A Domain Ontology Production Tool Kit Based on Automatically Constructed Case Frames

Yoji Kiyota[*], **Hiroshi Nakagawa**[*]

[*]Information Technology Center, University of Tokyo
General Library, 7-3-1 Hongo, Bunkyo-ku, Tokyo 113-0033 Japan
kiyota@r.dl.itc.u-tokyo.ac.jp, nakagawa@dl.itc.u-tokyo.ac.jp

Abstract

This paper proposes a tool kit to produce a domain ontology for text mining, based on case frames automatically constructed from a raw corpus of a specific domain. Since case frames are strongly related to implicit facts hidden in large domain-specific corpora, we can say that case frames are a promising device for text mining. The aim of the tool kit is to enable automatic analysis of event reports, from which implicit factors of the events are to be extracted. The tool kit enables us to produce a domain ontology by iterating associative retrieval of case frames and manual refinement. In this study, the tool kit is applied to the Japan Airlines pilot report collection, and a domain ontology of contributing factors in the civil aviation domain is experimentally produced. A lot of interesting examples are found in the ontology. In addition, a brief examination of the production process shows the efficiency of the tool kit.

1. Background

Recently, report collection systems on safety events have been developed, to prevent accidents and failure cases in civil aviation, railway systems, medicine and other industries. For example, the FAA (Federal Aviation Administration) and NASA (National Aeronautics and Space Administration) operate the ASRS[1] (Aviation Safety Reporting System), which collects voluntarily submitted aviation safety incident reports from pilots, air traffic controllers and others. The JST (Japan Science and Technology Agency) also developed the Failure Knowledge Database[2], which stores stories of accidents and failure cases in various fields. As a result, a large amount of reports written in natural languages have been accumulated.

As for civil aviation, most of accidents and incidents are considered to be caused by human errors. Human errors are induced by various contributing factors, e.g., misjudgment, carelessness, ignorance, organizational problem, and poor planning. These factors are also induced by other factors, e.g., narrow view, poor communication, lack of knowledge, and poor management. However, resources (e.g., time and budget) for taking measures to reduce those factors are usually limited. To cope with the factors efficiently, we have to decide priority order of the measures, by estimating which contributing factors are more crucial.

To make use of the collected reports for deciding measures to reduce accidents, a domain ontology is essential because of the following reasons: the amount of reports is too much to analyze manually, so automatic analysis based on text mining techniques is required; the contributing factors are often not described explicitly in each report, but are implied by other expressions. For example, if an incident was induced by pressure to keep schedule punctual, the reports may imply the existence of the factor by typical expressions, including “departure delays”, “arrive n minutes late”, and “put off departure”. Since those expressions are usually specific to a domain, to find out implicit factors with expressions from a vast amount of the above report texts, a domain ontology of such expressions is required.

Because of huge cost needed to produce a domain ontology manually, there have been demands for automatic production techniques. However, quality of an automatically produced ontology is not currently sufficient for real-world applications. Our approach to cope with this problem is to combine automatic clustering and manual refinement efficiently.

Now, what kind of domain ontology is required? To find such contributing factors, the following relations should be defined:

(a) Relations between keywords, such as synonyms, antonyms, and word classes.

(b) Relations between phrasal expressions. For example, “find cracks” and “detect damages” are almost synonymous.

(c) Association of the above keywords and phrasal expressions with contributing factors which the keywords and expressions imply.

We think that extraction of the above relations (a)-(c) can be integrated into clustering of case frames, which describe what kinds of cases each predicate (usually verbs) has and what kinds of examples (usually nouns) can fill case slots. Figure 1 shows some case frames in the civil aviation domain in Japanese. For example, a verb 遅れる ‘delay’ has cases such as nominative(nom), instrumental(ins) and ablative(abl); and 出発 ‘departure’ can fill the nominative case. The reasons why we use case frames are as follows:

- Predicate-argument pairs are useful for deriving relations between words such as synonyms, as Hindle (1990) pointed out.
- Each case frame usually corresponds to a typical phrase in a domain. If two phrases are synonymous, and their corresponding case frames are similar to each other, those phrases can be extracted as synonymous phrases based on similarity between case frames.

[1]http://asrs.arc.nasa.gov/
[2]http://shippai.jst.go.jp/

遅れる ‘delay’(1)

が (nom):	出発 ‘departure’:30
で (ins):	delay:5, 積雪 ‘snow’:5
より (abl):	SKD‘scheduled’:10

生ずる ‘cause’(1)

を (acc):	delay:25
で (ins):	不具合 ‘defect’:5
の (gen):	[TIME]:20

なる ‘be’(1)

が (nom):	出発 ‘departure’:100
に (dat):	delay:10
より (abl):	SKD‘scheduled’:10, [TIME]:80

なる ‘be’(2)

が (nom):	到着 ‘arrival’:70
に (dat):	delay:50
より (abl):	SKD‘scheduled’:20, [TIME]:10

Figure 1: Case Frames

- Case frames which are related to a same contributing factor often share same examples. For example, case frames related to a factor “pressure to keep schedule punctual” share examples such as “delay”, “scheduled”, and “departure”.
- An occurrence of a phrase often strongly implies existence of a contributing factor, than that of an individual term occurrence does.

This paper proposes a tool kit to extract the above relations (currently excluding the (a) relation), based on the automatic case frame construction method proposed by Kawahara and Kurohashi (2001), which clusters predicate-argument pairs in a raw corpus into high-quality case frames. The tool kit enables us to gather up case frames related to a specific contributing factor, by iterating associative retrieval of case frames and manual refinements.

2. Tool Kit Overview

2.1. Architecture

The tool kit supports us to produce a so-called contributing factor ontology, in which each factor is associated with related case frames. Iteration of associative retrieval of case frames and manual refinements will make it possible to produce a high-quality ontology, at a relatively low cost. Figure 2 shows the architecture of the tool kit. It consists of the case frame construction module, the case frame associative retrieval module, and the user interface.

The case frame construction module applies the method proposed by Kawahara and Kurohashi (2001) to a raw corpus of a targeted domain, and constructs domain-specific case frames. See Section 3. for more details.

The case frame associative retrieval module receives a list of case frames, and retrieves similar case frames to the received ones, by calculating similarity between case frames. See Section 4. for more details.

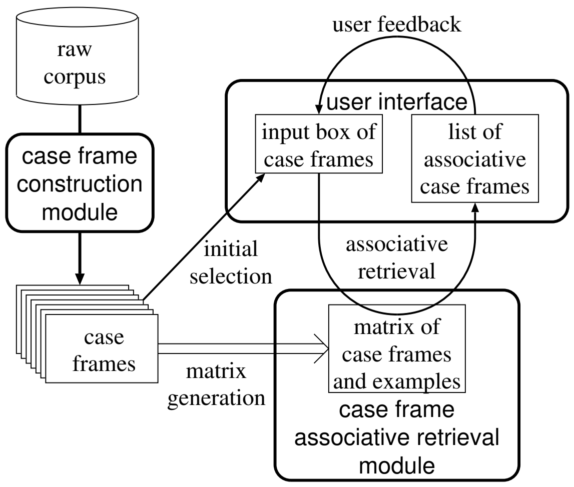
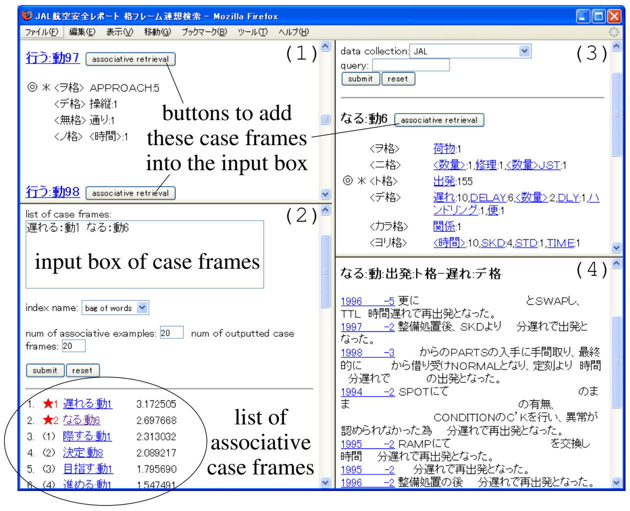

raw corpus
case frame construction module
case frames
initial selection
matrix generation
user feedback
user interface
input box of case frames
list of associative case frames
associative retrieval
matrix of case frames and examples
case frame associative retrieval module

Figure 2: Architecture


JAL航空安全レポート 格フレーム連想検索 - Mozilla Firefox
ファイル(F) 編集(E) 表示(V) 移動(G) ブックマーク(B) ツール(T) ヘルプ(H)
(1)
(2)
(3)
(4)
buttons to add these case frames into the input box
input box of case frames
list of associative case frames
associative retrieval
submit
reset

行う:動97

◎ ＊ <ヲ格> APPROACH:5
<デ格> 操縦:1
<無格> 通り:1
<ノ格> <時間>:1

行う:動98

list of case frames:
遅れる:動1 なる:動6

index name: bag of words

num of associative examples: 20 num of outputted case frames: 20

1. ★1 遅れる:動1 3.172505
2. ★2 なる:動6 2.697668
3. (1) 際する:動1 2.313032
4. (2) 決定:動8 2.089217
5. (3) 目指す:動1 1.795690
6. (4) 進める:動1 1.547491

data collection: JAL
query:

なる:動6

<ヲ格> 荷物:1
<二格> <数量>:1,修理:1,<数量>JST:1
◎ ＊ <ト格> 出発:155
<デ格> 遅れ:10,DELAY:6,<数量>:2,DLY:1,ハンドリング:1,便:1
<カラ格> 関係:1
<ヨリ格> <時間>:10,SKD:4,STD:1,TIME:1

なる:動:出発:ト格−遅れ:デ格

1996 -5 更に とSWAPし、TTL 時間遅れで再出発となった。
1997 -2 整備処置後、SKDより 分遅れで出発となった。
1998 -3 からのPARTSの入手に手間取り、最終的に から借り受けNORMALとなり、定刻より 時間 分遅れで の出発となった。
1994 -2 SPOTにて のまま の有無、 CONDITIONのC’Kを行い、異常が認められなかった為 分遅れで再出発となった。
1995 -2 RAMPにて を交換し 時間 分遅れで再出発となった。
1995 -2 分遅れで再出発となった。
1996 -2 整備処置の後 分遅れで再出発となった。

Figure 3: User Interface

The user interface has an input box of case frames, and a display window of associative case frames. The interface enables us to add any case frame into the input box, and to provide the tool kit with feedback by adding relevant case frames in the display window into the input box. See below for more details.

2.2. User Interface

Figure 3 shows the user interface of the tool kit. The interface has the following four windows:

(1) **the list of all case frames**: We can scroll up and down this window, and can select any case frame as an initial one. If we click the “associative retrieval” button, the case frame is added to the input box in window (2).

(2) **case frame associative retrieval**: This window has the input box of case frames, and the list of associative case frames. If we update the input box and click the “submit” button, the list is updated.

(3) **display of specific case frames**: If we click one of the case frames of the list in window (2), the detailed

information about the case frame is displayed. We can also retrieve case frames by specifying a verb in the query box.

(4) **display of original sentences**: If we click one of examples of a case frame in window (3), the original sentences in which the example occurs are displayed.

We can gradually grow a collection of case frames which are related to a specific contributing factor, by iterating the following steps:

1. decide which contributing factor is to be targeted (*e.g.* pressure to keep schedule punctual).
2. at window (1), select one of the case frames which seem to be related to the contributing factor (*e.g.* 遅れる 'delay'(1) in Figure 1), and add it to the input box in window (2).
3. click the submit button at window (2), and then the system displays the list of associative case frames, in order of similarity score.
4. examine the list of associative case frames, and add case frames which seem to be related to the contributing factor (*e.g.* なる 'be'(1) in Figure 1) into the input box in window (2).
5. go back to step 3.

If we can find no further relevant case frames in the list, the above process is terminated. The collection of case frames in the input box can be used as a domain ontology related to the contributing factor.

3. Case Frame Construction

The tool kit uses case frames as the basic unit of a domain ontology. Each case frame describes usages of a verb (predicate), including what kind of cases each verb has and what kinds of nouns can fill a case slot. Case frames are regarded as knowledge for natural language understanding, and are required by a lot of natural language processing techniques, *e.g.*, syntactic analysis, semantic analysis, question answering, text summarization, and machine translation. The tool kit constructs case frames automatically from a raw corpus of a specific domain, based on the method proposed by Kawahara and Kurohashi (2001).

3.1. Brief Overview of Japanese Case Frames

In Japanese, word order does not play a case-marking role. Instead, postpositions function as case markers. The basic structure of a Japanese sentence is as follows:

積雪	で	SKD	より	出発	が	遅れる
snow	ins	scheduled	abl	departure	nom	delay

'departure delays from the scheduled time due to snow'

で (ins), より (abl), and が (nom) are postpositions which mean instrumental, ablative, and nominative, respectively. Each postposition shows which case of the verb 遅れる 'delay' the adjacent noun fills. For examples, 出発 'departure' fills the nominative case of the verb 遅れる 'delay'.

Figure 1 shows some case frames constructed by the Kawahara's method from a raw corpus of the civil aviation domain. For example, looking at the case frame 遅れる 'delay'(1), we can know the verb 遅れる 'delay' has cases such as nominative(nom), instrumental(ins) and ablative(abl); and 出発 'departure' can fill the nominative case. The number after each example followed by a colon shows the frequency of the example in the raw corpus.
Note that each verb usually has multiple meanings, so it should have multiple case frames. In Figure 1, the case marker written in bold letters, *e.g.*, **が (nom)**, shows the closest case component, which determines which case frame should be used. The Kawahara's method clusters verb usages in the raw corpus, by the combination of a verb and its closest case component. For example, the following sentence has a different closest case component 発見が 'discovery nom' to the verb 遅れる 'delay', so another case frame for the verb 遅れる 'delay' is constructed.

故障	の	発見	が	遅れる
trouble	gen	discovery	nom	delay

'discovery of the trouble is delayed'

If the given raw corpus belongs to a specific domain, each case frame usually corresponds to a typical phrase in the domain, and is strongly related to a specific contributing factor. For example, the case frame 遅れる 'delay'(1) in Figure 1 corresponds to a phrase 出発が遅れる 'departure delays', and is related to a factor "pressure to keep schedule punctual".

3.2. Process of Case Frame Construction

The tool kit automatically constructs domain-specific case frames based on the method proposed by Kawahara and Kurohashi (2001). The method derives high quality case frames from a raw corpus, by coupling verbs and their closest case components. The process of the case frame construction is as follows:

1. The given raw corpus is parsed by JUMAN[3] and KNP[4] (Kurohashi and Nagao, 1994), and reliable predicate-argument relations are extracted from the parse results. These relations is called **examples**.
2. The extracted examples are distinguished by the verb and its closest case component. These data is called **example patterns**.
3. The example patterns are clustered based on a thesaurus (In this paper, no thesaurus is used currently).
4. The example patterns are converged into case frames.

4. Associative Retrieval of Case Frames

The tool kit applies associative retrieval techniques of text retrieval systems to retrieval of similar case frames. To calculate similarity score among case frames, the tool kit preliminarily generates the cooccurrence matrix of case frames and their examples. Given a list of case frames as a query, the tool kit retrieves associative case frames, and shows

[3]http://www.kc.t.u-tokyo.ac.jp/nl-resource/juman-e.html
[4]http://www.kc.t.u-tokyo.ac.jp/nl-resource/knp-e.html

	出発 ‘de-par-ture’	到着 ‘ar-ri-val’	delay	SKD ‘sched-uled’	積雪 ‘snow’	不具合 ‘def-ect’	[TIME]
遅れる ‘delay’(1)	30		5	10	5		
生ずる ‘cause’(1)			25			5	20
なる ‘be’(1)	100		10	10			80
なる ‘be’(2)		70	50	20			10

Figure 4: Matrix of Case Frames and Examples

them. The matrix generation and associative retrieval are implemented using the GETA[5] (Generic Engine for Transposable Association) (Takano et al., 2001). See (Takano et al., 2000) about the concept of the associative text retrieval.

4.1. Matrix of Case Frames and Examples

To apply associative text retrieval techniques, the tool kit preliminarily generates the cooccurrence matrix of case frames and their examples. Namely, each case frame is regarded as a document, and each example in a case frame is regarded as a occurrence of a term in a document. Frequency of examples is also considered. Given the four case frames in Figure 1, the matrix in Figure 4 is generated.
To handle the matrix of case frames and examples efficiently, the tool kit uses the GETA, which is designed for manipulating very large dimensional sparse matrices, which typically appear as index files for large scale text retrieval. The GETA is implemented using C language, and efficiently calculates similarity of texts, terms, and so on.

4.2. Similarity Calculation

The tool kit retrieves associative case frames for given case frames, as shown in Figure 5. Firstly, a list of case frames q_c is input as a query. Next, associative examples q_e, which frequently occur in q_c are retrieved. Finally, associative case frames o_c, in which examples in q_e frequently occur, are retrieved and output.
Note that frequency of examples remarkably differs among case frames, so naive similarity measures such as tf.idf may give frequently occurred case frames with too large scores. To modify diversity of frequency, the pivoted document length normalization measure proposed by Singhal et al. (1996) is introduced. These measures have already been implemented as functions of the GETA: `WT_SMARTAW` and `WT_SMARTWA`.

Step 1
Given a list of i case frames $q_c = \{c_1, \cdots, c_i\}$, similarity score of each example e_y is calculated as follows:

$$\mathrm{sim}(e_y|q_c) = \frac{1}{\#\{q_c\}} \cdot \mathrm{idf}_c(e_y) \sum_{c_x \in q_c} \mathrm{tf}_{\mathrm{norm}}(e_y|c_x) \quad (1)$$

$$\mathrm{tf}_{\mathrm{norm}}(e_y|c_x) = \frac{1 + \log(\mathrm{tf}(e_y|c_x))}{1 + \log(\overline{\mathrm{tf}}(c_x))} \quad (2)$$

[5]http://geta.ex.nii.ac.jp/e/

$$\mathrm{idf}_c(e_y) = \log \frac{N_c}{\mathrm{df}_c(e_y)} \quad (3)$$

where $\mathrm{tf}(e_y|c_x)$ is the total # of occurrences of example e_y in case frame c_x, $\overline{\mathrm{tf}}(c_x)$ is the average # of example occurrences in c_x, N_c is the # of case frames in the matrix, $\mathrm{df}_c(e_y)$ is the # of unique case frames in which e_y occurs, and $\#\{q_c\}$ is the # of case frames in q_c. These values are easily calculated based on the matrix in Figure 4. The GETA is able to calculate the values efficiently, by ignoring zero factors.
Examples which have scores more than zero are output as associative examples $q_e = \{e_1, \cdots, e_j\}$, ranked on the descending order of the scores. If the number of examples which have scores more than zero exceeds the given limit (set to 50 experimentally), only the examples with largest scores are output.

Step 2
Given a list of j examples $q_e = \{e_1, \cdots, e_j\}$ with scores $\mathrm{sim}(e_y|q_c)$, similarity score of each case frame c'_z is calculated as follows:

$$\mathrm{sim}(c'_z|q_e) = \frac{1}{\mathrm{norm}(c'_z)} \sum_{e_y \in q_e} \mathrm{tf}_{\mathrm{norm}}(e_y|c'_z) \cdot \mathrm{sim}(e_y|q_c) \quad (4)$$

$$\mathrm{norm}(c'_z) = \overline{\mathrm{df}}_e + \textit{slope} \cdot (\mathrm{df}_e(c'_z) - \overline{\mathrm{df}}_e) \quad (5)$$

where $\mathrm{df}_e(c'_z)$ is the # of unique examples in case frame c'_z, and $\overline{\mathrm{df}}_e$ is the average # of unique examples per case frame. Currently, the parameter *slope* is set to 0.20.
Case frames which have scores more than zero are output as associative case frames $o_c = \{c'_1, \cdots, c'_k\}$, ranked on the descending order of the scores. If the number of case frames which have scores more than zero exceeds the given limit (set to 100 experimentally), only the case frames with the largest scores are output.

5. A Case Study: the JAL Pilot Reports

In order to estimate how the tool kit is effective in producing a domain ontology of contributing factors, we applied it to the JAL (Japan Airlines) pilot reports, which had been de-identified for data security and anonymity. And then, we experimentally produced a domain ontology of three contributing factors using the tool kit. How efficiently the tool kit helped us gather case frames related to the factors was evaluated.

5.1. Case Frame Construction from the Pilot Reports

The JAL pilot report collection (1992-2003) consists of about 6,600 reports, and it contains about 82,000 sentences. As a result of automatic case frame construction from the collection, 2,142 case frames are constructed. Figure 1 shows a subset of the case frames constructed from the collection.

5.2. Domain Ontology Production from the Pilot Reports

We experimentally produced a domain ontology of the following three contributing factors, by iterating the steps described in the Subsection 2.2.. In this experiment, we start

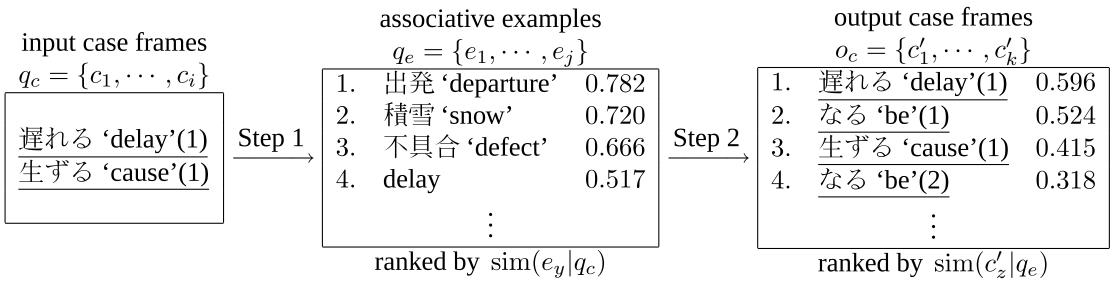

input case frames
$q_c = \{c_1, \cdots, c_i\}$
遅れる 'delay'(1)
生ずる 'cause'(1)
Step 1
associative examples
$q_e = \{e_1, \cdots, e_j\}$
1. 出発 'departure' 0.782
2. 積雪 'snow' 0.720
3. 不具合 'defect' 0.666
4. delay 0.517
ranked by $\mathrm{sim}(e_y|q_c)$
Step 2
output case frames
$o_c = \{c'_1, \cdots, c'_k\}$
1. 遅れる 'delay'(1) 0.596
2. なる 'be'(1) 0.524
3. 生ずる 'cause'(1) 0.415
4. なる 'be'(2) 0.318
ranked by $\mathrm{sim}(c'_z|q_e)$

Figure 5: Associative Retrieval of Case Frames

the iteration with one case frame related to the factors, and we add one case frame to the input box in each iteration.
The above part of Figure 6 shows some case frames related to the factors gathered by the tool kit. To improve readability, each case frame is replaced by a representative phrase of the case frame.

Pressure to keep schedule punctual
Usually, existence of pressure to keep schedule punctual is implied by expressions related to delays, but there are several expressions for delays in Japanese.
To find expressions related to the factor, we started with a case frame 出発が遅れる 'departure is delayed'. As a result, we collected 26 case frames as the domain ontology of the factor, in addition to the given case frame. Most of case frames in the ontology contains words *delay*, 遅れ 'delay' (noun), 遅れる 'delay' (verb), 遅延 'delay'. However, some case frames do not have such words. For example, a case frame ... へ ferry となる 'a deadhead flight for ... (airport)' also implies existence of pressure to keep schedule punctual, because deadhead flights are sometimes operated after a deep delay.

Successful resolution by aids
In order to make use of the pilot reports for reducing incidents and accidents, cases in which human errors and other failures have been successfully resolved by well-judged aids of other people (*e.g.,* passengers, flight attendants, air traffic controllers and mechanics) are also important. In general, such situations are implied by gratitude for the people.
To find expressions related to gratitude, we started with a case frame 協力に感謝する 'thank for cooperation'. As a result, we collected 29 case frames as the domain ontology for gratitude, in addition to the given case frame. Most of case frames in the ontology contains words お礼 'thanks' (noun), 感謝する 'thank' (verb) and 謝意 'gratitude', but some other case frames have other words such as 礼状 'letter for appreciation' and 敬意 'honor'.

Harassment by passengers
Sometimes harassment by passengers induces critical situations such as injury of crews and other passengers, ground turn-back, air turn-back, and so on.
To find expressions related to the factor, we started with a case frame 迷惑行為を行う 'do harassment', and we finally corrected 15 case frames as the domain ontology of the factor, in addition to the given case frame. Some of the case frames do not contain words related to 迷惑行為 'harassment': 搭乗を断る 'refuse boarding', 警告書を発行する 'issue a notice', and 誓約書を取得する 'put ... on oath'.

5.3. Efficiency of the Tool Kit

To estimate how efficiently the proposed method works, we examined the processes of domain ontology production described in Subsection 5.2., from the following aspects:

- How high the ranks of appropriate case frames (calculated by $\mathrm{sim}(c'_z|q_e)$) in each iteration? (Higher the ranks, easier we can find the next case frame related to the factors.)
- How many case frames are newly ranked as the best nth ones (*e.g.,* $n = 50$) throughout the entire iterations? (More case frames are newly ranked, better the user feedback iterations work.)

The bottom part of Figure 6 shows the results of the examination. The vertical axes mean the numbers of iterations, and the horizontal axes mean the ranks calculated by $\mathrm{sim}(c'_z|q_e)$, excluding case frames which have been already added in the input box, *i.e.,* q_c. And then, "o", "N" and "*" mean case frames related to the targeted contributing factors: "o" means a case frame which is added in the next iteration, "N" means a case frame which is firstly ranked as the best 50th ones, and "*" means other case frames related to the factors.
As the figure shows, "o" is mostly ranked as at most the 10th case frames, so we think that the similarity calculation scheme proposed in Section 4. worked well for the three contributing factors. In addition, since "N" is found even after the 10th iteration, we can conclude that the user feedback iteration of the tool kit is efficient.

6. Conclusion

In this paper, we proposed a novel approach to text mining, that is, domain ontology production based on automatically constructed case frames. The case frames can be used to mine implicit facts hidden in large corpora of specific domains, *e.g.,* contributing factors of incidents and accidents. Based on the capability of case frames, we implemented a domain ontology production tool kit using automatically constructed case frames. Since the case frame construction method does not depend on any specific domain, the tool kit will be applicable to any domain. We applied the tool

pressure to keep schedule punctual	successful resolution by aids	harassment by passengers
n 分遅れで出発となる ‘departure time is delayed for n minutes’, delay が生ずる ‘delay is caused’, 遅延が生ずる ‘delay is caused’, delay が予想される ‘delay is expected’, 遅れが重なる ‘delay occurs after another’, ...へ ferry となる ‘a deadhead flight for ... (airport)’	お礼を申し上げる ‘express thanks’, 礼状を送る ‘send a letter for appreciation’, スタッフの方に感謝する ‘thank for staffs’, 謝意を表す ‘express gratitude’, 努力に感謝する ‘thank for efforts’, 敬意を表す ‘give honor to’ 協力を得る ‘obtain cooperation’	迷惑行為が発生する ‘harassment occurs’, 迷惑行為に及ぶ ‘happen to do harassment’, 迷惑行為を続ける ‘continue to do harassment’, 搭乗を断る ‘refuse boarding’, 迷惑行為の事実を認める ‘admit the fact of harassment’, 警告書を発行する ‘issue a notice’, 誓約書を取得する ‘put ... on oath’

```
           10        20        30        40        50

 1    o    N            N NN   N     N   N     N N  N
 2       o  *NNNN**  *   *  *NN N NN    **N       N
 3    o**  * ** *** *N** *        **  *  **       N *
 4 o* *** *** *    *     * * *     *    * **  * *
 5 o******  *        *  *   * **   *      **  * *
 6 o***** *          * * *      * * * *      *
 7 o**** *        *      *  *    * ***   *       NNN
 8 o*** *     *          * *     **** *   *       * *
 9 o** *        *       *  *    **  ***   *         N
10 o* *        *        * *   * *  ** * *
11  o*        *        * *   * *  ** * *            N
12   *    o        *  *   ** * **     *     N * N  N
13  o            ** ***  ** *    *     *     *
14         o    ** **** *     *  **
15     o*  *** **    *          **
16    o**** * *            *    *                  N
17    o**** *           *    *                    *
18   o*** *         *          *                   *
19  o**  *          *        *                      *
20  o*      *     *           *
21  o     *      *         *
22         o *           *
23         o     *              N
24             o            *
25                   o                 N
26                       o
```

```
           10        20        30        40        50

 1 oN NNN  NNNNNNNNN   N NN  N                 NNN
 2 o ****  ********     * ***N NN          *
 3  o***  ********  *     *** ***
 4  o**  *****     ***  *** ***   *
 5  o*  *****       ***** ****   *
 6  o  *****       ****** ***  *
 7    o****       **** *****  *
 8    o***       * ******** *
 9    o**       N***** ****  *            N         N
10     o*  *** * *  *** **                    *
11     o*****  *    ** **            *          N
12    o** ***      ***  *               *    *      N
13  o *  ***  *    * **                  ** *
14 o    ****      *   **             **   *
15     o***      **    *           *   *   *
16     o**     *   *   *        * *         *
17     o*   *       * *         * *         *
18     o*           **     *   *   *        N     N
19     o *  *  N*      *      * *           *
20      o**  *    *       * *         *            N
21     o* *       *      * * *              *
22     o *         *     * * *                 *
23      o         *     *  * *             *
24                o   *   *  *          *
25     o * N       *        *
26     o*     *           *
27     o  *           *
28       o            *
29            o
```

```
           10        20        30        40        50

 1 oNNNNNNNN N   N
 2 o******* **
 3 o****** **
 4 o***** **
 5 o**** **
 6 o** ***
 7 o* ***
 8 o ***
 9  o**
10  o*         N     N
11  o         *      *
12      o     *     N          *
13        o      *          *
14              o           *
15              o
```

Figure 6: Domain Ontology Production from the JAL Pilot Reports

kit to the JAL pilot reports, and then we could derive a domain ontology of contributing factors in the civil aviation domain. In the ontology, a lot of interesting examples were contained. In addition, a brief examination of the domain ontology production process showed the efficiency of the tool kit. Now we are planning to evaluate the usefulness of the produced domain ontology, by applying the ontology to text mining of the pilot reports.

Currently, relations between keywords are not used in the tool kit, however, usage similarity between keywords can be also calculated using the matrix of case frames and example keywords. We have a plan to integrate relations of keywords and case frames into a domain ontology.

The tool kit is applicable only to Japanese texts now. However, if the automatic case frame construction is available for other languages, our method is expected to work for those languages, too.

7. References

Donald Hindle. 1990. Noun classification from predicate-argument structures. In *Proceedings of the 28th Annual Meeting of the Association for Computational Linguistics*, pages 268–275.

Daisuke Kawahara and Sadao Kurohashi. 2001. Japanese case frame construction by coupling the verb and its closest case component. In *Proceedings of First International Conference on Human Language Technology Research (HLT 2001)*, pages 204–210, San Diego, California.

Sadao Kurohashi and Makoto Nagao. 1994. A syntactic analysis method of long Japanese sentences based on the detection of conjunctive structures. *Computational Linguistics*, 20(4).

Amit Singhal, Chris Buckley, and Mandar Mitra. 1996. Pivoted document length normalization. In *Proceedings of SIGIR’96*.

Akihiko Takano, Yoshiki Niwa, Shingo Nishioka, Makoto Iwayama, Toru Hisamitsu, Osamu Imaichi, and Hirofumi Sakurai. 2000. Information access based on associative calculation. In *Lecture Notes in Computer Science*, volume 1963. Springer.

Akihiko Takano, Shingo Nishioka, Osamu Imaichi, Makoto Iwayama, Yoshiki Niwa, Toru Hisamitsu, Masakazu Fujio, Takenobu Tokunaga, Manabu Okumura, Hajime Mochizuki, and Tadashi Nomoto. 2001. Development of the generic association engine for processing large corpora. In *The FY 2001 Project Reports*, http://www.ipa.go.jp/NBP/13nendo/reports/. Information-technology Promotion Agency, Japan. (In Japanese).

Acknowledgment

The JAL pilot report collection was offered by the Flight Operations Engineering Section, Japan Airlines International Co., Ltd. We are grateful to Mr. Takashi Saito, Dr. Akira Terada, Mr. Toshihiro Sasaki and Mr. Ryuichi Toyota.

We also thank to Dr. Kunio Matsui, Mr. Takahiro Saito, and Mr. Isamu Watanabe at Fujitsu Laboratories Ltd. for their comments and intensive discussion with us.

The automatic case frame construction was implemented using the tools offered by Dr. Daisuke Kawahara.

The associative retrieval of case frames was implemented using the Generic Engine for Transposable Association (GETA), which is a product supported by the IPA (Information-technology Promotion Agency of Japan).