or verb phrases) are the most typical device for referring to events. However, languages have another large class of words for event reference, namely verbal nouns or nominalized forms of verbs. In Japanese, for example, verbal nouns such as *kenkyu* (research) constitute the largest morphological category used for event reference.

Japanese verbal nouns have dual statuses, as verbs and nouns. When occurring with the verb *suru* (do-PRES), verbal nouns function as a verb as in (3a). On the other hand, when accompanied by case markers such as *ga* (NOMINATIVE) and *o* (ACCUSATIVE), they function as a noun as in (3b). Finally, but even more importantly, when accompanied by a large variety of suffixes, verbal nouns constitute compound nouns highly productively as in (3c).

(3) a. *Ken-ga* *gengo-o* *kenkyu-suru*
Ken-NOM language-ACC research-PRES
Ken researches on language.

b. *Ken-ga* *gengo-no* *kenkyu-o* *yame-ta*
Ken-NOM language-on research-ACC quit-PAST
Ken quitted research on language.

c. *-sha* (person):
e.g. *kenkyu-sha* (researcher)
-shitsu (place):
e.g. *kenkyu-shitsu* (laboratory)

-*go* (after):

e.g. *kenkyu-go* (after research)

These characteristics of verbal nouns can be made use of to substantially increase both co-occurrence instances and candidate co-occurrence patterns (see Section 4.1 for statistics). For example, the verbal noun *kenkyu* (research) often co-occurs with the verb *jikken* (experiment) in the pattern of (4a). From those co-occurrences, one may learn that *jikken-suru* (to experiment) is an action that is often taken as a part of *kenkyu-suru* (to research). In such a case, we may consider a pattern as shown in (4b) useful for acquiring *part-of* relations between actions.

(4) a. *kenkyu-shitsu-de* *jikken-suru*
research-place-in experiment-VERB
conduct experiments in the laboratory

b. *(Act-X)-shitsu-de* *(Act-Y)-suru*
(Act-X)-place-in (Act-X)-VERB
(Act-Y) is often done in doing (Act-X)

When functioning as a noun, verbal nouns are potentially ambiguous between the event reading and the entity/object reading. For example, the verbal noun *denwa* (phone) in the context *denwa-de* (phone-by) may refer to either a phone-call event or a physical phone. While, ideally, such event-hood ambiguities should be resolved before collecting co-occurrence samples with verbal nouns, we simply use all the occurrences of verbal nouns in collecting co-occurrences in our experiments. It is an interesting issue for future work whether event-hood determination would have a strong impact on the performance of event relation extraction.

4. Experiments

4.1 Settings

For an empirical evaluation, we used a sample of approximately 500M sentences taken from the Web corpus collected by Kawahara and Kurohashi [10]. The sentences were part-of-speed-parsed with ChaSen [1], and were dependency-parsed with CaboCha [12]. Before using the sentences for experiments, we removed some sentences that satisfy one of the following conditions.

- A number of bunsetsu chunks of a sentence is one or more than thirteen.
- A number of characters of a bunsetsu chunk of a sentence is more than thirty three.
- A sentence has any symbols without punctuations.
- A sentence has unknown words.
- A sentence does not include more than two event mentions.

In addition, we removed punctuations from sentences. The sentences were extracted co-occurrence samples of event mentions. Event mentions with patterns whose frequency was less than 20 were discarded in order to reduce computational costs.

As a result, we obtained 34M co-occurrence tokens with 11M types. Note that among those co-occurrence samples 15M tokens (44%) with 4.8M types (43%) are those with verbal nouns, suggesting the potential impacts of using verbal nouns.

4.2 Evaluation

In our experiments, we considered one of Inui et al. [9]'s four types of causal relations: *action-effect* relations (Effect in Inui et al.'s terminology). An *action-effect* relation holds between events x and y if and only if non-volitional event y is likely to happen as either a direct or indirect effect of volitional action x. For example, the action *X-ga undou-suru* (X exercises) and the event *X-ga ase-o kaku* (X sweats) are considered to be in this type of relation.

Note that in these experiments we do not differentiate between relations with the same subject and those with a different subject. However, we plan to conduct further experiments in the future that make use of this distinction.

In addition, we have collected *action-effect* relation instances for a baseline measure. The baseline consists of instances that co-occur with eleven patterns that indicate *action-effect* relation. The difference between the extended Espresso and baseline is caused by the low number and constant scores of patterns.

Table 3.2. Examples of acquired co-occurrence patterns and relation instances for the action-effect relation

freq	co-occurrence patterns	relation instances
94477	⟨*verb;action*⟩***temo***⟨*verb;effect*⟩**nai** (to do ⟨action⟩ though ⟨effect⟩ dose not happen)	*sagasu::mitsukaru* (search::be found), *asaru::mitsukaru* (hunt::be found), *purei-suru::kuria-suru* (play::finish)
6250	⟨*verb;action*⟩***takeredomo***⟨*verb;effect*⟩**nai** (to do ⟨action⟩ though ⟨effect⟩ dose not happen)	*shashin-wo-toru::toreru* (shot photograph::be shot), *meiru-wo-okuru::henji-ga-kaeru* (send a mail::get an answer)
1851	⟨*noun;action*⟩***wo-shitemo***⟨*verb;effect*⟩**nai** (to do ⟨action⟩ though ⟨effect⟩ dose not happen)	*setsumei-suru::nattoku-suru* (explain::agree), *siai-suru::katsu* (play::win), *siai-suru::makeru* (play::lose)
1329	⟨*verb;action*⟩***yasukute***⟨*adjective;effect*⟩ (to simply do ⟨action⟩ and ⟨effect⟩)	*utau::kimochiyoi* (sing::feel good), *hashiru::kimochiyoi* (run::feel good)
4429	⟨*noun;action*⟩***wo-kiite***⟨*verb;effect*⟩ (to hear ⟨action⟩ so that ⟨effect⟩)	*setsumei-suru::nattoku-suru* (explain::agree), *setsumei-suru::rikai-dekiru* (explain::can understand)

4.3 Results

We ran the extended Espresso algorithm starting with 971 positive and 1069 negative seed relation instances. As a result, we obtained 34,993 co-occurrence patterns with 173,806 relation instances after 20 iterations of pattern ranking/selection and instance ranking/selection. The threshold parameters for selecting patterns and instances were decided in a preliminary trial. Some of the acquired patterns and instances are shown in Table 3.2.

4.4 Precision

To estimate precision, 100 relation instances were randomly sampled from each of four sections of the ranks of the acquired instances for each of the two relations (1–500, 501–1500, 1501–3500 and 3501–7500), and the correctness of each sampled instance was judged by two graduate students (i.e. 800 relation instances in total were judged).

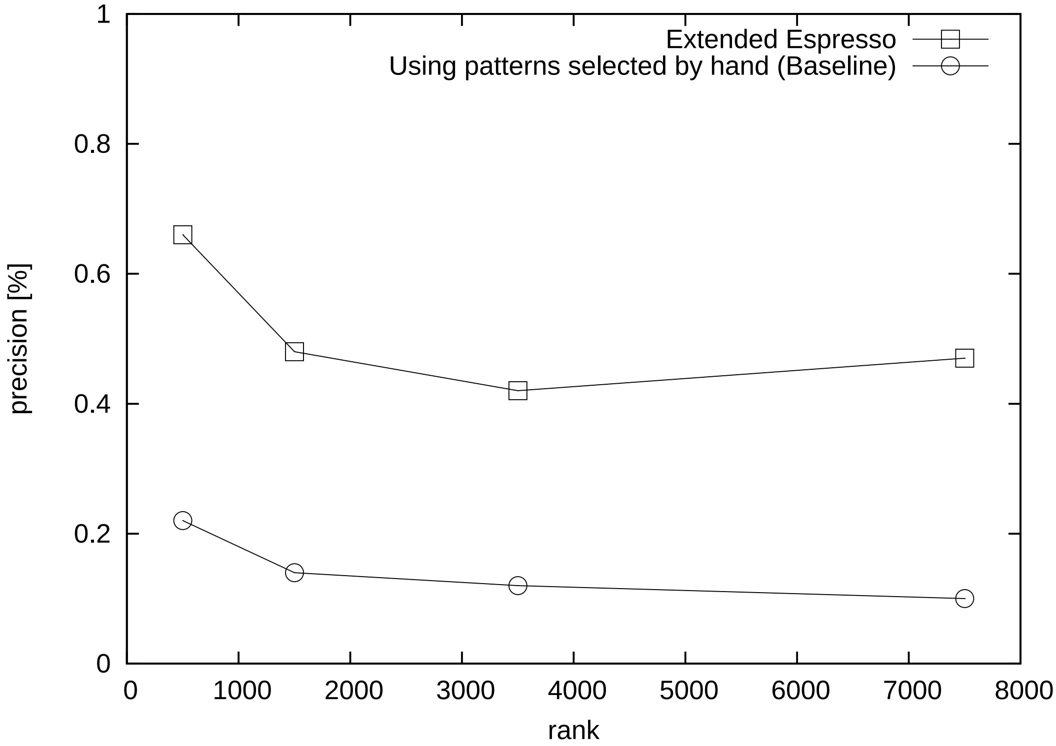

Figure 3.8. Precision of Extended Espresso

Note that in these experiments we asked the assessors to both (a) the degree of the likeliness that the effect takes place and (b) which arguments are shared between the two events. For example, while *nomu* (drink) does not necessarily result in *futsukayoi-ni naru* (have a hangover), the assessors judged this pair correct because one can at least say that the latter *sometimes* happens *as a result of* the former. For criterion (b), as shown in Table 3.2, the relation instances judged correct include both the X-*ga* VP_1::X-*ga* VP_2 type (i.e. two subjects are shared) and the X-*o* VP_1::X-*ga* VP_2 type (the object of the former and the subject of the latter are shared). The issue of how to control patterns of argument sharing is left for future work.

Figure 3.8 shows the assessed samples. It compare between Extended Espresso and a baseline. The baseline is to use few co-occurrence patterns created by hand. Extended Espresso outperform the baseline. The result shows followings.

- Original Espresso has capable of applying event relation acquisition.
- Our extension is appropriate for event relation acquisition.

As a result in judgment, the inter-assessor agreement was moderate. The kappa statistics was 0.53 for Extended Espresso and 0.55 for baseline.

4.5 Effect of Seed Size

We reran the extended Espresso algorithm for the *action-effect* relation, starting with 500 positive and 500 negative seed relation instances. The precision is shown in Figure 3.9[8]. This precision is fairly lower than that of *action-effect* relations with all seed instances. Additionally, the number of seed instances affects the precision of both higher-ranked and lower-ranked instances. This result indicates that while the proposed algorithm is designed to work with a small seed set, in reality its performance severely depends on the number of seeds.

4.6 Effect of Using Verbal Nouns

We also examine the effect of using verbal nouns. We compare a precision including an effect of verbal nouns with a precision excluding an effect of verbal nouns. We show procedures to exclude the effect of verbal nouns.

[8] It was only judged by one assessor.

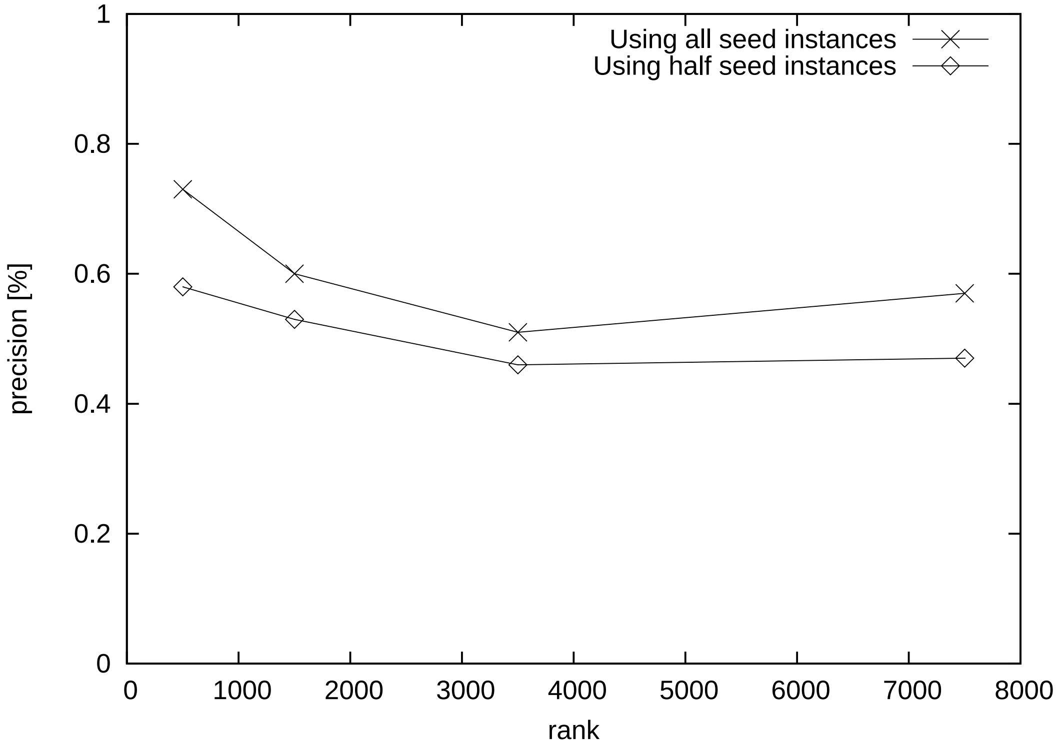

Figure 3.9. Effect of seed size

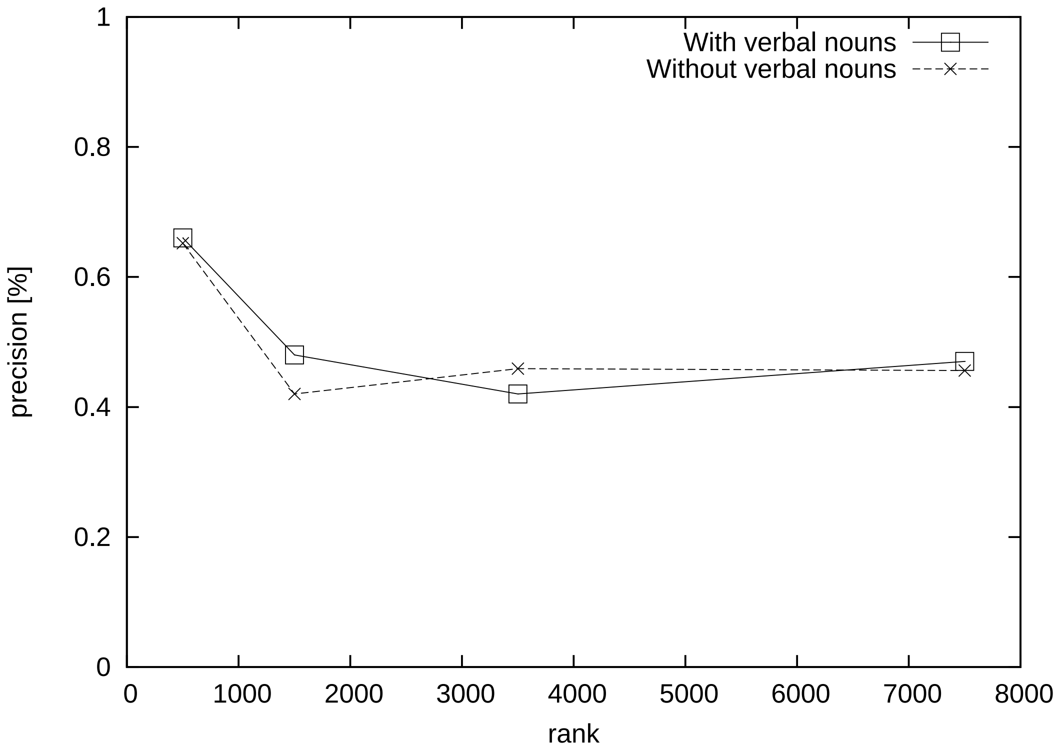

Figure 3.10. Presition without verbal nouns

- If verbal noun occurs with pattern, reliability of co-occurrence pattern is 0.
- We recalculate reliability of instances.
- We reorder to instances by reliability.

Figure 3.10 shows that the precision including the effect of verbal nouns and the precision excluding the effect of verbal nouns. The precision of results are similar. Consequently, the verbal nouns do not affect the precision.

4.7 Argument Selection

According to our further investigation on argument selection, 49 instances (12%) of the correct *action-effect* relation instances that are judged correct have a specific argument in at least one event, and all of them would be judged incorrect (i.e. over-generalized) if they did not have those arguments (Recall the example of *kogeme-ga tsuku* (get brown)

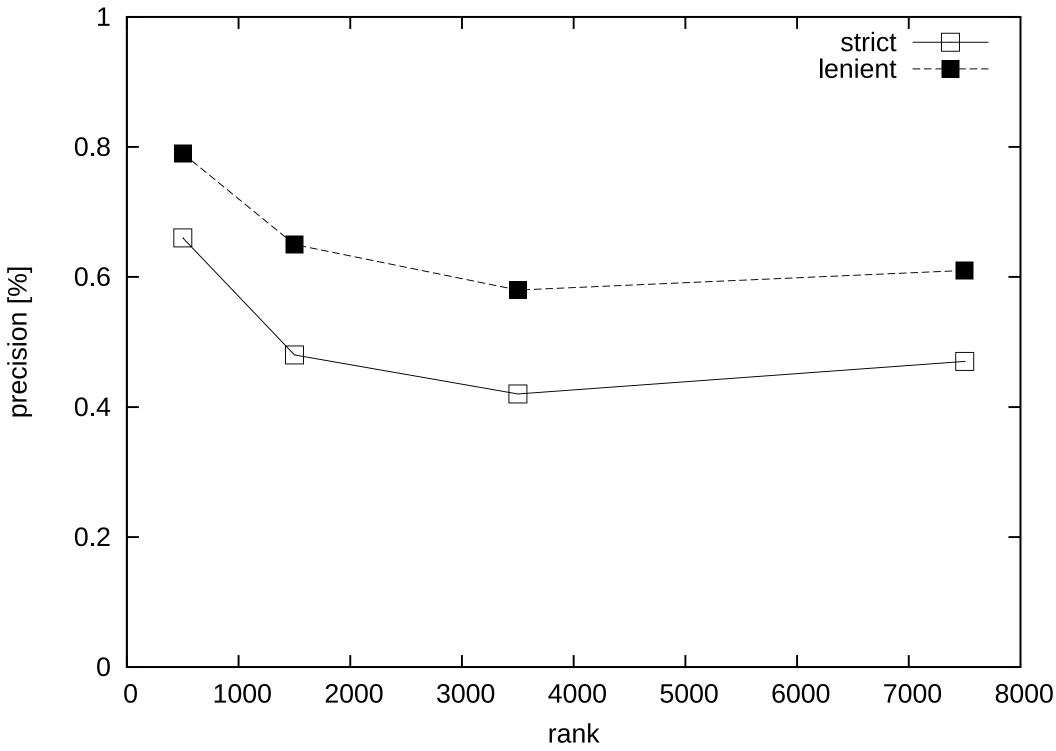

Figure 3.11. Presition without arguments

in Section 3.3). This figure indicates that our method for argument selection works to a reasonable degree.

However, clearly there is still much room for improvement. According to our investigation, up to 26% of the instances that are judged incorrect could be saved if appropriate arguments were selected. For example, *X-ga taberu* (X eats) and *X-ga shinu* (X dies) would constitute an *action-effect* relation if the former event took such an argument as *dokukinoko-o* (toadstool-ACC). The overall precision could be boosted if an effective method for argument selection method were devised.

5. Conclusion and Future Work

In this chapter, we have addressed the issue of how to learn lexico-syntactic patterns useful for acquiring event relation knowledge from a large corpus, and proposed several extensions to a state-of-the-art method originally designed for entity relation extraction, reporting on the present results of our empirical evaluation. The results show followings.

(a) There are indeed specific co-occurrence patterns useful for event relation acquisition.

(b) The use of co-occurrence samples involving verbal nouns has positive impacts on both recall and precision.

(c) Over five thousand relation instances are acquired from the 500M-sentence Web corpus with a precision of about 66% for *action-effect* relations.

Clearly, there is still much room for exploration and improvement. First of all, more comprehensive evaluations need to be done. For example, the acquired relations should be evaluated in terms of recall and usefulness. A deep error analysis is also needed. Second, the experiments have revealed that one major problem to challenge is how to optimize argument selection. We are seeking a way to incorporate a probabilistic model of predicate-argument co-occurrences into the ranking function for relation instances. Related to this issue, it is also crucial to devise a method for controlling argument sharing patterns. One possible approach is to employ state-of-the-art techniques for coreference and zero-anaphora resolution [8, 11] in preprocessing co-occurrence samples.

Chapter 4

Two-phase Method

1. Introduction

The growing interest in practical Natural Language Processing (NLP) applications such as Question Answering, Information Extraction and Multi-Document Summarization has greatly increased for identification of relations between textual fragments such as entailment and causal relations. Such applications often need to rely on a large amount of lexical semantic knowledge. For example, a causal (and entailment) relation holds between the verb phrases *wash something* and *something is clean*, which reflects the commonsense notion that if someone has washed something, this object is clean as a result of the washing event. A crucial issue is how to obtain and maintain a potentially huge collection of event relation instances. This thesis addresses the problem of how to automatically acquire such instances of relations between events (henceforth, *event relation instances*) from a large-scale text collection.

Motivated by this problem, several research groups have reported on experiment on automatic acquisition of causal, temporal and entailment relations between event mentions (typically verbs or verb phrases) [14, 9, 4, 28, 22, 29, 2]. As we explain below, however, none of these studies fully achieves the goal we pursue in this thesis.

An important aspect to consider in event relation acquisition is that each event has arguments. For example, the causal relation between *wash something* and *something is clean* can be represented naturally as:

$$\textit{wash}(\mathrm{obj}{:}X) \rightarrow_{cause} \textit{is\_clean}(\mathrm{subj}{:}X)$$

where X is a logical variable denoting that the filler of the object slot of the *wash* event should be shared (i.e. identical) with the filler of the subject slot of the *is_clean* event.

To be more general, an instance of a given relation R can be represented as:

$$predicate_1(\mathrm{arg}_1{:}X) \rightarrow_R predicate_2(\mathrm{arg}_2{:}X)$$

where $predicate_i$ is a natural language predicate, typically a verb or adjective, and X is a logical variable denoting which argument of one predicate and which argument of the other are shared.

The goal we pursue in this method is therefore not only (a) to find predicate pairs that are of a given relation type, but also (b) to identify the arguments shared between the predicates if any. We call the former subtask *predicate pair acquisition* and the latter *shared argument identification*.

However, a pattern-based method has a problem that is difficult to identify a shared argument. The problem is caused by using a co-occurrence pattern. The co-occurrence pattern represents words of a sentence. Two event phrases co-occurring with the co-occurrence pattern are in same sentence. In addition, it is rare that same phrases appear two times more than in a sentence. For example, we say, "He grills the meat so that it gets brown," however we say rarely, "He grills the meat so that it gets brown *the meat*." For those reasons, it is difficult to identify a shared argument by a pattern-based method.

In this chapter, we propose *Two-phrase method* for the problem of shared argument identification.

1.1 Existing Methods

Two-phrase method is inspired by some existing approaches. We show relation between two-phrase method and some existing approaches.

Existing methods for event relation acquisition can be classified into two approaches, which we call the *pattern-based approach* and *anchor-based approach* in this thesis.

The common idea behind the pattern-based approach is to use a small number of manually selected generic lexico-syntactic co-occurrence patterns (LSPs or simply patterns). Perhaps the simplest way of using LSPs for event relation acquisition can be seen in the method Chklovski and Pantel [4] employ to develop their knowledge resource called *VerbOcean*. Their method uses a small number of manually selected generic LSPs such as *to* ⟨Verb-X⟩ *and then* ⟨Verb-Y⟩ to obtain six types of semantic relations including *strength* (e.g. *taint – poison*) and *happens-before* (e.g. *marry – divorce*). The use of such generic patterns, however, tends to be high recall but low precision. Chklovski and Pantel [4], for example, report that their method obtains about 29,000 verb pairs with 65.5% precision.

The anchor-based approach, on the other hand, has emerged mainly in the context of paraphrase and entailment acquisition. This approach uses information of argument fillers (i.e. anchors) of each event expression as a useful clue for identifying event relations. A popular way of using such argument information relies on the distributional hypothesis [6] and identifies synonymous event expressions by seeking a set of event expressions whose argument fillers have a similar distribution. Such algorithms as DIRT [14] and TE/ASE [27] represent this line of research.

Another way of using argument information is proposed by Pekar [22], which identifies candidate verb pairs for the entailment relation by imposing criteria: (a) the two verbs must appear in the same local discourse-related context and (b) their arguments need to refer to the same participant, i.e. anchor. For example, if a pair of clauses *Mary bought a house.* and *The house belongs to Mary.* appear in a single local discourse-related context, two pairs of verbs, *buy*(obj:*X*) – *belong*(subj:*X*) and *buy*(subj:*X*) – *belong*(to:*X*) are identified as candidate entailment pairs.

It is by now clear that the above two approaches, which apparently have emerged somewhat independently, could play a complementary role with each other. Pattern-based methods, on the one hand, are designed to be capable of discriminating relatively fine-grained relation types. For example, the patterns used by Chklovski and Pantel [4] identify six relation types, while Abe et al. [2] identify two of the four causal relation types defined by Inui et al. [9]. However, these methods are severely limited for the purpose of shared argument identification because lexico-syntactic patterns are not a good indication of argument-shared structure in general. The anchor-based approach, on the other hand, works well for identifying shared arguments simply because it relies on argument information in identifying synonymous or entailment verb pairs. However, it has no direct means to discriminate more fine-grained specific relations such as causality and backward presupposition. To sum up, the pattern-based approach tends to be rather *relation-oriented* while the anchor-based approach tends to be *argument-oriented*.

Table 4.1. Difference of event relation acquisition approaches

Type of method	Research	Type of relation	Shared argument identification
Pattern-based method	[9, 4, 2]	Various relations	Hard
Anchor-based method	[14, 27, 22]	Synonym or entailment only	Easy
Combination method	[28]	Inference rule only	Easy
Two phrase method	This paper	Various relations	Easy

In spite of this complementarity, however, to our best knowledge, the issue of how to benefit from both approaches has never been paid enough attention. An interesting exception could be found in Torisawa [28]'s method of combining verb pairs extracted with a highly generic connective pattern ⟨*Verb-X*⟩ *and* ⟨*Verb-Y*⟩ together with the co-occurrence statistics between verbs and their arguments. While the reported results for inference rules with temporal ordering look promising, it is not clear yet, however, whether the method applies to other types of relations because it relies on relation-specific heuristics.

2. Contribution

The goal we pursue in this thesis is therefore not only (a) to find predicate pairs that are of a given relation type, but also (b) to identify the arguments shared between the predicates if any. We call the former subtask *predicate pair acquisition* and the latter *shared argument identification*. However, existing state-of-the-art methods for event relation acquisition are designed to achieve only either of these two subtasks but *not both*. We propose *Two-phase method*, which first uses lexico-syntactic patterns to acquire predicate pairs for a given relation type and then uses two kinds of anchors to identify shared arguments. Table 4.1 shows a position of our method.

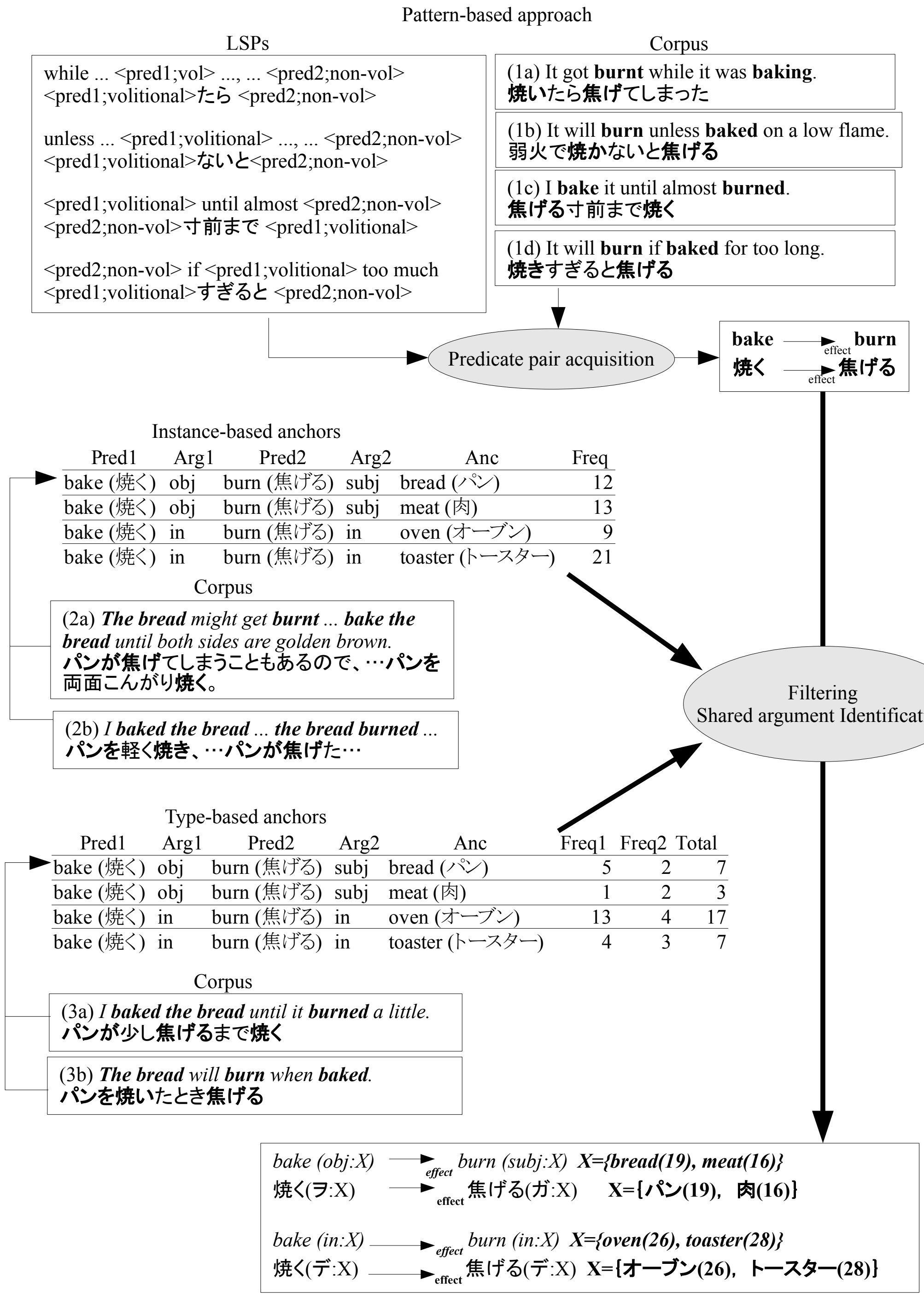

Figure 4.1. Two-phase method

3. Method

3.1 The Basic Idea

The complementarity between the pattern-based relation-oriented approach and the anchor-based argument-oriented approach as discussed above naturally leads us to consider combining them. The method we explore in this thesis is illustrated in Figure 4.1.

The overall process has two phrases: predicate pair acquisition followed by shared argument identification. Given a relation type for acquisition, we first acquire candidate predicate pairs that are likely to be of the given relation exploiting a state-of-the-art pattern-based method.

We then, in the second phase, seek anchors indicative of the shared argument for each acquired predicate pair. We consider two kinds of anchors: instance-based anchors and type-based anchors. If anchors are found, the predicate pair is verified and the associated argument pair is identified as the shared argument; otherwise, the predicate pair is discarded.

As we demonstrate in the section for empirical evaluation, this verification process boosts the accuracy as well as identifying shared arguments.

3.2 Predicate Pair Acquisition

For predicate pair acquisition, we can choose one from a range of state-of-the-art pattern-based methods. Among others, in our experiments, we adopted Abe et al. [2]'s method because it had an advantage in that it was capable of learning patterns as well as relation instances.

Abe et al. [2]'s method is based on Pantel and Pennacchiotti [20]'s *Espresso* algorithm, which is originally designed to acquire relations between entities. Espresso takes as input a small number of seed instances of a given target relation and iteratively learns co-occurrence patterns and relation instances in a bootstrapping manner. Abe et al. have made several extensions to it so that it can be applied to event relations. Since the details of this phase are not the focus of this thesis, we refer the reader to [2] for further information.

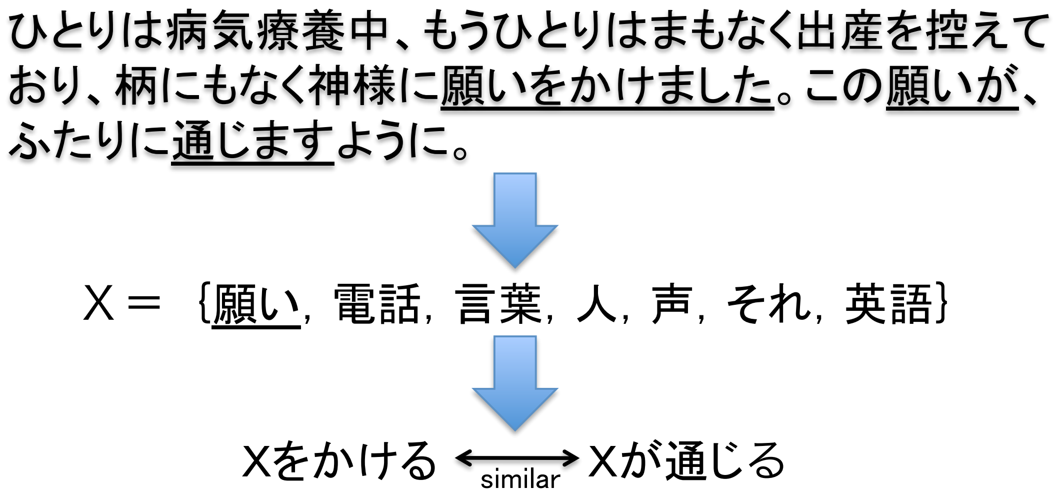

Figure 4.2. An example of instance based anchor

3.3 Shared Argument Identification

For each of the predicate pairs acquired in the previous phase, in shared argument identification, we use anchors to identify which argument is shared between the predicate pair. To find anchors indicative of shared arguments, we have so far examined two methods. We detail each below.

3.4 Instance-based Anchors

Inspired by Pekar [22]'s way of using anchors for verb entailment acquisition, we assume that if two related predicates have a shared argument, they must tend to appear in the same local discourse-related context with the shared argument filled with the same noun phrase (i.e. anchor). As an example, let us consider discourse (2a) in Figure 4.1. In this local discourse context, the noun *bread* appears twice, and one *bread* fills the subject slot of *burn* while the other fills the object slot of *bake*. In such a case, we assume the two *bread*s refer to the same object, namely anchor, and the subject of *burn* and the object of *bake* are shared with each other. We call such anchors *instance-based anchors* for the sake of contrast with *type-based anchors*, which we describe in Section 3.5.

Method

We implement this assumption in the following way. Given a pair of predicates $Pred_1$ and $Pred_2$, we search a corpus for tuples $\langle Pred_1\text{-}Arg_1; Pred_2, Arg_2; Anc\rangle$ satisfying the following conditions:

(a) Anchor word *Anc* is the head of a noun phrase filling argument Arg_1 of $Pred_1$ appearing in a Web page.

(b) *Anc* also fills argument Arg_2 of $Pred_2$ appearing in the same Web page as above.

(c) *Anc* must not be any of those in the stop list.

(d) $\mathrm{pmi}(Pred_i, Arg_i) \geq -1.0$ for $i \in \{1, 2\}$

For our experiments, we manually created the stop list, which contained 219 words including pronouns, numerals and highly generic nouns such as "こと (thing)", "もの (thing)" and "とき (time)". $\mathrm{pmi}(Pred_i, Arg_i)$ in condition (d) is the point-wise mutual information between $Pred_i$ and Arg_i. This condition is imposed for pruning wrong anchors misidentified due to parsing errors.

Difference of Pekar

While Pekar carefully defines boundaries of local discourse-related context, we simply assume that every pair of predicates sharing an anchor in a Web page is somewhat related — unlike Pekar, we do not impose such constraints as paragraph boundaries. Nevertheless, as we show later in the evaluation section, our assumption works precisely enough because the looseness of our discourse boundary constraint is compensated by the constraints imposed by lexico-syntactic patterns.

We finally calculate an anchor set for each argument pair $Pred_1\text{-}Arg_1$ and $Pred_2\text{-}Arg_2$ by accumulating the obtained tuples:

$$\begin{aligned}&AnchorSet(Pred_1\text{-}Arg_1, Pred_2\text{-}Arg_2)\\&= \{Arg | \langle Pred_1\text{-}Arg_1; Pred_2\text{-}Arg_2; Anc\rangle\}.\end{aligned}$$

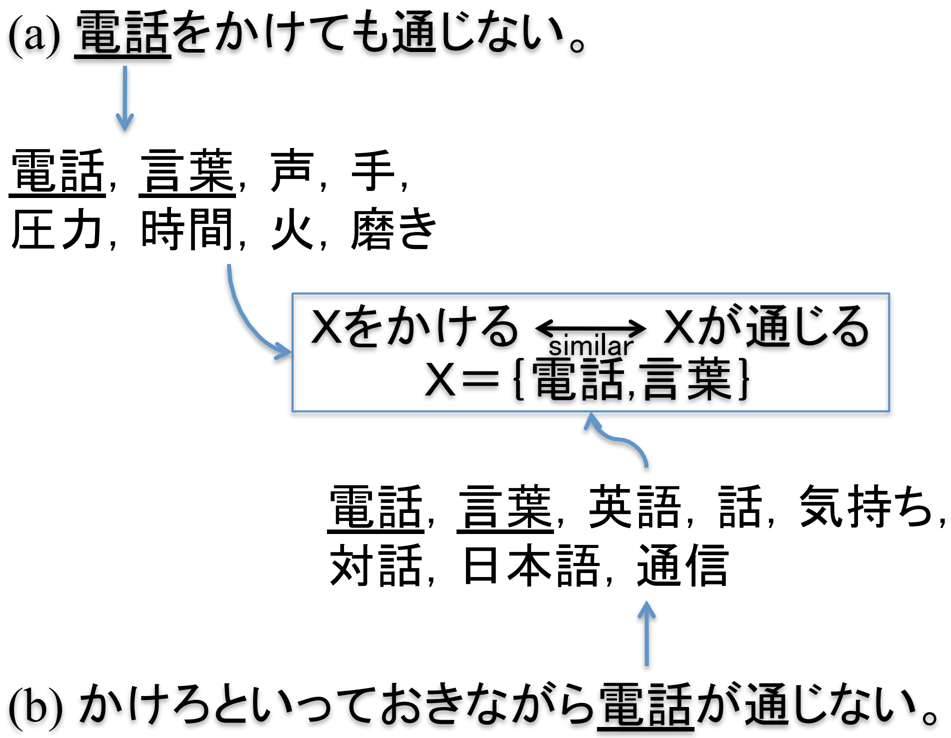

Figure 4.3. An example of type based anchor

3.5 Type-based Anchors

Let us consider sentences (3a) and (3b) in Figure 4.1. These two sentences both contain predicates *bake* and *burn*. In (3a), the noun *bread* fills the object slot of *bake*, while in (3b) the same noun *bread* fills the subject slot of *burn*. In such a case, we assume the noun *bread* to be an anchor indicating that the object of *bake* and the subject of *burn* are shared with each other. We call such anchors *type-based anchors* because *bread* in (3a) and *bread* in (3b) do not refer to the same object but are identical just as type.

Method

Given a pair of predicates $Pred_1$ and $Pred_2$, we search a corpus for sentences where $Pred_1$ and $Pred_2$ co-occur, and calculate the frequency counts of their argument fillers appearing in those sentences:

- If argument Arg_1 of $Pred_1$ is filled by noun Anc, increment the count of $\langle Pred_1\text{-}Arg_1; Pred_2; Anc\rangle$.
- If argument Arg_2 of $Pred_2$ is filled by noun Anc, increment the count of $\langle Pred_1; Pred_2\text{-}Arg_2; Anc\rangle$.

We then identify the intersection between the filler sets of $Pred_1$-Arg_1 and $Pred_2$-Arg_2 as the anchor set of that argument pair. Namely,

$$AnchSet(Pred_1\text{-}Arg_1, Pred_2\text{-}Arg_2) = S_1 \cap S_2,$$

where

$$S_1 = \{Arg | \langle Pred_1\text{-}Arg_1; Pred_2; Anc\rangle\},$$

$$S_2 = \{Arg | \langle Pred_1; Pred_2\text{-}Arg_2; Anc\rangle\}.$$

3.6 Application of Anchor Sets

We say an argument pair *covered by anchors* only if any anchor is found for it. Analogously, we say a predicate pair *covered by anchors* only if any argument pair associated with it is covered by anchors. In the phase of shared argument identification, for each given predicate pair, we carry out the following procedure:

1. Discard the predicate pair if it is not covered by anchors.
2. Choose maximally k-most frequent argument pairs associated with the predicate pair ($k = 3$ in our experiments).
3. Choose maximally l-most frequent anchors for each chosen argument pair ($l = 3$).

4. Experiments

4.1 Settings

For an empirical evaluation, we used a sample of approximately 500M sentences taken from the Web corpus collected by Kawahara and Kurohashi [10]. The sentences were part-of-speed-parsed with ChaSen [1], and were dependency-parsed with CaboCha [12]. Before using the sentences for experiments, we removed some sentences that satisfy one of the following conditions.

- A number of bunsetsu chunks of a sentence is one or more than thirteen.
- A number of characters of a bunsetsu chunk of a sentence is more than thirty three.
- A sentence has any symbols without punctuations.
- A sentence has unknown words.
- A sentence does not include more than two event mentions.

In addition, we removed punctuations from sentences. The sentences were extracted co-occurrence samples of event mentions. Event mentions with patterns whose frequency was less than 20 were discarded in order to reduce computational costs.

In our experiments, we considered two of Inui et al. [9]'s four types of causal relations: *action-effect* relations (Effect in Inui et al.'s terminology) and *action-means* relations (Means).

Action-effect

An *action-effect* relation holds between events x and y if and only if non-volitional event y is likely to happen as either a direct or indirect effect of volitional action x. For example, the action *X-ga undou-suru* (X exercises) and the event *X-ga ase-o-kaku* (X sweats) are considered to be in this type of relation. We did not require the necessity for an effect.

For example, while *nomu* (drink) does not necessarily result in *futsukayoi-ni naru* (have a hangover), the assessors judged this pair correct because one can at least say that the latter *sometimes* happens *as a result of* the former.

Action-means

An *action-means* relation, on the other hand, holds between events x and y if and only if volitional action y is likely to be done as a part/means of volitional action x.

For example, if case a event-pair is *X-ga hashiru* (X runs) is considered as a typical action that is often done as a part of the action *X-ga undou-suru* (X exercises).

Table 4.2. Examples of Two phase method

	Pred1	Arg1	Pred2	Arg2	Anc
action-effect	begin(開始する)	obj(ヲ)	finish(終了する)	subj(ガ)	installation(インストール), transaction(トランザクション)
action-effect	design(デザインする)	obj(ヲ)	be pretty(かわいい)	subj(ガ)	logotype(ロゴ)
action-effect	sleep(寝る)	in(デ)	be sleep(眠れる)	in(デ)	bed(ベット), futon(布団)
action-means	cure(治療する)	by(デ)	prescribe(処方する)	obj(ヲ)	medicine(薬)
action-means	cure(治療する)	obj(ヲ)	prescribe(処方する)	for(ニ)	patient(患者)
action-means	go home(帰宅する)	by(デ)	drive(運転する)	obj(デ)	car(車), car(自動車)
action-means	use(利用する)	obj(ヲ)	copy(コピーする)	obj(ヲ)	file(ファイル), data(データ)

Volitional Label

For our experiments, we manually built a lexicon of over 12,000 verbs with volitionality labels, obtaining 8,968 volitional verbs, 3,597 non-volitional and 547 ambiguous. Volitional verbs include *taberu* (eat) and *kenkyu-suru* (research), while non-volitional verbs include *atatamaru* (get warm), *kowareru* (to break-*vi*) and *kanashimu* (be sad). Volitionality information was used as a feature of predicate slots in pattern-based predicate pair acquisition.

5. Results and Discussion

5.1 Predicate Pair Acquisition

We ran the extended Espresso algorithm starting with 25 positive and 4 negative seed relation instances for the *action-effect* relation and 174 positive and 131 negative seed relations for the *action-means* relation. As a result, we obtained 9,511 patterns with 22,489 relation instances for *action-effect* and 14,119 co-occurrence patterns with 13,121 relation instances for *action-means* after 40 iterations of pattern and instance ranking/selection. The threshold parameters for selecting patterns and instances were decided in a preliminary trial. Some of the acquired instances are shown in Table 3.2.

We next randomly sampled 100 predicate pairs from each of four sections (1–500, 501–1500, 1501–3500 and 3501–7500) of the ranks of the acquired pairs for each relation class. Two annotators were asked to judge the correctness of each predicate pair (i.e. 800 pairs in total). They judged a predicate pair to be correct if they could produce an appropriate relation instance from that pair by adding some shared argument. For example, the pair かける (*hang/put/call*) and つながる (*connect*) was judged correct because it could constitute such a relation instance as:

(5) かける (を:X) $\rightarrow_{effect}$ つながる (が:X)
($X \in$ { 電話 })
make(obj:X) $\rightarrow_{effect}$ *go-through*(subj:X)
($X \in$ {*phone-call*})

Unfortunately, the two annotators did not agree with each other very much. out of the 400 samples, they agreed only on 294 for *action-effect* and 297 for *action-means*. However, a closer look at the results revealed that the judgements of the one annotator were considerably but very consistently more tolerant than the other. Assuming that the judgements of the latter correct, the precision and recall of those of the former would be 0.71 and 0.97 for *action-effect*, and 0.75 and 0.99 for *action-means*. These figures indicate that the two annotators agreed quite well with respect to the “goodness” of a sample, while having different criteria for strictness. For our evaluation, we decided to lean to the strict side and considered a sample correct only if it was judged correct by both annotators. The accuracy and recall achieved by the pattern-based model is shown in the column “all” under “LSPs” in Table 4.3.

Table 4.3. Accuracy and recall of relation classification

	LSPs		covered by anchors		
	all	top-N	instance	type	combined
action-effect	400	254	175	169	254
	269	185	144	143	206
(accuracy)	(0.67)	(0.72)	(0.82)	(0.84)	(0.81)
(recall)	(1.00)	(0.68)	(0.53)	(0.53)	(0.76)
action-means	400	254	178	176	254
	280	193	143	140	200
(accuracy)	(0.70)	(0.75)	(0.80)	(0.79)	(0.78)
(recall)	(1.00)	(0.68)	(0.51)	(0.50)	(0.71)

We then applied the anchor-based methods described in Section 3.3 to the above 800 sampled predicate pairs. The results are shown in the column "covered by anchors" of Table 4.3. Since the tendency for both relation classes is more or less the same, let us focus only on the results for *action-effect*.

Discussion

As shown in the column "all" under "LSPs" in the table, the pattern-based method covered 269 out of the 400 predicate pairs sampled above. The instance-based anchors ("instance") covered 175 out of the 400 predicate pairs sampled above, and 144 of them were correct with respect to relation type. We calculate its accuracy by dividing 144 by 175 and recall by dividing 144 by 269. These figures indicate that the instance-based anchors chose correct predicate pairs at a very high accuracy while sacrificing recall. The recall, however, can be extensively improved without losing accuracy by combining the instance-based and type-based anchors, where we considered a predicate pair *covered* if it was covered by either of the instance-based and type-based anchors. The results are shown in the column "combined" under "covered by anchors" in the same table. While the type-based anchors exhibited the same tendency as the instance-based anchors (namely, high accuracy and low recall), their coverage reasonably differed from each other, which contributed to the improvement of recall.

To summarize so far, the pattern-based method we adopted in the experiment gen-

Table 4.4. Accuracy of shared argument identification

		action-effect			action-means		
		anc-strict	anc-lenient	anc-any	anc-strict	anc-lenient	anc-any
arg-strict	instance	0.64	0.71	0.71	0.61	0.66	0.66
	type	0.60	0.63	0.65	0.61	0.65	0.67
	combined	0.60	0.65	0.66	0.58	0.62	0.64
arg-lenient	instance	0.78	0.80	0.80	0.73	0.75	0.76
	type	0.68	0.71	0.72	0.67	0.69	0.71
	combined	0.74	0.76	0.77	0.71	0.73	0.74

erated a substantial number of predicate pairs with a accuracy comparative to the state of the art. The accuracy was, however, further boosted by applying both instance-based and type-based anchors. This effect is particularly important because, to our best knowledge, very few pattern-based relation acquisition models have been reported to achieve as high a accuracy as what we achieved. In the case of our pattern-based model, for reference, the 254 highly ranked pairs of the 400 samples included only 185 correct pairs, which is worse than the 206 pairs covered by anchors for both accuracy and recall (see the "top-N" column under "LSPs" in Table 4.3. This difference also leads us to consider incorporating our anchor-based filtering into the boot-strapping cycles of pattern-based predicate pair acquisition.

5.2 Shared Argument Identification

We next investigated the accuracy of shared argument identification. For each of the aforementioned predicate pairs covered by anchors (the 254 pairs for *action-effect* and 254 for *action-means*), we asked the same two annotators as above to judge the correctness of the shared argument information. The results of combination are shown in Table 4.4.

"arg-strict" shows the results of the strict judgments where the shared argument was considered to be correctly identified only when the most frequent argument pair was judged correct, while "arg-lenient" shows the results of the lenient judgments where the shared argument was considered to be correctly identified when either of the three most frequent argument pairs was judged correct. For judging the correctness of an argument pair, we had three degrees of strictness. In the most strict criterion ("anc-strict"), an argument pair was judged correct only when its maximally three anchor words were all correct, while in "anc-lenient", an argument pair was judged correct when any of the three most frequent anchor words was correct. In "anc-any", an argument pair was judged correct as far as an annotator could think of any appropriate anchor word for it. While the inter-annotator agreement was not very high, with the kappa coefficient in the "arg-strict" and "anc-any" setting 0.47 for *action-effect* and 0.42 for *action-effect*), one was again consistently more tolerant than the other. For the same reason as argued in 4.2.1, we considered an acquired relation correct only if both annotators judged it correct.

In this experiment, predicate pairs that had been judged wrong with respect to relation types were all considered wrong in all the settings. The upper bounds of accuracy, therefore, are given by those in Table 4.3. For "arg-$*$" with the "combined" anchors, for example, the upper bound of accuracy is 0.81. Since "arg-lenient" with "combined" and "anc-lenient" achieved 0.76 accuracy, our method turned out to be reasonably precise in identifying argument pairs and their fillers. Paying attention to "arg-strict" and "anc-strict", on the other hand, one can see a considerable drop from the lenient case, which needs to be further investigated.

5.3 Error Analysis

We show typical errors of the results of the combined system.

(1) Incorrect arguments

- **(1a)** Since incorrect argument usage in corpus, our system identifies incorrect argument.
- **(2b)** Since error of dependency-parsed sentences, our system identifies incorrect argument.

(2) Incorrect shared arguments

- **(2a)** Expressions of arguments of two verbs are same, however, the arguments refer to different objects. From the arguments, our system acquires incorrect instance-based anchors, so that the anchors create incorrect shared arguments.

(2b) Expressions of arguments of two verbs are same, however, the arguments refer to different meaning. From the arguments, our system acquires incorrect type-based anchors, so that the anchors create incorrect shared arguments. The error often occurs by ambiguity noun.

We can discard lower frequency co-occurrence examples to solve problems of case (1a). In a similar way, to solve problems of case (1b), we can restrict strongly the threshold described in Section 3.4. However, the both solutions sacrifice recall. If we employ larger corpus or discard incorrect examples using another language model, the problem of sacrificing recall is not importance. However, the solutions (large corpus or language model) cannot solve the problems of case (2a) and (2b). Solutions of the problems require reference resource approach. We show examples of case (2a) and (2b).

X が操作する $\rightarrow_{action-effect}$ X が動く （X = プレイヤー）
X を雇う $\rightarrow_{action-effect}$ X に雇われる （X = 他人）
X から到着する $\rightarrow_{action-means}$ X から旅する （X = 空港）

6. Conclusion and Future Work

Motivated by the complementarity between the pattern-based relation-oriented approach and the anchor-based argument-oriented approach to event relation acquisition, we have explored a two-phased approach, which first uses patterns to acquire predicate pairs and then uses two types of anchors to identify shared arguments, reporting on the present results of our empirical evaluation. The results have shown that (a) the anchor-based filtering extensively improves the accuracy of predicate pair acquisition, (b) the instance-based and type-based anchors are almost equally contributive and combining them improves recall without losing accuracy, and (c) the anchor-based method also achieves high accuracy in shared argument identification.

Our future direction will be two-fold. One is evaluation. Clearly, more comprehensive evaluation needs to be done. For example, the acquired relation instances should be evaluated in some task-oriented manner. The other intriguing issue is how our anchor-based method for shared argument identification can benefit from recent advances in coreference and zero-anaphora resolution [8, 11].

Chapter 5

Conclusion

1. Contribution

The goal we pursue in this thesis was to acquire event relation instances. We therefore proposed *Extended Espresso* and *Two-phase method*.

In Chapter 3, we proposed *Extended Espresso*. We have addressed the issue of how to learn lexico-syntactic patterns useful for acquiring event relation knowledge from a large corpus, and proposed several extensions to a state-of-the-art method originally designed for entity relation extraction, reporting on the present results of our empirical evaluation. The results have shown that (a) there are indeed specific co-occurrence patterns useful for event relation acquisition, (b) the use of co-occurrence samples involving verbal nouns has positive impacts on both recall and precision, and (c) over five thousand relation instances are acquired from the 500M-sentence Web corpus with a precision of about 66% for *action-effect* relations.

In Chapter 4, we proposed *Two-phase method*. Motivated by the complementarity between the pattern-based relation-oriented approach and the anchor-based argument-oriented approach to event relation acquisition, we have explored a two-phase approach, which first uses patterns to acquire predicate pairs and then uses two types of anchors to identify shared arguments, reporting on the present results of our empirical evaluation. The results have shown that (a) the anchor-based filtering extensively improves the accuracy of predicate pair acquisition, (b) the instance-based and type-based anchors are almost equally contributive and combining them improves recall without losing accuracy, and (c) the anchor-based method also achieves high accuracy in shared argument identification.

Extended Espresso and *Two-phase method* were new approaches of event relation acquisition. In this thesis, we proposed the new approaches, and we evaluated the approaches on 500M-sentence Web corpus.

2. Future Work

This thesis proposes methods of event relation acquisition for NLP applications. The methods therefore provide for NLP applications, however, the methods are not evaluated by NLP applications yet. Application-based evaluation such as following questions is future work.

- Is number of acquired instances enough?
- Is precision of acquired instances enough?
- Can proposal method acquire more various relation instances?

In this thesis, we assume that an event relation instance is represented by following equation.

$$\mathit{predicate}_1(\mathrm{arg}_1{:}X) \rightarrow_R \mathit{predicate}_2(\mathrm{arg}_2{:}X)$$

In Chapter 4, we acquire concrete entities filling X. However, for applications, sometimes it is better that X is filled by an abstract entity instead of a concrete entity. For example, we consider following event relation instance.

$$\mathit{wash}(\mathrm{obj}{:}X) \rightarrow_{cause} \mathit{is\_clean}(\mathrm{subj}{:}X)$$

For X, the method of Chapter 4 probably acquires such as "car" or "cup". The result is correct, however, for applications, it is better that X means "material" (it is hypernym of "car" and "cup"). An extension to generalize an entity filling X is future work.

For the representation of shared arguments, another desirable extension exists. The representation is simple, does not cover some of actuality examples. For example, (5.1) shares "someone", our method can acquire it, it is not importance. An important point of the example is that "telephone" appears twice and one "telephone" is verb. Our shared argument identification can only recognize noun, can not recognize verb. However, it is better that the method can recognize verb and noun, therefore, the method can identify “telephone” as a shared argument (or a shared word).

$$\textit{telephone someone} \longrightarrow_{\textit{synonym relation}} \textit{call someone by telephone} \tag{5.1}$$

(2) is also can not identified by our method because the method can not recognize a part of word sequence. A desirable shared argument (or shared word) of the example is “hay fever”. Second “hay fever” is a part of “a hey fever drug”.

$$\textit{treat a hay fever} \longrightarrow_{\textit{causal relation}} \textit{take a hay fever drug}$$

(2) is also can not identified by our method because the example does not have twice word. However, a causal relation holds between the verb phrases *play baseball* and *hit a ball*.

$$\textit{play baseball} \longrightarrow_{\textit{causal relation}} \textit{hit a ball}$$

An extension for those example is also future work.

Acknowledgements

I am grateful to Professor Yuji Matsumoto, Associate Professor Kentaro Inui, Assistant Professor Masashi Shimbo and Assistant Professor Masayuki Asahara for giving me many comments. I learned a lot not only about research but about daily life from them. I also thank the thesis committee including Professor Kiyohiro Shikano for giving me valuable comments.

I would like to thank all previous and current members of Prof. Matsumoto laboratory. They gave me a lot of useful and interesting knowledge about computer science.

References

[1] Chasen. http://chasen-legacy.sourceforge.jp/.

[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 3rd International Joint Conference on Natural Language Processing*, pages 497–504, 2008.

[3] Rahul Bhagat and Deepak Ravichandran. Large scale acquisition of paraphrases for learning surface patterns. In *Proceedings of ACL-08: HLT*, pages 674–682, Columbus, Ohio, June 2008. Association for Computational Linguistics.

[4] Timothy Chklovski and Patrick Pantel. Global path-based refinement of noisy graphs applied to verb semantics. In *Proceedings of Joint Conference on Natural Language Processing (IJCNLP-05)*, pages 792–803, 2005.

[5] Roxana Girju, Adriana Badulescu, and Dan Moldovan. Automatic discovery of part-whole relations. *Comput. Linguist.*, 32(1):83–135, 2006.

[6] Zelling Harris. Mathematical structures of language. *Interscience Tracts in Pure and Applied Mathematics*, 1968.

[7] M Hearst. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the Fourteenth International Conference on Computational Linguistics*, pages 539–545, July 1992.

[8] Ryu Iida, Kentaro Inui, and Yuji Matsumoto. Exploiting syntactic patterns as clues in zero-anaphora resolution. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the ACL*, pages 625–632, 2006.

[9] 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 *Proceedings of the 6th International Conference on Discovery Science*, pages 180–193, 2003. An extended version: Takashi Inui, Kentaro Inui, and Yuji Matsumoto (2005). Acquiring causal knowledge from text using the connective marker tame. *ACM Transactions on Asian Language Information Processing (TALIP), 4(4):435–474.*

[10] 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, 2006.

[11] Mamoru Komachi, Ryu Iida, Kentaro Inui, and Yuji Matsumoto. Learning based argument structure analysis of event-nouns in japanese. In *Proceedings of the Conference of the Pacific Association for Computational Linguistics (PACLING)*, pages 120–128, 2007.

[12] Taku Kudo and Yuji Matsumoto. Japanese dependency analysis using cascaded chunking. In *CoNLL 2002: Proceedings of the 6th Conference on Natural Language Learning 2002 (COLING 2002 Post-Conference Workshops)*, pages 63–69, 2002.

[13] Douglas B. Lenat. Cyc: a large-scale investment in knowledge infrastructure. *Commun. ACM*, 38(11):33–38, 1995.

[14] Dekang Lin and Patrick Pantel. Dirt: discovery of inference rules from text. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 323–328, 2001.

[15] Suguru Matsuyoshi, Satoshi Sato, and Takehito Utsuro. Compilation of a dictionary of japanese functional expressions with hierarchical organization. In *Proceedings of the 21st International Conference on Computer Processing of Oriental Languages*, pages 395–402, 2006.

[16] George A. Miller. Wordnet: A lexical database for english. *Communications of the ACM*, 38:39–41, 1995.

[17] Shachar Mirkin, Ido Dagan, and Maayan Geffet. Integrating pattern-based and distributional similarity methods for lexical entailment acquisition. In *Proceedings of the COLING/ACL on Main conference poster sessions*, pages 579–586, Morristown, NJ, USA, 2006. Association for Computational Linguistics.

[18] Tamara Munzner, Francois Guimbretiere, and George Robertson. Constellation: A visualization tool for linguistic queries from mindnet. In *In Proceedings of the 1999 IEEE Symposium on Information Visualization*, pages 132–135. Press, 1999.

[19] Bo Pang, Kevin Knight, and Daniel Marcu. Syntax-based alignment of multiple translations: extracting paraphrases and generating new sentences. In *NAACL '03: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 102–109, Morristown, NJ, USA, 2003. Association for Computational Linguistics.

[20] Patric Pantel and Marco Pennacchiotti. Espresso: Leveraging generic patterns for automatically harvesting semantic relations. In *the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL*, pages 113–120, 2006.

[21] Viktor Pekar. Acquisition of verb entailment from text. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 49–56, 2006.

[22] Viktor Pekar. Acquisition of verb entailment from text. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 49–56, 2006.

[23] Deepak Ravichandran and Eduard Hovy. Learning surface text patterns for a question answering system. In *Proceedings of the 21st International Conference on Computational Linguistics and 40th Annual Meeting of the Association for Computational Linguistics*, pages 41–47, 2002.

[24] E. Riloff and J Shepherd. A corpus-based approach for building semantic lexicons. In *Proceedings of the Second Conference on Empirical Methods in Natural Language Processing (EMNLP-2)*, 1997.

[25] J. Ruppenhofer, M. Ellsworth, M.R.L. Petruck, C.R. Johnson, and J. Scheffczyk. FrameNet II: Theory and Practice. online publication, 2006.

[26] Push Singh, Thomas Lin, Erik T. Mueller, Grace Lim, Travell Perkins, and Wan Li Zhu. Open mind common sense: Knowledge acquisition from the general public. In *On the Move to Meaningful Internet Systems, 2002 - DOA/CoopIS/ODBASE 2002 Confederated International Conferences DOA, CoopIS and ODBASE 2002*, pages 1223–1237, London, UK, 2002. Springer-Verlag.

[27] Idan Szpektor, Hristo Tanev, Ido Dagan, and Bonaventura Coppola. Scaling web-based acquisition of entailment relations. In Dekang Lin and Dekai Wu, editors, *Proceedings of EMNLP 2004*, pages 41–48, Barcelona, Spain, 2004. Association for Computational Linguistics.

[28] Kentaro Torisawa. Acquiring inference rules with temporal constraints by using japanese coordinated sentences and noun-verb co-occurrences. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 57–64, 2006.

[29] Fabio Massimo Zanzotto, Marco Pennacchiotti, and Maria Teresa Pazienza. Discovering asymmetric entailment relations between verbs using selectional preferences. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 849–856, 2006.

Appendix

A. Effects of Discarding Lower Frequencies

As we described Section 4.1 and Section 4.1, we discarded event mentions with patterns whose frequency was less than 20 in order to reduce computational costs. This section shows effects of the discarding.

Figure 5.1 and Figure 5.2 show a coverage of types of a pair between an event mention pair and a pattern co-occurring with the event mention pair when discarding a pair with its frequency (Figure 5.2 is a magnification part of lower frequency of Figure 5.1). For example, frequency 20 shows a discarding coverage 0.13 % that we discard a pair between an event mention pair and a pattern co-occurring with the event mention pair with frequency less than 20[1]. In Figure 5.1 and Figure 5.2, a discarding coverage of frequency two is 5.65 %, and a discarding coverage of frequency three is 2.21 %. The results show that a majority of types is in lower frequency. We discarded examples with frequency 20. The discarding coverage of 20 is 0.13 %. The discarding coverage is low because of a majority in lower frequency.

Figure 5.3 is a discarding coverage of tokens version of Figure 5.1. Figure 5.4 is also a discarding coverage of tokens version of Figure 5.2. A discarding coverage of frequency two is 21.69 %, and a discarding coverage of frequency three is 15.98 %. The results shows also that a majority of tokens is in lower frequency. We discarded examples with frequency 20. The discarding coverage of 20 is 7.10 %. The discarding coverage is also low because of a majority in lower frequency.

A coverage (not a discarding coverage) of tokens of frequency one is 78.31 %. It means that about 80 % of examples appears only once if we collect all examples

[1]A discarding coverage of frequency one always shows 100 % because of discarding a mention with frequency less than one ("less than one" equals "or less zero").

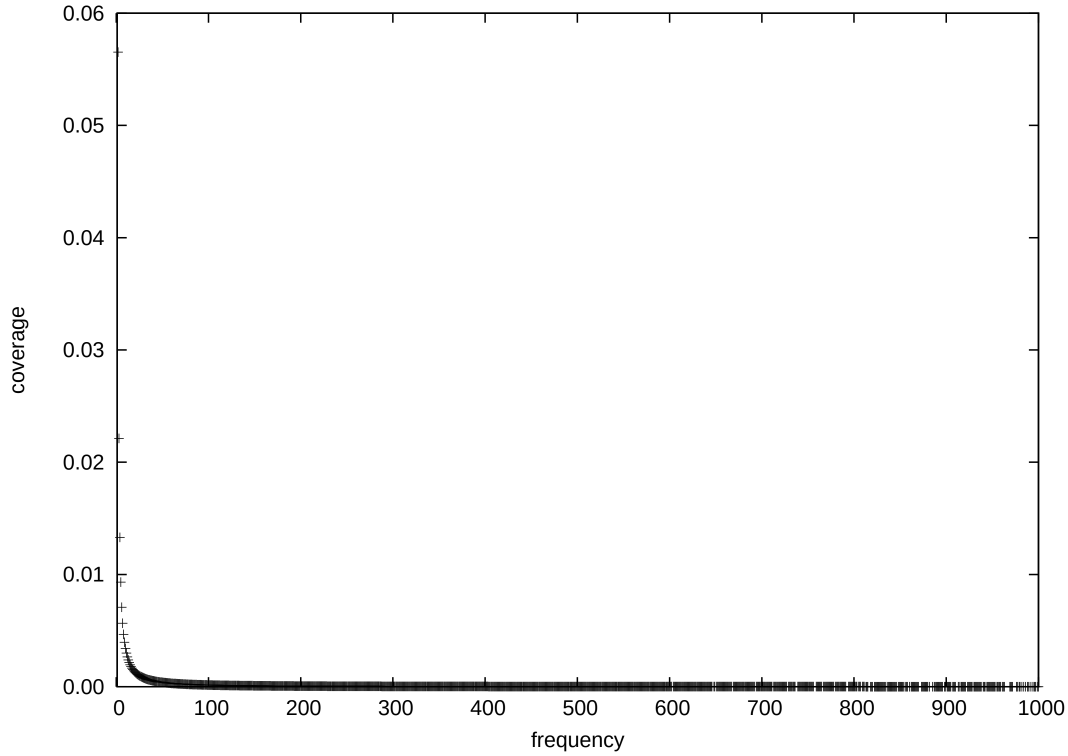

Figure 5.1. Effect of discarding: a coverage of types

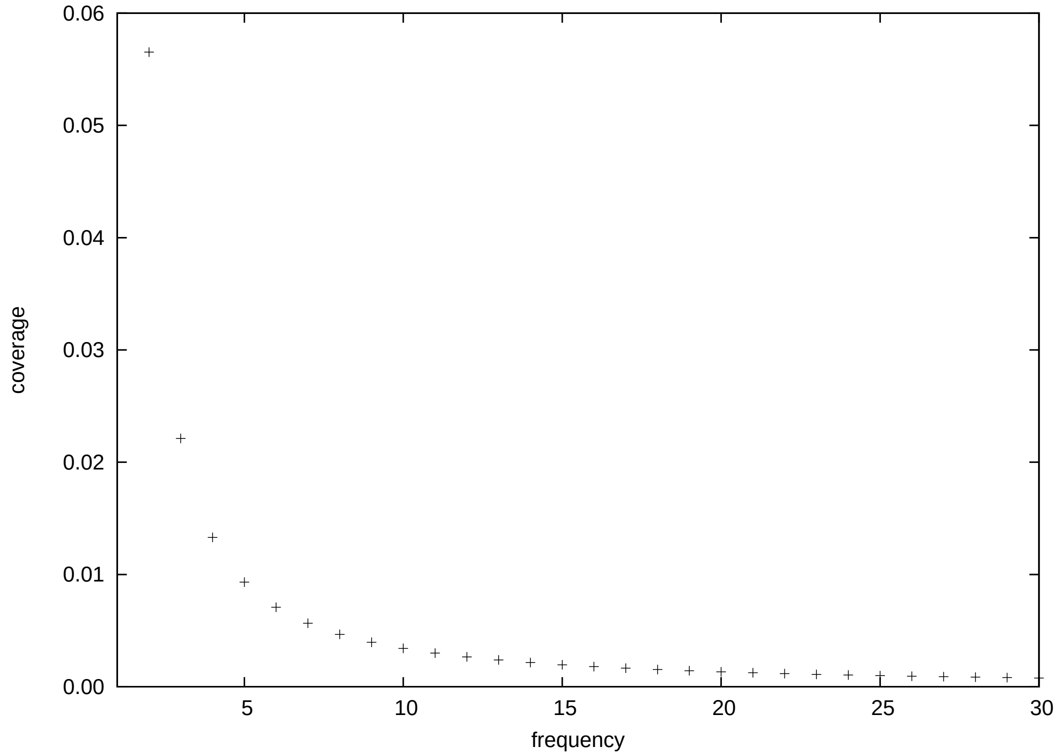

Figure 5.2. Effect of discarding: a coverage of types (lower frequency)

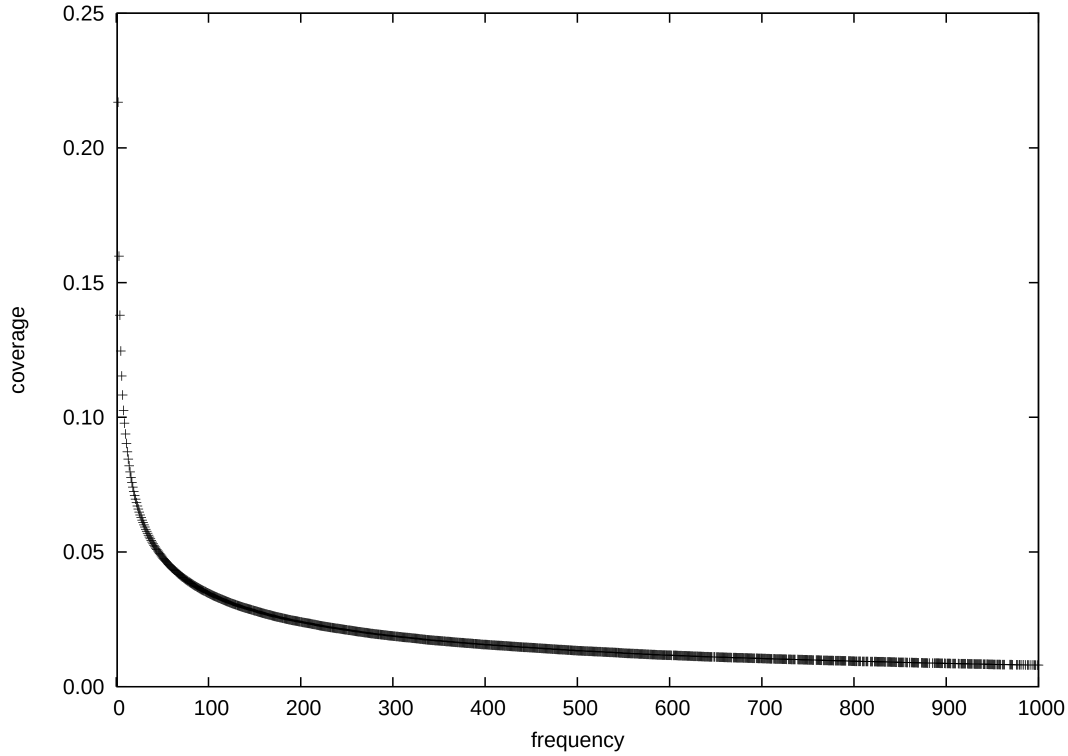

Figure 5.3. Effect of discarding: a coverage of tokens

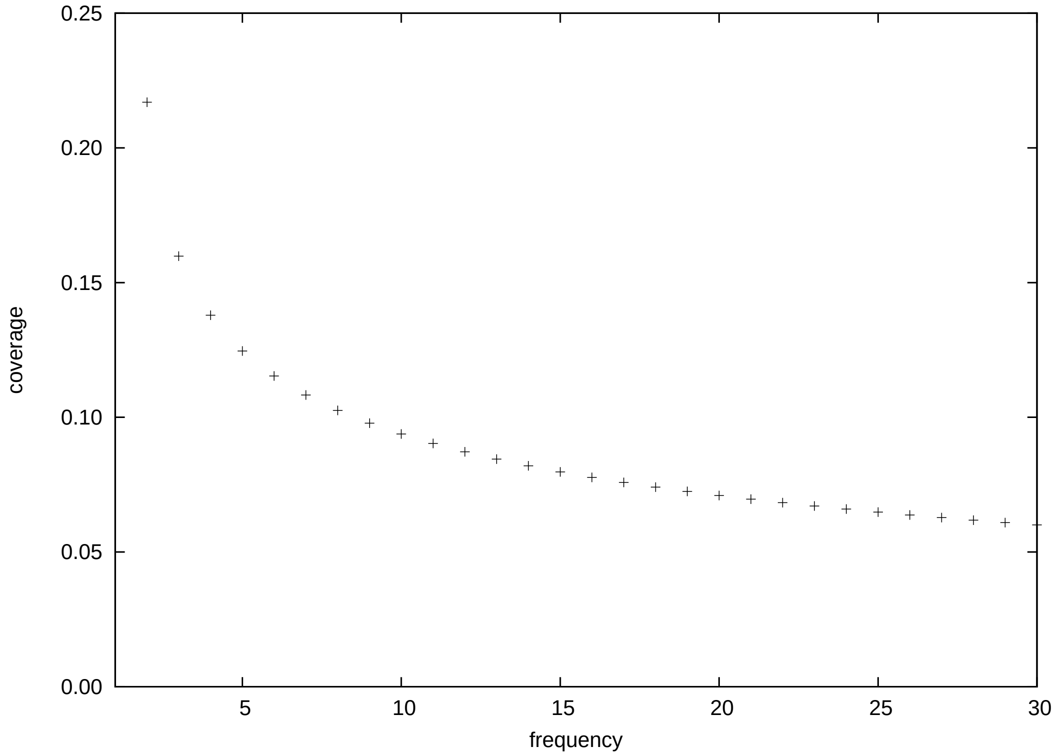

Figure 5.4. Effect of discarding: a coverage of tokens (lower frequency)

including examples of frequency one. It seems that examples of lower frequency is not useful for applications.

The results of a discarding coverage of types and tokens show that a majority of examples is in lower frequency. If we cover examples of lower frequency, we require more computational power.

B. List of Publications

B.1 Peer Review Journal Paper

- 阿部修也, 乾健太郎, 松本裕治，共起パターンの学習による事態間関係知識の獲得，自然言語処理, Vol.16, No.5, pp.79–100, October 2009.
- 阿部修也, 乾健太郎, 松本裕治，項の共有関係と統語パターンを用いた事態間関係獲得，自然言語処理, Vol.17, No.1, pp.121–139, January 2010.

B.2 Peer Review International Conference

- Kentaro Inui, Shuya Abe, Hiraku Morita, Megumi Eguchi, Kazuo Hara, Koji Murakami, Suguru Matsuyoshi, Asuka Sumida, and Chitose Sao, 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, pp314–321, Dec. 2008.
- 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), pp.1–8. Manchester, UK, August 2008.
- 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 3rd International Joint Conference on Natural Language Processing, pp.497–504. Hyderabad, India, January 2008.

B.3 Conference

- 阿部修也, 江口萌, 隅田飛鳥, 大崎梓, 乾健太郎，みんなの経験：ブログから抽出したイベントおよびセンチメントのＤＢ化，言語処理学会第 15 回年次大会発表論文集, pp.296–299, March 2009.
- 阿部修也, 乾健太郎, 松本裕治, 文内共起パターンと格要素共有情報による事態間関係知識の獲得，言語処理学会第 14 回年次大会論文集, pp.797–800, March 2008.

- 阿部修也, 乾健太郎, 松本裕治，事態含意名詞の利用と共起パターンの学習による事態間関係知識の獲得，言語処理学会第 13 回年次大会論文集, pp.883–886, March 2007.

- 青山桜子, 阿部修也, 大西良明, 乾健太郎, 松本裕治，事態間関係の獲得のための動詞語釈文の構造化，言語処理学会第 13 回年次大会論文集, pp.286–289, March 2007.

- 阿部修也, 乾健太郎, 松本裕治，論理関係に基づく複文間の言い換え含意関係の認識と生成，言語処理学会第 12 回年次大会論文集, pp.204–207, March 2006.

B.4 Workshop

- 阿部修也, 江口萌, 隅田飛鳥, 大崎梓, 乾健太郎，Web からの経験マイニング，人工知能学会 知識ベースシステム研究会 SIG-KBS-A803，pp.39–44，January 2009.

- 阿部修也，乾健太郎，松本裕治，2 種類のアンカー情報と共起パターンの組み合わせによる事態間関係獲得，情報処理学会研究報告, 自然言語処理・言語理解とコミュニケーション合同研究会, 信学技報 Vol.108 No.141, 2008-NL-186, pp.19–24, July 2008.

- 阿部修也, 乾健太郎, 松本裕治，事態含意名詞を用いた事態間関係知識の獲得，情報処理学会研究報告, 自然言語処理研究会, 2006-NL-176, pp.95–100, November 2006.

- 竹内孔一，乾健太郎，藤田篤，竹内奈央，阿部修也，分類の根拠を明示した動詞語彙概念構造の構築，自然言語処理研究会 2005-NL-169.

B.5 Symposium

- 阿部修也, 事態オントロジー構築のための知識獲得，『語彙資源の深化とＮＬＰ新時代』科研・合同シンポジウム, September 2006.

- 中野智子, 菅原昌平, 阿部修也, 乾健太郎，視覚障がい者の聞きやすさを考慮した難語の平易化，ヒューマンインタラクションシンポジウム, September 2006.