Two-Phased Event Relation Acquisition: Coupling the Relation-Oriented and Argument-Oriented Approaches

Shuya Abe **Kentaro Inui** **Yuji Matsumoto**
Graduate School of Information Science,
Nara Institute of Science and Technology
{shuya-a,inui,matsu}@is.naist.jp

Abstract

Addressing the task of acquiring semantic relations between events from a large corpus, we first argue the complementarity between the pattern-based relation-oriented approach and the anchor-based argument-oriented approach. We then propose a two-phased 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 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.

1 Introduction

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 paper addresses the issue of how to automatically acquire such instances of relations between events (henceforth, *event relation instances*) from a large-scale text collection.

Motivated by this issue, several research groups have reported their experiments on automatic acquisition of causal, temporal and entailment relations between event expressions (typically verbs or verb phrases) (Lin and Pantel, 2001; Inui et al., 2003; Chklovski and Pantel, 2005; Torisawa, 2006; Pekar, 2006; Zanzotto et al., 2006; Abe et al., 2008, etc.). As we explain below, however, none of these studies fully achieves the goal we pursue in this paper.

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:

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

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:

(2) $\textit{pred}_1(\text{arg}_1{:}X) \rightarrow_R \textit{pred}_2(\text{arg}_2{:}X)$,

where $\textit{pred}_i$ 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 paper 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*. As we review in the next section, 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 paper, 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.

2 Previous work

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 paper.

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 (2005) 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 (2005), 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 (Chklovski and Pantel, 2005; Torisawa, 2006; Zanzotto et al., 2006). Another way is to incorporate a classifier trained with supervision. Inui et al. (2003), 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. (2008) propose to extend Pantel and Pennacchiotti (2006)'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.

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 (Harris, 1968) and identifies synonymous event expressions by seeking a set of event expressions whose argument fillers have a similar distribution. Such algorithms as DIRT (Lin and Pantel, 2001) and TE/ASE (Szpektor et al., 2004) represent this line of research.

Another way of using argument information is proposed by Pekar (2006), 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 (2005) identify six relation types, while Abe et al. (2008) identify two of the four causal relation types defined by Inui et al. (2003). 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

[1] A ⟨⟩ included in an LSP denotes, throughout this paper, 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.

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*.

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 (2006)'s method of combining verb pairs extracted with a highly generic connective pattern $\langle$*Verb-X*$\rangle$ *and* $\langle$*Verb-Y*$\rangle$ 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 Two-phased event relation acquisition

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 paper is illustrated in Figure 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. (2008)'s method because it had an advantage in that it was capable of learning patterns as well as relation instances.

Abe et al. (2008)'s method is based on Pantel and Pennacchiotti (2006)'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 paper, we refer the reader to (Abe et al., 2008) for further information.

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.

3.3.1 Instance-based anchors

Inspired by Pekar (2006)'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 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 3.3.2.

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:

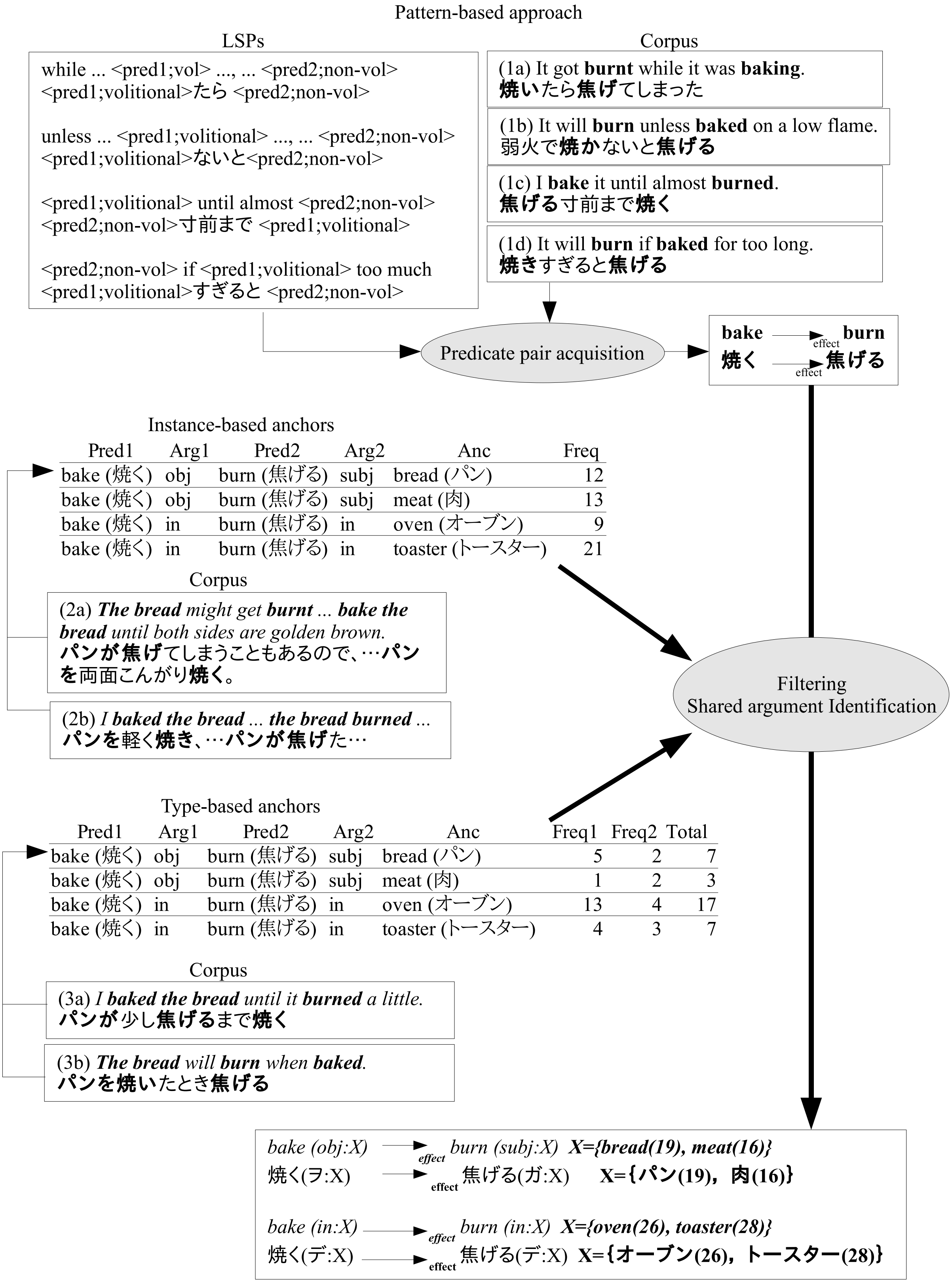
(a) Anchor word *Anc* is the head of a noun phrase filling argument Arg_1 of $Pred_1$ appearing in a Web page.

(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


Pattern-based approach
LSPs
Corpus
Predicate pair acquisition
Instance-based anchors
Type-based anchors
Corpus
Corpus
Filtering
Shared argument Identification
bake →effect **burn**
焼く →effect **焦げる**

while ... <pred1;vol> ..., ... <pred2;non-vol>
<pred1;volitional>たら <pred2;non-vol>

unless ... <pred1;volitional> ..., ... <pred2;non-vol>
<pred1;volitional>ないと<pred2;non-vol>

<pred1;volitional> until almost <pred2;non-vol>
<pred2;non-vol>寸前まで <pred1;volitional>

<pred2;non-vol> if <pred1;volitional> too much
<pred1;volitional>すぎると <pred2;non-vol>

(1a) It got **burnt** while it was **baking**.
焼いたら**焦げ**てしまった

(1b) It will **burn** unless **baked** on a low flame.
弱火で**焼か**ないと**焦げる**

(1c) I **bake** it until almost **burned**.
焦げる寸前まで**焼く**

(1d) It will **burn** if **baked** for too long.
焼きすぎると**焦げる**

Instance-based anchors

Pred1	Arg1	Pred2	Arg2	Anc	Freq
bake (焼く)	obj	burn (焦げる)	subj	bread (パン)	12
bake (焼く)	obj	burn (焦げる)	subj	meat (肉)	13
bake (焼く)	in	burn (焦げる)	in	oven (オーブン)	9
bake (焼く)	in	burn (焦げる)	in	toaster (トースター)	21

(2a) ***The bread*** *might get* ***burnt*** *...* ***bake the bread*** *until both sides are golden brown.*
パンが焦げてしまうこともあるので、…**パン**を両面こんがり**焼く**。

(2b) ***I baked the bread*** *...* ***the bread burned*** *...*
パンを軽く**焼き**、…**パンが焦げ**た…

Type-based anchors

Pred1	Arg1	Pred2	Arg2	Anc	Freq1	Freq2	Total
bake (焼く)	obj	burn (焦げる)	subj	bread (パン)	5	2	7
bake (焼く)	obj	burn (焦げる)	subj	meat (肉)	1	2	3
bake (焼く)	in	burn (焦げる)	in	oven (オーブン)	13	4	17
bake (焼く)	in	burn (焦げる)	in	toaster (トースター)	4	3	7

(3a) ***I baked the bread*** *until it* ***burned*** *a little.*
パンが少し**焦げる**まで**焼く**

(3b) ***The bread*** *will* ***burn*** *when* ***baked****.*
パンを焼いたとき**焦げる**

bake (obj:X) →*effect* *burn (subj:X)* ***X={bread(19), meat(16)}***
焼く(ヲ:X) →effect 焦げる(ガ:X) **X={パン(19)，肉(16)}**

bake (in:X) →*effect* *burn (in:X)* ***X={oven(26), toaster(28)}***
焼く(デ:X) →effect 焦げる(デ:X) **X={オーブン(26)，トースター(28)}**

Figure 1: Two-phased event relation acquisition

"こと (thing)", "もの (thing)" and "とき (time)". $\mathrm{pmi}(\mathit{Pred}_i, \mathit{Arg}_i)$ 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.

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-Arg_1 and Pred_2-Arg_2 by accumulating the obtained tuples:

$$\begin{aligned}&\mathit{AnchorSet}(\mathit{Pred}_1\text{-}\mathit{Arg}_1, \mathit{Pred}_2\text{-}\mathit{Arg}_2)\\ &= \{\mathit{Arg} | \langle \mathit{Pred}_1\text{-}\mathit{Arg}_1; \mathit{Pred}_2\text{-}\mathit{Arg}_2; \mathit{Anc}\rangle\}.\end{aligned}$$

3.3.2 Type-based anchors

Let us consider sentences (3a) and (3b) in Figure 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.

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 $\langle \mathit{Pred}_1\text{-}\mathit{Arg}_1; \mathit{Pred}_2; \mathit{Anc}\rangle$.
- If argument Arg_2 of Pred_2 is filled by noun *Anc*, increment the count of $\langle \mathit{Pred}_1; \mathit{Pred}_2\text{-}\mathit{Arg}_2; \mathit{Anc}\rangle$.

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,

$$\mathit{AnchSet}(\mathit{Pred}_1\text{-}\mathit{Arg}_1, \mathit{Pred}_2\text{-}\mathit{Arg}_2) = S_1 \cap S_2,$$

where

$$S_1 = \{\mathit{Arg} | \langle \mathit{Pred}_1\text{-}\mathit{Arg}_1; \mathit{Pred}_2; \mathit{Anc}\rangle\},$$

$$S_2 = \{\mathit{Arg} | \langle \mathit{Pred}_1; \mathit{Pred}_2\text{-}\mathit{Arg}_2; \mathit{Anc}\rangle\}.$$

3.3.3 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 (2006). The sentences were dependency-parsed with CaboCha (Kudo and Matsumoto, 2002), and co-occurrence samples of event mentions were extracted. 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. (2003)'s four types of causal relations: *action-effect* relations (Effect in Inui et al.'s terminology) and *action-means* relations (Means). 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. 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).

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.

4.2 Results and discussion

4.2.1 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 1.

We next randomly sampled 100 predicate pairs from each of four sections (1–500, 501–1500, 1501–3500 and 3500–7500) 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:

(3) かける (を:*X*) $\rightarrow_{effect}$ つながる (が:*X*)
(*X* ∈ { 電話 })
make(obj:*X*) $\rightarrow_{effect}$ *go-through*(subj:*X*)
(*X* ∈ {*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 2.

Table 2: 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 2. Since the tendency for both relation classes is more or less the same, let us focus only on the results for *action-effect*.

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

Table 1: Examples

	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)

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 generated 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 2. This difference also leads us to consider incorporating our anchor-based filtering into the bootstrapping cycles of pattern-based predicate pair acquisition.

4.2.2 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 3.

"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 2. 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 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 los-

Table 3: 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

ing 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 (Iida et al., 2006; Komachi et al., 2007, etc.).

References

Abe, Shuya, Kentaro Inui, and Yuji Matsumoto. 2008. 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.

Chklovski, Timothy and Patrick Pantel. 2005. Global path-based refinement of noisy graphs applied to verb semantics. In *Proceedings of Joint Conference on Natural Language Processing*.

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

Iida, Ryu, Kentaro Inui, and Yuji Matsumoto. 2006. 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. Association for Computational Linguistics.

Inui, Takashi, Kentaro Inui, and Yuji Matsumoto. 2003. 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.

Kawahara, Daisuke and Sadao Kurohashi. 2006. A fully-lexicalized probabilistic model for japanese syntactic and case structure analysis. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 176–183.

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

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

Lin, Dekang and Patrick Pantel. 2001. 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.

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

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

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

Torisawa, Kentaro. 2006. Acquiring inference rules with temporal constraints by using japanese coordinated sentences and noun-verb co-occurrences. In *Proceedings of Human Language Technology Conference/North American chapter of the Association for Computational Linguistics annual meeting (HLT-NAACL06)*, pages 57–64.

Zanzotto, Fabio Massimo, Marco Pennacchiotti, and Maria Teresa Pazienza. 2006. 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.