PAPER *Special Issue on Text Processing for Information Access*

Corpus Based Method of Transforming Nominalized Phrases into Clauses for Text Mining Application

Akira TERADA[†] ***and*** **Takenobu TOKUNAGA**[†], ***Nonmembers***

SUMMARY Nominalization is a linguistic phenomenon in which events usually described in terms of clauses are expressed in the form of noun phrases. Extracting event structures is an important task in text mining applications. To achieve this goal, clauses are parsed and the argument structure of main verbs are extracted from the parsed results. This kind of preprocessing has been commonly done in the past research. In order to extract event structure from nominalized phrases as well, we need to establish a technique to transform nominalized phrases into clauses. In this paper, we propose a method to transform nominalized phrases into clauses by using corpus-based approach. The proposed method first enumerates possible predicate/argument structures by referring to a nominalized phrase (noun phrase) and makes their ranking based on the frequency of each argument in the corpus. The algorithm based on this method was evaluated using a corpus consisting of 24,626 aviation safety reports in English and it achieved a 78% accuracy in transformation. The algorithm was also evaluated by applying a text mining application to extract events and their cause-effect relations from the texts. This application produced an improvement in the text mining application's performance.

key words: *nominalization, predicate/argument structure, text mining, corpus based method*

1. Introduction

The rapid increase in the number of electronic documents in recent years has made automatic document processing by computer indispensable. Text mining is an important document processing technology that finds implicit facts and relations in documents. It is considered to be an application of data mining techniques to written texts. However, text mining is significantly different from data mining in that it should deal with unstructured text, whereas data mining aims to reveal hidden facts and relations in data that are usually well structured, such as in databases. Therefore preprocessing to transform unstructured texts into well-structured data is a crucial aspect of text mining.

Text mining preprocessing must be able to deal with nominalized phrases. Nominalization is a linguistic phenomenon in which events usually described by clauses are expressed in the form of noun phrases. For example, a clause "The army destroyed the city" describes an event, which could be restated by noun phrases "The destruction of the city by the army", "The army's destruction of the city" and so on. When extracting event structures from texts, it is necessary to extract the same event structure from these different surface expressions.

To extract the event structure, clauses are parsed and the argument structure of a main verb is extracted from the parsed results. This kind of preprocessing has been done in past text mining research. To extract the event structure from nominalized phrases as well, we need to establish a technique to extract the structure from nominalized phrases or to transform nominalized phrases into clauses, and then apply the traditional preprocessing. In this paper, we take the latter approach and propose a method of transforming nominalized phrases into clauses by using a corpus-based approach.

Section 2 of this paper describes the characteristics of nominalization and reviews related work. Section 3 describes the proposed method to transform nominalized phrases into clauses. Section 4 describes the evaluation experiments we performed. We evaluated the proposed method in terms of the accuracy of the transformation of nominalized phrase into clauses and in terms of improvement of text mining performance. Section 5 discusses the applications of this method, and Sect. 6 summarizes the paper and looks at future work.

2. Nominalization

In general, a noun can be characterized as being "stative" such as stable, concrete, or abstract. In contrast, a verb is naturally characterized as being "dynamic" as it indicates actions, and changing conditions [12]. According to Quirk et al., a nominalized phrase is a noun phrase which has a systematic correspondence with a clause including a verb, where the head nouns of the nominalized phrase is related morphologically to the verb [12]. In English text, nominalization occurs frequently, and the interpretation of the nominalization is crucial for analyzing text.

A Nominalized phrase is different from a clause in many aspects [8]:

- Any verbal role may only be filled once (uniqueness constraint).
- The grammar of nominalization is less rigid than

Manuscript received November 30, 2002.
Manuscript revised February 4, 2003.

[†] The authors are with the Department of Computer Science, Tokyo Institute of Technology, Tokyo, 152-8552 Japan.

that of clauses.

- Arguments are optional and their order is not rigid.
- Subject and object are represented by a noun in the pre-nominal modifier. In particular, a genitive noun strongly suggests the subject role.
- In pre-nominal modifiers, subject precedes object (ordering constraint).

Macleod et al. compiled a dictionary of English nominalization, called NOMLEX [8]. NOMEX includes the corresponding verb arguments with their selectional constraints and oblique verbal complements. NOMLEX was compiled for automatic processing of nominalizations, and it has 1,015 entries defining the relations between clauses and corresponding nominalized phrases.

Consider the following example. (For simplicity, determiners are omitted.)

1. Aircraft penetrated space.

The possible nominalizations are shown as follows:

2. Aircraft space penetration.
3. Space penetration by aircraft.
4. Aircraft penetration.
5. Space penetration.
6. Aircraft's penetration of space.

However, nominalization below is not permitted due to the ordering constraint.

7. *Space aircraft penetration.

Methods to deal with nominalization can be classified into knowledge-based ones and corpus-based ones.

Dahl et al. [2] used the knowledge-based approach to analyze failure messages sent by Navy ships. They used syntactic analysis and produced a semantic representation by using a verb decomposition dictionary.

Hull et al. [5] used a manually compiled dictionary to distinguish nominalized phrases from ordinary noun phrases, and further, to disambiguate the verbal sense and determine the fillers of the thematic roles of the verb.

In these knowledge-based approaches, we have to compile or modify the dictionary when porting the system to a new domain. Compiling and maintaining the dictionary is an expensive and time consuming task. This defect led us to choose another approach, a corpus-based one.

Grefenstette et al. [4] compared arguments attached to verbal forms and potential nominalization to find semantically emptied support verbs (e.g., *make* a proposal). They selected the three most common prepositions following a particular nominalization and extracted verbs for which these nominalizations were considered to be direct objects. Their assumption is that a nominalized phrase has a syntactic structure parallel to the original verb. Note that their approach utilized only prepositional phrase similarity.

structured part

General Information	
Report Number:	80094
Local Date(YrMon):	198801
Light Condition:	DAYLIGHT
Primary Problem Area:	AIRCRAFT AND THEIR SUBSYSTEMS
...	...

unstructured part

Narrative:
JUST AFTER LEVEL OFF AT CRS, CABIN BEGAN TO CLB AT APPROX 3000 FPM. AUTO FAIL AND STAND BY LIGHTS CAME ON. PRESSURE CTLR SET TO MANUAL. AC AND DC AND GND WERE SELECTED AND USED. VALVE INDICATOR SHOWED CLOSED ENTIRE TIME. PACKS SWITCHED TO HIGH-REDUCED CLB TO 1500'. DURING ABOVE CALLED ATC AND CLRD TO FL240, ...

Synopsis
PRESSURIZATION CTL ON ACR MLG LOST AS ACFT REACHED CRUISE ALT OF FL350. ARTCC AMENDED ALT ASSIGNMENT TO 10000' ON REQUEST. OXYGEN MASKS DROPPED BEFORE PRESSURIZATION CTL REGAINED. ACFT RETURNED TO DEP ARPT.

Fig. 1 An example of an ASRS report.

Our corpus-based approach to deal with nominalization employs the characteristics mentioned above. In the above examples, various nominalizations share the same predicate/argument structure of the corresponding clause. For text mining applications, we need to transform unstructured texts into structured data. In this process, texts that have the same meaning but are represented with different expressions should be transformed into the same clause. Hence, we transform nominalized phrases into clauses, and then transform them again into predicate/argument structures. For simplicity, we use only subjects, objects and the first prepositional phrases in the predicate/argument structure. We collect the instances of these argument fillers from a corpus and analyze their distribution.

3. Methods

3.1 Target Domain

There are a large number of aviation safety reports available in electronic form. It is becoming increasingly important to learn from such reports to prevent future accidents.

Safety reports are written as unstructured texts by reporters and submitted to the safety report organization. Analysts read these reports and change some parts into structured data (such as "weather", "location"), but the main parts of the reports remain in their unstructured form. Thus, while analysis of unstructured text is of utmost importance, it obviously takes a great deal of time for humans to read every report. Therefore, analyzing these reports by computer becomes necessary. However, there is very few works in this domain, and the treatment of nominalization has not been done [10], [13].

Our corpus was composed of aviation safety reports of ASRS (Aviation Safety Reporting System) which is administrated by NASA. ASRS reports are collections of voluntarily submitted aviation safety incident reports written in English whose aim is to lessen the likelihood of aviation accidents in the future. ASRS reports consist of two parts, structured and unstructured parts. The structured part has information with a fixed format, such as date, weather and so on. The unstructured part includes descriptions in plain text written by people who were involved in or observed the incidents. This part is called the "narrative". The unstructured part has a subpart named "synopsis" which is a summarization of the narrative part written by the analysts. We used 24,626 ASRS synopses, in which the average words per document are 18.1 words (Fig. 1).

3.2 Abbreviation Expansion

In the ASRS reports, abbreviations make up approximately 10.5% of all words based on a manual count of words in an 8000-word sample. We defined an abbreviation to be a short representation of a single word (e.g., "ACFT"–"aircraft"). To expand the abbreviations, we applied a previously developed method [17]. After manually correcting the system errors, we made an abbreviation list of 293 abbreviations. We expanded the abbreviations in the text by using this abbreviation list. For example, "Flight crew was *clred* for a visual approach." is expanded to "Flight crew was *cleared* for a visual approach." Because the abbreviations are of various parts of speech, they should be expanded before parsing the text, to improve the parser performance.

The ASRS reports are about 6.4% acronyms. The number of acronyms is rather small, and we made a list of 198 of them. Because their parts of speech are all nouns, we didn't need to expand them, but only to tell the parser their parts of speech.

3.3 Clause Extraction

We employed the Apple Pie Parser to parse the text [15]. To recover root forms from inflected words, we employed TreeTagger [14]. "Subject", "verb", "direct object", and the first "prepositional phrase" were identified using the parsed trees. If the "sentence" consisted of only a noun phrase, it was identified as a noun phrase.

An example of a sentence and the predicate/argument structure extracted from it is shown below:

8. Aircraft took off without clearance.
9. $(\text{Aircraft})_{sbj}$ $(\text{without clearance})_{pp}$ $(\text{take\_off})_{v}$

Our goal was to extract the same predicate/argument structure (9) from the nominalized phrase (10) indicating the sentence with the same meaning (8).

10. Aircraft takeoff without clearance.

3.4 Nominalization Detection and Expansion

The algorithm for transforming the nominalized phrase into clauses is triggered when a head noun of a noun phrase can be derived from a verb. This criterion is the same as Hull et al.'s [5]. In order to identify the relations between verbs and nouns derived from them, we compiled a list of nouns with their corresponding verbs that could be nominalized. We called this list the "nominalization candidate list". To enumerate the nouns, WordNet was consulted to see if any sense of the noun could be classified into "actions" or "events" [9]. The total number of the nouns in the list was 776 and only 314 of them were included in NOMLEX. This suggests that the rest of the nominalizations are domain dependent.

To extract the predicate/argument structure, we took account of a noun phrase (NP) and the nearest prepositional phrase (PP) modifying that noun phrase.

Our algorithm works as follows. It receives a noun phrase as input and outputs predicate/argument structures with scores. If the input noun phrase has no nominalization of a clause, the algorithm outputs nothing. It means the input noun phrase is not a nominalization.

step 1 Mark the right-most noun, except ones governed by a preposition in the input NP, as the head noun. If the head noun is in the nominalization candidate list, goto step 2, otherwise output no structure.

step 2 If there is a genitive noun preceding the head noun, mark it as *sbj*.

step 3 If there is a PP headed by "by", generate a hypothesis marking its complement NP as *sbj* unless another noun has been already marked as *sbj* in step 2, and the other complement NP as *obj*. Then goto step 6.

step 4 If there is a PP headed by "of", generate two hypotheses marking its complement NP as *sbj* and *obj* and other complement NP(s) as *obj* and *sbj*, respectively. If a noun has already been marked as *sbj* in step 2, reject the hypothesis marking the PP's complement as *sbj*. Then goto step 6.

step 5 If there is a PP headed by neither "by" nor "of", reserve it for the later use.

(a) If there is a noun preceding the head noun and it is not a genitive noun, generate three hypotheses marking the noun as *sbj*, *obj* and *pp*. If the noun has already been marked as *sbj* in step 2, reject the hypothesis marking the noun as *sbj*.

(b) If there is more than one noun preceding the head noun, enumerate all possible segmentations of the nouns into one or two elements. For example, if the preceding nouns are $n_1n_2n_3$, there are three possible segmentations: $n_1n_2n_3$, n_1/n_2n_3 and n_1n_2/n_3. If a genitive noun exists among the nouns, the segment boundary is restricted to just after the right-most genitive noun. Segmentations including elements that do not appear in the corpus are filtered out in this step.

When the segmentation consists of one element ($n_1n_2n_3$ case in the above example), the process is as in step 5(a). When the segmentation consists of two elements, generate three hypotheses marking the two elements as *sbj-obj*, *sbj-pp* and *obj-pp*. Note that the combination *obj-sbj* is not allowed by the ordering constraint mentioned in Sect. 2. If there is a genitive noun and its role assignment is inconsistent with that of step 2, reject the hypothesis.

Goto step 6.

step 6 If there is no hypothesis at this stage, exit the algorithm without output. Otherwise, for each hypothesis, count the occurrences of the marked elements in the corpus. When counting the occurrences, the corresponding verb of the head noun is identified by referring to the nominalization candidate list, and each element is checked to see if it appears with the same role of the verb as marked in the previous steps. If an element is a compound noun, the occurrence of the right-most noun is counted.

When counting the *pp* role, the occurrences are counted for each preposition and the maximum preposition occurrence is used as the *pp* occurrence. After checking against the corpus, if there is no *pp* role and there is a reserved PP in step 5, add the reserved PP as the *pp* role element.

Hypotheses with an element not appearing in the corpus are rejected in this step.

step 7 Form the sum of the occurrences of all elements in the hypotheses, and return the hypothesis with the highest sum.

The rationale of our algorithm is that a nominalization can be transformed into its predicate/argument structure by referring to the occurrences of corresponding clauses, in particular, its verb form and arguments. Semantic constraints of a particular verb is automatically obtained for the predicate/argument structure by referring to the corpus.

We will show a couple of examples to help the reader see how the algorithm works.

11. Commuter smt[†] propeller strike.

The algorithm starts by identifying the head noun "strike". "Strike" is found in the nominalization candidate list (step 1). Since there aren't any prepositional phrases, the algorithm goes to step 5 and enumerates the possible segmentations of the pre-nominal modifier. In this case, the possible segmentations are: "commuter smt propeller", "commuter smt/propeller", and "commuter/smt propeller". The possible hypotheses that can be derived from the above segmentations are shown below (*p* denotes a missing preposition):

12. (Commuter smt propeller)$_{sbj}$ (strike)$_v$
13. (Commuter smt propeller)$_{obj}$ (strike)$_v$
14. (Commuter smt propeller)$_{pp}$ (strike)$_v$
15. (Commuter smt)$_{sbj}$ (propeller)$_{obj}$ (strike)$_v$
16. (Commuter smt)$_{sbj}$ (*p* propeller)$_{pp}$ (strike)$_v$
17. (Commuter smt)$_{obj}$ (*p* propeller)$_{pp}$ (strike)$_v$
18. (Commuter)$_{sbj}$ (smt propeller)$_{obj}$ (strike)$_v$
19. (Commuter)$_{sbj}$ (*p* smt propeller)$_{pp}$ (strike)$_v$
20. (Commuter)$_{obj}$ (*p* smt propeller)$_{pp}$ (strike)$_v$

No occurrence of "Commuter smt propeller", and "smt propeller" in the corpus is to be found in the text; thus the hypotheses 12, 13, 14, 18, 19, and 20 are filtered out in step 5(b). Since there is one occurrence of "smt" in the subject role in a noun phrase "acr smt" and one occurrence of "propeller" in the object role in the corpus, the hypothesis 15 is given a score of 2. On the other hand, there is no occurrence of "propeller" as PP complement NP; thus, the hypotheses 16 and 17 are rejected. There remains only one hypothesis 15, which is returned as output.

The second example is

21. Pilot altitude deviation.

"Deviation" is identified as the head noun of NP. When the preceding nouns "pilot altitude" is not segmented, the algorithm enumerates the possible hypotheses as follows:

22. (pilot altitude)$_{sbj}$ (deviate)$_v$
23. (pilot altitude)$_{obj}$ (deviate)$_v$
24. (*p* pilot altitude)$_{pp}$ (deviate)$_v$

There is no occurrence of "pilot altitude" in the corpus; thus hypotheses 22, 23, and 24 are filtered out.

When the preceding nouns "pilot altitude" are segmented into "pilot/altitude", the algorithm enumerates the possible hypotheses as follows:

25. (pilot)$_{sbj}$ (altitude)$_{obj}$ (deviate)$_v$
26. (pilot)$_{sbj}$ (*p* altitude)$_{pp}$ (deviate)$_v$
27. (pilot)$_{obj}$ (*p* altitude)$_{pp}$ (deviate)$_v$

[†] "Smt" is an acronym of "small transport".

"Pilot" appears in the right-most segment as the subject of "deviate" 22 times (e.g., "pilot":13, "ga student pilot":1,...). "Altitude" appears in the right-most segment as the object of "deviate" 2 times. Thus hypothesis 25 obtains a score of 24. As "Pilot" does not appear in the right-most noun as the object of "deviate", hypothesis 27 is given a score of 0. "Altitude" appears in the right-most noun complement of the preposition "from" of "deviate" 21 times (e.g., "clearance altitude:3, "assigned altitude":15,...). "Altitude" appears in the right-most noun complement of the preposition "at" of "deviate" 2 times (e.g., "same altitude:1, "low altitude";1). Then maximum preposition occurrence, i.e., the number of occurrences of "from", is selected and hypothesis 26 is given a score of 43. The highest scoring hypothesis 26 is returned as output.

4. Evaluation

We conducted a series of experiments in order to see how well the proposed nominalization transformation algorithm works and also to see how the algorithm contributes to text mining applications. We used 24,626 synopses of ASRS reports, their size being about 2.9 Mbytes.

4.1 Nominalization Transformation

As test sets, we used the 20 most frequent nominalizations in the nominalization candidate list that appear as the head noun in a noun phrase.

In Table 1, the column "Rank" shows the frequency rank of nominalizations in the nominalization candidate list, the column "Freq." shows the nominalizations frequency in the corpus, the column "Rank in all" shows the frequency rank of nominalizations among all nouns in the corpus, and the column "Nominalization" shows each nominalization. The 20 most frequent nominalizations appear in the top 50 nouns. This suggests that the treatment of nominalization is important.

Most nominalizations have both a "verbal sense" and a "non-verbal sense". Here, "verbal sense" means a usage that has a corresponding clause of the same meaning. This is the usage that the proposed algorithm should deal with. On the other hand, the example "operational deviation" has no corresponding clause, thus it is classified as having a "non-verbal sense". The column "Usage dist." shows the distribution of the usage, which was manually judged.

We selected 20 cases of every nominalization (if the total count of specific nominalizations as the head noun of NP is less than 20, all occurrences are used.) and checked whether the nominalization transformation algorithm works correctly. The results are shown in Table 1. Table 1 shows that the algorithm correctly extracts predicate/argument structures from nominalized phrases with a 54% accuracy on average and also identifies non-verbal usages with an 85% accuracy. The column "Total" means the total accuracy of verbal and non-verbal usages.

There are two main causes of error in transforming nominalizations. One is the case where corresponding arguments do not appear in the clause structure in the corpus. For example, "Atx takeoff" can not be transformed into "$(\text{Atx})_{sbj}$ $(\text{take off})_v$", because there is no occurrence of "atx" in the right-most noun as a subject role of "take off" in the ASRS corpus. This is a limitation of our algorithm.

Other cases of error are as follows. When an adjective modifies the nominalization, it is not always mapped to an adverb in the clause structure, as shown in the following examples [1], [6]:

28. John's sudden refusal.
29. John refused suddenly.
30. John's curious resemblance to Bill
31. *John resembles Bill curiously

We could expect an improvement in accuracy if we use a mapping list of adjectives. We manually compiled a list defining corresponding expressions of adjectives, such as in "unauthorized" corresponds to "without authorization", and applied it to the algorithm. The accuracy in parentheses in Table 1 shows the accuracy after using this mapping list. Using the list increased the total accuracy to 78%.

There are also parser errors. We will discuss these in Sect. 5.

4.2 Application to Text Mining

We tested whether applying the nominalization transformation algorithm could improve text mining applications. We evaluated the algorithm in a task that extracted events and their cause-effect relations, which are important semantic relations.

Cause-effect relations are expressed implicitly or explicitly in texts. Khoo et al. [7] used linguistic patterns to identify explicitly expressed cause-effect relations, to improve information retrieval performance. They used hand-crafted rules. Girju et al. [3] proposed a method of semi-automatic detection of causal relationships by detecting lexico-syntactic patterns. They used simple causal patterns to find new verbs indicating cause-effect relations. We evaluated the proposed algorithm by using both methods.

4.2.1 Simple Cause-Effect Relations

We used the following surface pattern to identify cause-effect relations.

X $\langle clue \rangle$ Y

Here $\langle clue \rangle$ is one of the following surface clue:

- prepositional phrase : "because of", "due to"

Table 1 Nominalization transformation results.

Rank	Freq.	Rank	Nominalization	Usage dist.	Accuracy		Total
		in all		verbal:non-verbal	Verbal	Non-Verbal	
1	5,232	2	flight	3 : 17	67%	65%	65%
2	3,518	5	pilot	0 : 20	N.A.	90%	90%
3	3,439	6	approach	12 : 8	42%(58%)	88%	60%(70%)
4	2,729	9	landing	14 : 6	43%(86%)	100%	60%(90%)
5	2,299	11	deviation	7 : 13	86%	100%	95%
6	2,288	12	takeoff	18 : 2	56%(72%)	100%	60%(75%)
7	2,024	13	separation	0 : 20	N.A.	100%	100%
8	1,204	17	error	0 : 20	N.A.	100%	100%
9	1,182	18	descent	12 : 8	58%(92%)	88%	70%(90%)
10	1,129	19	departure	5 : 11	40%	36%	38%
11	799	27	control	0 : 10	N.A.	80%	80%
12	777	28	climb	13 : 7	31%(38%)	71%	45%(50%)
13	696	32	land	17 : 3	53%	100%	60%
14	560	39	track	N.A.	N.A.	N.A.	N.A.
15	552	40	failure	20 : 0	70%	N.A.	70%
16	535	41	restriction	0 : 11	N.A.	100%	100%
17	521	43	loss	19 : 1	73%(84%)	100%	75%(85%)
18	513	44	penetration	20 : 0	40%(85%)	N.A.	40%(85%)
19	489	47	communication	3 : 10	66%	90%	85%
20	455	50	turn	1 : 8	0%	50%	44%
Total				48% : 52%	54%(71%)	85%	70%(78%)

- causative verb : "cause", "caused by", "result in", "result from", "lead to"

For example, example 32 matches the pattern "NP1 *caused* NP2"; therefore a causal relation "(Aircraft loss of oil pressure)$_{NP1}$ → (engine shutdown)$_{NP2}$" will be extracted.

32. Aircraft loss of oil pressure caused engine shutdown.
33. Aircraft lost oil pressure causing engine shutdown.

Example 33 also conveys the same meaning as example 32. Our goal is to identify example 32 and example 33 as having the same predicate/argument structure by using the nominalization transformation algorithm.

We evaluated four different methods of extracting cause-effect relations from a sentence. Each method allows different elements to be X and Y in the above surface pattern.

- NP: If X or Y is NP, only the base NP is allowed to be X or Y.
- NP+PP: If X or Y is NP with PP, the NP with the first PP is allowed to be X or Y. If X or Y is NP without PP, the base NP is extracted as either X or Y.
- P/A: In addition to NP+PP, the clause is allowed to be X or Y. When a clause is found, it is parsed and transformed into the predicate/argument structure.
- P/A+: The proposed algorithm is applied to NP and NP+PP, if possible. Otherwise, it is the same as the P/A method.

The results of extraction using the four methods on example 32 and example 33 are shown below:

- NP: 32: (aircraft loss)$_{np}$ → (engine shutdown)$_{np}$
  33: N.A.
- NP+PP: 32:(aircraft loss)$_{np}$ (of oil pressure)$_{pp}$
  → (engine shutdown)$_{np}$
  33: N.A.
- P/A: 32: (aircraft loss)$_{np}$ (of oil pressure)$_{pp}$
  → (engine shutdown)$_{np}$
  33: (aircraft)$_{sbj}$ (oil pressure)$_{obj}$ (lose)$_{v}$
  → (engine shutdown)$_{np}$
- P/A+: 32: (aircraft)$_{sbj}$ (oil pressure)$_{obj}$ (lose)$_{v}$
  → (engine)$_{obj}$ (shut_down)$_{v}$
  33: (aircraft)$_{sbj}$ (oil pressure)$_{obj}$ (lose)$_{v}$
  → (engine)$_{obj}$ (shut_down)$_{v}$

These results show only P/A+ can identify example 32 and example 33 as having the same predicate/argument structure. After identifying cause-effect relations, we applied Prefix Span for sequential pattern mining to find frequent patterns [11]. We chose minimum support 2 to filter out spelling errors and errors caused by the parser and to extract frequent patterns.

The results were evaluated by three annotators, two airline captains and one flight engineer whose work concerned aviation safety. We requested the annotators to classify the extracted events or patterns accordingly:

useful: The extracted events or patterns seem to be useful for further investigation to prevent future incident/accidents, e.g., "altitude deviation".

vague: The extracted events or patterns seem to be somewhat too broad in meaning to be understood as meaning something specific, but nonetheless may be of some help for further investigation. For example, "fatigue" or "loss" as in "Loss of oil pressure" may be classified as "useful".

Table 2 Cause-effect relations.

	Useful	Vague	Meaningless	Total
NP (Ave. no.)	5.3	4.7	1	11
(%)	51.5	39.4	9.1	
NP+PP (Ave. no.)	3.7	0.3	0	4
(%)	92.0	8.3	0	
P/A (Ave. no.)	13.3	9	0.6	23
(%)	58.0	39.1	2.9	
P/A+ (Ave. no.)	16.3	6.3	1.4	23
(%)	71.0	27.5	1.4	

Table 3 Cause-effect events.

	Useful	Vague	Meaningless	Total
NP (Ave. no.)	57	21	14	92
(%)	62.0	22.8	15.2	
NP+PP (Ave. no.)	55.7	9.7	7.7	73
(%)	76.3	13.2	10.5	
P/A (Ave. no.)	151.7	43.7	25.7	221
(%)	68.6	19.8	11.6	
P/A+ (Ave. no.)	166.3	43.3	26.3	236
(%)	70.5	18.4	11.2	

meaningless: The extracted events or patterns cannot be understood, or seem to be too broad in meaning, or seem not to be useful for further investigation, e.g., “landing”.

Table 2 shows the average number of instances in each class as selected by the annotators. The average κ coefficient of the annotators was 0.49, which means their classifications were consistent at the medium level [16]. The relations obtained by P/A are about two times more numerous than those obtained by NP, because there are many causal relations including clauses (e.g., “Aircraft rejected takeoff due to engine problem”).

We obtained only 4 relations by using NP+PP, because the same NP+PP does not appear more than once, as different PPs modified NP. We obtained 23 relations with P/A and P/A+.

The number of average useful relations obtained by P/A+ is higher than that by P/A by 3. Three patterns were transformed from NP into P/A. For example, “(distraction)$_{np}$ → (altitude deviation)$_{np}$” can be transformed into “(distraction)$_{np}$ → (from altitude)$_{pp}$ (deviate)$_{v}$”. However, a detailed investigation of the annotators’ classifications revealed that these three patterns were classified as “useful” both before and after the nominalization transformation. Consequently, these transformations did not contribute to any improvement, and we found that the results showing a small improvement were accidental.

Next, we evaluated the results with the minimum pattern length 1 and minimum support 2 (Table 3). This means that the extracted patterns included events (pattern length 1). Thus, the number of extracted patterns was about 10 times more than those in Table 2.

Table 3 shows that the result of P/A+ was the best of the four. Detailed investigation of P/A and P/A+ also showed that introducing nominalization transformation algorithm improved the events extraction performance. For example, events categorized as NP “oil quantity loss” and “loss of oil quantity” appear once respectively in the ASRS corpus; thus these same-meaning events were not extracted in P/A (minimum support 2). These events were transformed into others with the same predicate/argument structure “(oil quantity)$_{obj}$ (lose)$_{v}$” and then classified as “useful”. These kinds of transformation occurred 5 times and contributed to an improvement in the text mining (e.g., “(damage)$_{v}$ (propeller)$_{obj}$”, “(without authorization)$_{pp}$ (land)$_{v}$”, “(nose gear)$_{sbj}$(collapse)$_{v}$”, “(passenger)$_{sbj}$(injure)$_{v}$”).

4.3 Finding Causative Verbs

Girju et al. [3] proposed a method to find verbs expressing causality. There are several causative verbs that do not always indicate causal relations, such as “make”. They collected surface patterns from WordNet that always indicate causality, such as “NP1 causes NP2” [9]. Then they collected NP1-NP2 pairs from these patterns, and considered that the verbs in the context of “NP1 verb NP2” also indicate causality.

In their experiments, they used only noun phrases to find verbs. We conducted experiments by using NP, NP+PP, and P/A+. We found 61 patterns by using NP of which 30 were correct, 16 patterns by using NP+PP of which 8 were correct, and 21 patterns by using P/A+ of which only 10 were correct. We could not find sufficient patterns with P/A+.

5. Discussion

5.1 Parser Error Correction

Throughout our experiments, we noticed that certain noun phrases cause a parser error in which a noun phrase is combined with the succeeding subject role noun phrase as in the following example:
“acr mlg altitude deviation crossing restriction not met.”
The parser returned the result:

34. (acr mlg altitude deviation crossing restriction)$_{sbj}$
    (not met)$_{v}$

But the correct parse is:

35. (acr mlg altitude deviation)$_{np}$
    (crossing restriction)$_{sbj}$ (not met)$_{v}$

We modified the algorithm to find the nominalization not only at the end of the noun phrase excluding succeeding PP, but also at the pre-nominal position (step 1). This modification generates more hypotheses,

Table 4 WSJ corpus nominalization transformation.

	Accuracy
loss	40%
decline	10%
decision	20%
proposal	20%

many of which are incorrect interpretations. To reduce incorrect analyses, we set the threshold on the score to return a hypothesis as a result. This threshold was applied only when a head noun is identified in the prenominal position.

We evaluated the modified algorithm by using 20 test cases that were not correctly analyzed by the original algorithm. The threshold value was set to 20. Fourteen were correctly analyzed (70% accuracy). Three errors were caused by the fact that these cases violated the assumptions of the algorithm, like in the following:

36. Ga sma unauthorized penetration tca.

In sentence 36, "penetration" is identified as a nominalization, but the object argument "tca" appears after the head noun without "of", which is inconsistent with the algorithm's assumption. Two errors were due to the parser error. One error was due to two nominalizations being connected with a coordinate conjunction.

5.2 Porting to Other Domain

We evaluated the algorithm by using an excerpt of the 1989 WSJ corpus that included 541,910 words and whose size was about 2.9 Mbytes. We tested four frequent nominalizations, "loss", "decline", "decision", and "proposal" (10 examples of each nominalization).

Table 4 shows that average accuracy was 23%. The reason for this worse result was mainly due to the parser error. 55% of the errors were due to parser errors. 20% of the errors were due to the fact that the corpus did not include clause structures corresponding to nominalizations. Another reason for the poorer performance was that our algorithm does not deal with relative clauses (e.g., "Mr. Spoon said ... that has reported *decline* in operating profit ...").

6. Conclusion

We proposed a corpus-based method of transforming nominalized phrases into clauses. The results of 20 nominalizations showed a 78% average accuracy with aviation domain texts. Many of the past researches used a semantic dictionary for analyzing the corresponding argument role when transforming nominalized phrases into clauses. This means that porting the system to other domains requires re-construction of the dictionary. We showed that instead of constructing a semantic dictionary, we can obtain a semantic restriction of arguments automatically by using corpora.

We applied the proposed algorithm to a text mining application that extracts events and their causal relations in texts. The results showed that our algorithm improved the text mining performance. We also showed that the predicate/argument structure is more useful than noun-only phrases when extracting causal relations.

To port the system to other domains, we have to extend surface patterns to extract clause structures. For example, through the experiments using the WSJ corpus, we found that relative clauses were a promising resource to extract predicate/argument structures.

Acknowledgment

The authors would like to thank the anonymous referees for their useful comments on the earlier version of this paper. We also thank Satoshi Sekine (New York University) for letting us use of the Apple Pie Parser, Helmut Schmid for letting us use of the TreeTagger, and Taku Kudo for letting us use of the PrefixSpan program. We also thank our annotators A. Ueno, T. Sasaki, and S. Hoshino of Japan Air Lines Co., Ltd., for their assistance.

References

[1] N. Chomsky, "Remarks on nominalization" Readings in English Transformational Grammar, pp.184–221, 1970.

[2] D.A. Dahl, M.S. Palmer, R.J. Passonneau, "Nominalizations in PUNDIT," Proc. 25th Annual Meeting of the ACL, pp.131–137, 1987.

[3] R. Girju and D. Moldovan, "Mining answer for causation questions," Proc. AAAI Spring Symposium, 2002.

[4] G. Grefenstette and S. Teufel, "Corpus-based method for automatic identification of support verbs for nominalizations," EACL-95, 1995.

[5] R.D. Hull and F. Gomez, "Semantic interpretation of nominalizations," Proc. AAAI-96, pp.1062–1068, 1996.

[6] K. Inoue, H. Yamada, T. Kawano, and H. Narita, Meishi (Nominal), Kenkyuusha, 1985.

[7] C.S.G. Khoo, S.H. Myaeng, and R.N. Oddy, "Using cause-effect relations in text to improve information retrieval precision," Inf. Process. and Manage., vol.31 no.1, pp.119–145, 2001.

[8] C. Macleod, A. Meyers, R. Grishman, L. Barrett, and R. Reeves, "Designing a dictionary of derived nominals," Proc. Recent Advances in Natural Language Processing, 1997.

[9] G.A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. Miller, "Introduction to wordNet: An on-line lexical database,", Technical Report, Princeton, CSL Report 43, 1993.

[10] Z. Nazeri, E. Bloedom,and P. Ostwald, "Experience in mining aviation safety data," Proc. ACM SIGMOD 2001, pp.562–566, 2001.

[11] H. Pinto, J. Han, J. Pei, and K. Wang, "Multi-dimensional sequential pattern mining," Proc. 10th ACM International Conference on Information and Knowledge Management(CIKM'01), 2001.

[12] R. Quirk, S. Greenbaum, G. Leech, and J. Svartvik, A Comprehensive Grammar of the English Language, LONGMAN group, 1985.

[13] B. Sada, “Making effective use of aviation safety narratives”, GAIN Asia-Pacific Regional Conference, 2002.
[14] H. Schmid, “Probabilistic part-of-speech tagging using decision trees,” Proc. International Conference on New Methods in Language Processing, pp.44–49, 1994.
[15] S. Sekine and R. Grishman, “A corpus-based probabilistic grammar with only two non-terminals,” Proc. International Workshop on Parsing Technology, 1995.
[16] S. Siegel and Castellan Jr., “Nonparametric statics for the behavioral sciences,” Second ed. McGraw Hill, 1988.
[17] A. Terada and T. Tokunaga, “Automatic disabbreviation by using context information,” Proc. NLPRS’2001 Paraphrase Workshop, pp.21–28, 2001.

Akira Terada He is an employee of Japan Airlines Co.Ltd.,. He received the B.S. and M.S. degrees in Electrical Engineering from Kyoto University in 1976 and 1978, respectively. He received the Dr.Eng. degree from Tokyo Institute of Technology in 2003. His current research interests are information retrieval, computational linguistics, text mining, and aviation safety.

Takenobu Tokunaga He is an associate professor of Graduate School of Information Science and Engineering, Tokyo Institute of Technology. He received the B.S. degree in 1983 from Tokyo Institute of Technology, the M.S. and the Dr.Eng. degrees from Tokyo Institute of Technology in 1985 and 1991, respectively. His current interests are computational linguistics and information retrieval.