Systems and Computers in Japan, Vol. 38, No. 14, 2007
Translated from Denshi Joho Tsushin Gakkai Ronbunshi, Vol. J89-D, No. 11, November 2006, pp. 2470–2482

Domain Classification of Technical Terms Using the Web

Mitsuhiro Kida,[1] Masatsugu Tonoike,[2] Takehito Utsuro,[3] and Satoshi Sato[4]

[1]Nintendo Co., Ltd., Kyoto, 601-8116 Japan

[2]Nakai Research Center, Fuji Xerox Co., Ltd., Kanagawa, 259-0157 Japan

[3]Department of Intelligent Interaction Technologies, Graduate School of Systems and Information Engineering, University of Tsukuba, Tsukuba, 305-8573 Japan

[4]Department of Electrical Engineering and Computer Science, Graduate School of Engineering, Nagoya University, Nagoya, 464-8603 Japan

SUMMARY

This paper proposes a method of domain classification of technical terms using the Web. In the proposed method, it is assumed that, for a certain technical domain, a list of known technical terms of the domain is given. Technical documents of the domain are collected through the Web search engine, which are then used for generating a vector space model for the domain. The domain specificity of a target term is estimated according to the distribution of the domain of the sample pages of the target term. Experimental evaluation results show that the proposed method of domain classification of a technical term achieved mostly 90% precision/recall. We then apply this technique of estimating domain specificity of a term to the task of discovering novel technical terms that are not included in any existing lexicons of technical terms of the domain. Out of 1000 randomly selected candidates of technical terms per domain, we discovered about 100 to 200 novel technical terms. © 2007 Wiley Periodicals, Inc. Syst Comp Jpn, 38(14): 11–19, 2007; Published online in Wiley InterScience (www.interscience.wiley.com). DOI 10.1002/scj.20852

Key words: technical terms; Web; term extraction; domain classification.

1. Introduction

Lexicons of technical terms are one of the most important language resources both for human use and for computational research areas such as information retrieval and natural language processing. Among various research issues regarding technical terms, full-/semi-automatic compilation of technical term lexicons is one of the central issues. In various research fields, novel technologies are invented every year, and related research areas around such novel technologies keep growing. Along with such invention of technologies, novel technical terms are created year by year. Considering such a situation, a huge cost is required for manually compiling lexicons of technical terms for hundreds of thousands of technical domains. Therefore, it is inevitable that of a technique of full-/semi-automatic compilation of technical term lexicons for various technical domains will be invented.

The whole task of compiling a technical term lexicon can be roughly decomposed into two subprocesses:

(1) collecting candidates of technical terms of a technical domain and

(2) judging whether each candidate is actually a technical term of the target technical domain.

The technique of the first subprocess is closely related to research on automatic term recognition, and has been relatively well studied so far (e.g., Ref. 7). On the other hand, the technique of the second subprocess has not been studied well. Exceptional cases are works such as Refs. 1 and 2, where their techniques are mainly based on the tendency of technical terms appearing in technical documents of limited domains rather than in documents of daily use such as newspaper and magazine articles. Although the underlying idea of those previous works is very interesting, those works are quite limited in that they require the existence of a certain amount of technical domain corpus. It is not practical to manually collect technical domain corpus for hundreds of thousands of technical domains. Therefore, as for the second subprocess here, it is very important to invent a technique for automatically classifying the domain of a technical term.

Based on this observation, among several key issues regarding the second subprocess above, this paper mainly focuses on the issue of estimating the domain specificity of a term. In this paper, supposing that a target technical term and a technical domain are given, we propose a technique of automatically estimating the specificity of the target term with respect to the target domain. Here, the domain specificity of the term is judged among the following three levels:

(i) the term mostly appears in the target domain,
(ii) the term generally appears in the target domain as well as in other domains,
(iii) the term generally does not appear in the target domain.

The key idea of the proposed technique is as follows [8, 9]. In the proposed technique, we assume that sample technical terms of the target domain are available. Using such sample terms with search engine queries, we first collect a corpus of the target domain from the Web. In a similar way, we also collect sample pages that include the target term from the Web. Then, the similarities of the contents of the documents are measured between the corpus of the target domain and each of the sample pages that include the target term. Finally, the domain specificity of the target term is estimated according to the distribution of the domain of those sample pages.

Figure 1 illustrates a rough idea of this technique. Among the three example (Japanese) terms, the first term ("*impedance characteristic*") mostly appears in the documents of the "*electric engineering*" domain on the Web. In the case of the second term ("*electromagnetism*"), about half of the sample pages collected from the Web can be regarded as in the "*electric engineering*" domain, while the rest are not. On the other hand, in the case of the last term ("*response characteristic*"), only a few of the sample pages can be regarded as in the "*electric engineering*" domain. In our technique, such difference of the distribution can be

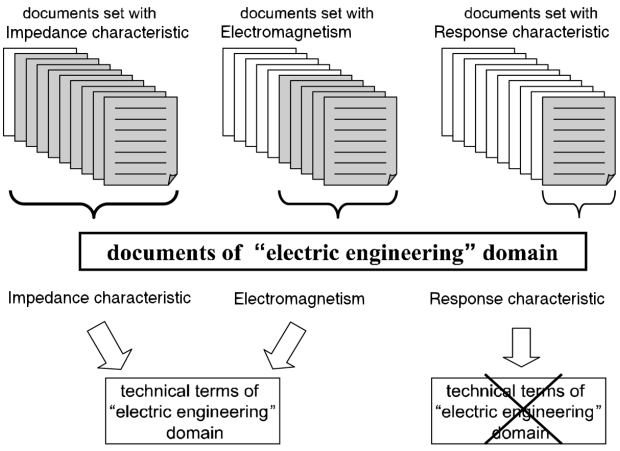

Fig. 1. Degree of specificity of a term based on the domain of the documents (example terms: "*impedance characteristic*," "*electromagnetism*," and "*response characteristic*").

easily identified, and the domain specificities of those three terms are estimated.

As experimental evaluation, we first evaluate the proposed technique of estimating domain specificity of a term using manually constructed development and evaluation term sets, where we achieved mostly 90% precision/recall. Furthermore, in this paper, we present the results of applying this technique of estimating domain specificity of a term to the task of discovering novel technical terms that are not included in any existing lexicons of technical terms of the domain. Candidates of technical terms are first collected from the Web corpus of the target domain. Then, about 70 to 80% of those candidates are excluded by roughly judging the domain of their constituent words. Finally, out of 1000 randomly selected candidates of technical terms per domain, we discovered about 100 to 200 novel technical terms that are not included in any existing lexicons of the domain, where we achieved about 75% precision and 80% recall.

2. Domain Specificity Estimation of Technical Terms Using Documents Collected from the Web

2.1. Outline

As introduced in the previous section, the underlying purpose of this paper is to invent a technique for automatically classifying the domain of a technical term. More specifically, this paper mainly focuses on the issue of estimating the domain specificity of a term t with respect to a

domain C, supposing that the term t and the domain C are given.

Generally speaking, the coarsest-grained classification of domain specificity of a term is binary classification, namely, the class of terms that are used in a certain technical domain versus the class of terms that are not used in a certain technical domain. In this paper, we further classify the degree $g(t, C)$ of the domain specificity into the following three levels:

$$g(t, C) = \begin{cases} + & (t \text{ mostly appears in the documents of the domain } C) \\ \pm & (t \text{ generally appears in the documents of the domain } C \text{ as well as in those of the domains other than } C) \\ - & (t \text{ generally does not appear in the documents of the domain } C) \end{cases}$$

(When we simply classify domain specificity of a term into two classes with the coarsest-grained binary classification above, we regard those with domain specificity "+" or "±" as those that are used in the domain, and those with domain specificity "–" as those that are not used in the domain.)

The input and output of the process of domain specificity estimation of a term t with respect to the domain C are given below:

input	target term t for domain specificity estimation, set T_C of sample terms of the domain C
output	domain specificity $g(t, C)$ of t with respect to C

The process of domain specificity estimation of a term is illustrated in Fig. 2, where the whole process can be decomposed into two subprocesses:

(a) that of constructing the corpus D_C of the domain C and

(b) that of estimating the specificity of a term t with respect to the domain C.

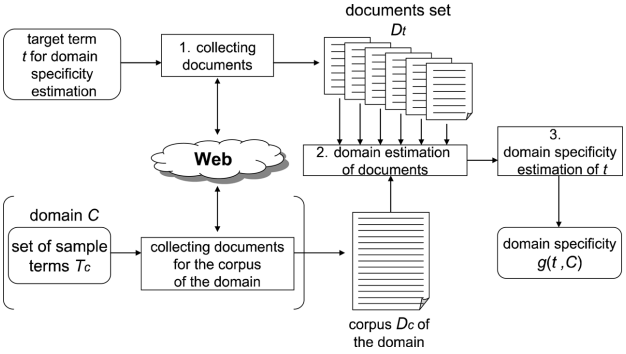

Fig. 2. Domain specificity estimation of terms based on Web documents.

In the process of domain specificity estimation, the domain of documents including the target term t is estimated, and the domain specificity of t is judged according to the distribution of the domains of the documents including t. The details of those two subprocesses are described below.

2.2. Constructing the corpus of the domain

When constructing the corpus D_C of the domain C using the set T_C of sample terms of the domain C, first, for each term t in the set T_C, we collect into a set D_t the top 100 pages obtained from search engine queries that include the term t.* The search engine queries here are designed so that documents that describe the technical term t are ranked high. When constructing a corpus of the Japanese language, the search engine "goo"† is used. The specific queries that are used in this search engine are phrases with topic-marking postpositional particles such as "t-toha," "t-toiu," "t-wa," and an adnominal phrase "t-no," and "t."

Then, union of the sets D_t for each t is constructed and denoted as $D(T_C)$:

$$D(T_C) = \bigcup_{t \in T_C} D_t$$

Finally, in order to exclude noise texts from the set $D(T_C)$, the documents in the set $D(T_C)$ are ranked according to the number of sample terms (of the set T_C) that are included in each document. Through a preliminary experiment, we decided here that it is enough to keep the top 500 documents, and regard them as the corpus D_C of the domain C.

2.3. Domain specificity estimation of technical terms

Given the corpus D_C of the domain C, domain specificity of a term t with respect to a domain C is estimated through the following three steps:

Step 1: Collecting documents that include the term t from the Web, and constructing the set D_t of those documents.

Step 2: For each document in the set D_t, estimating its domain by measuring similarity against the corpus D_C of the domain C. Then, given a certain lower bound L of document similarity, from D, extracting documents with large enough similarity values into a set $D_t(C, L)$.

*Related techniques for automatically constructing the corpus of the domain using the sample terms of the domain include those presented in Refs. 3 and 6. We are planning to evaluate the performance of those related techniques and compare them with the one employed in this paper.

†2 http://www.goo.ne.jp/

Step 3: Estimating the domain specificity $g(t, C)$ of t using the document set $D_t(C, L)$ constructed in step 2.

Details of the three steps are given below.

2.3.1. Collecting Web documents including the target term

For each target term t, documents that include t are collected from the Web. According to a procedure that is similar to that of constructing the corpus of the domain C described in Section 2.2, the top 100 pages obtained with search engine queries are collected into a set D_t.

2.3.2. Domain estimation of documents

For each document in the set D_t, its domain is estimated by measuring similarity against the corpus D_C of the domain C. Then, given a certain lower bound L of document similarity, documents with large enough similarity values are extracted from D_t into the set $D_t(C, L)$.

In the process of document similarity calculation, we simply employ a standard vector space model,* where a document vector is constructed, after removing 153 stop words, as a vector of frequencies of content words such as nouns and verbs in a document. Here, the corpus D_C of the domain C is regarded as a document d_C, and a document vector $dv(d_C)$ is constructed. For each document d_t in the set D_t, a document vector $dv(d_t)$ is also constructed. Then, the cosine similarity between $dv(d_t)$ and $dv(d_C)$ is calculated and is defined as the similarity $sim(d_t, D_C)$ between the document d_t and the corpus D_C of the domain C:

$$sim(d_t, D_C) = sim(d_t, d_C) = \cos(dv(d_t), dv(d_C)) = \frac{dv(d_t) \cdot dv(d_C)}{|dv(d_t)||dv(d_C)|}$$

Finally, suppose that a certain lower bound L of document similarity is given, documents d_t with the similarity value $sim(d_t, D_C)$ above or equal to L are regarded as those of the domain C, and are collected into a set $D_t(C, L)$:

$$D_t(C, L) = \{d_t | sim(d_t, D_C) \geq L\}$$

In the experimental evaluation of Section 2.4, the lower bound L is determined using a development term set for tuning various parameters of the whole process of estimating domain specificity of technical terms.

*An alternative here is to apply supervised document classification techniques such as those based on machine learning technologies (e.g., Ref. 4). In particular, since it is not easy to collect negative data in the task of domain specificity estimation of a term, it seems very interesting to apply recently developed techniques without labeled negative data (e.g., Ref. 5).

2.3.3. Domain specificity estimation of a term

The domain specificity of the term t with respect to the domain C is estimated using the document sets D_t and $D_t(C, L)$. Here, this is done by simply calculating the following ratio r_L of the numbers of documents within the two sets:

$$r_L = \frac{|D_t(C, L)|}{|D_t|}$$

Then, by introducing the two thresholds $a(\pm)$ and $a(+)$ for the ratio r_L, the specificity $g(t, C)$ of t is estimated with the following three levels:

$$g(t, C) = \begin{cases} + & (a(+) \leq r_L) \\ \pm & (a(\pm) \leq r_L < a(+)) \\ - & (r_L < a(\pm)) \end{cases}$$

In the experimental evaluation of Section 2.4, as in the case of the lower bound L of the document similarity, the two thresholds $a(\pm)$ and $a(+)$ are also determined using the development term set mentioned above.

2.4. Experimental evaluation

We evaluate the proposed method with five sample domains, namely, "*electric engineering*," "*optics*," "*aerospace engineering*," "*nucleonics*," and "*astronomy*." For each domain C of those five domains, the set T_C of sample (Japanese) terms is constructed by randomly selecting 100 terms* from an existing (Japanese) lexicon of technical terms for human use. For each of the five domains, we then manually constructed the development term set T_{dev} and the evaluation term set T_{eva}, each of which has 100 terms (those with frequency more than or equal to five, and hits of the search engine within 100 to 10,000), respectively. For each of the domain specificity "+," "±," and "–," Table 1 lists the number of terms of the class. In our experimental evaluation, the development term set T_{dev} is used for tuning the lower bound L of the document similarity, as well as the two thresholds $a(\pm)$ and $a(+)$, where those parameter values are determined so as to maximize the F score ($\alpha = 0.75$). Here, we chose the value of the weight α as 0.75, since we prefer precision to recall in our application such as automatic technical term collection, where automatically collected technical terms are recursively utilized as seed technical terms in the later steps of a bootstrapping process.

*Through a preliminary experiment, we conclude that it is not necessary to start with the set T_C of sample terms which has more than 100 sample terms. The number of minimum requirement for the size of T_C varies according to domains.

Table 1. Number of terms for experimental evaluation

	number of terms for each degree of domain specificity					
	development set T_{dev}			evaluation set T_{eva}		
	+	±	-	+	±	-
electric engineering	43	14	43	48	20	32
optics	35	15	50	40	24	36
aerospace engineering	39	10	51	36	24	40
nucleonics	22	24	54	34	28	38
astronomy	41	12	47	35	15	50

Table 2. Precision/recall of domain specificity estimation

(a) with threshold a(±)

	L	a(±)	development set T_{dev}		evaluation set T_{eva}	
			precision	recall	precision	recall
electric engineering	0.2	0.4	0.96(54/56)	0.95(54/57)	0.95(59/62)	0.87(59/68)
optics	0.2	0.4	0.94(49/52)	0.98(49/50)	1.00(60/60)	0.94(60/64
aerospace engineering	0.2	0.4	0.94(42/44)	0.86(42/49)	0.79(54/68)	0.90(54/60)
nucleonics	0.25	0.2	0.92(36/39)	0.78(36/46)	0.95(60/63)	0.97(60/62)
astronomy	0.15	0.4	0.96(51/53)	0.96(51/53)	0.86(48/56)	0.96(48/50)

(b) with threshold a(+)

	L	a(+)	development set T_{dev}		evaluation set T_{eva}	
			precision	recall	precision	recall
electric engineering	0.2	0.7	0.97(32/33)	0.74(32/43)	0.92(24/26)	0.50(24/48)
optics	0.2	0.7	0.83(20/24)	0.57(20/35)	0.82(23/28)	0.58(23/40)
aerospace engineering	0.2	0.5	0.90(28/31)	0.72(28/39)	0.53(27/51)	0.75(27/36)
nucleonics	0.25	0.3	.55(18/33)	0.82(18/22)	0.57(32/56)	0.94(32/34)
astronomy	0.15	0.7	0.89(34/38)	0.83(34/41)	0.87(33/38)	0.94(33/35)

$$\text{F score} = \frac{1}{\alpha(1/\text{precision}) + (1-\alpha)(1/\text{recall})}$$

Parameter values as well as experimental evaluation results are summarized in Table 2. Generally speaking, the task of discriminating the terms with domain specificity "+" against the rest is much harder than that of discriminating those with "+" and "±" against those with "–." In Table 2(b), especially, the results with the domains "*aerospace engineering*" and "*nucleonics*" have lower precision values than other domains. Each of these two domains has another closely related domain (e.g., "*military*" for "*aerospace engineering*," and "*radiation medicine*" for "*nucleonics*"), where technical terms of these two domains tend to appear also in the documents of such closely related domains. Most of the errors are caused by the existence of those close domains.

3. Collecting Novel Technical Terms of a Domain from the Web

3.1. Procedure

This section illustrates how to apply the technique of domain specificity estimation of technical terms to the task of discovering novel technical terms that are not included in any existing lexicons of technical terms of the domain. First, as shown in Fig. 3, from the corpus D_C of the domain C, candidates of technical terms are collected. In the case of the Japanese language, as candidates of novel technical terms, we collect compound nouns with frequency counts five or more, consisting of more than one noun. Here, we collect compound nouns which are not included in any existing lexicons of technical terms of the domain. Then, after excluding terms which do not share constituent nouns against the sample terms of the given set T_C, the domain specificity of the remaining terms are automatically estimated. Finally, we regard terms with domain specificity "+" or "a(±)" as those that are used in the domain, and collect them into the set $T_{web,C}$.

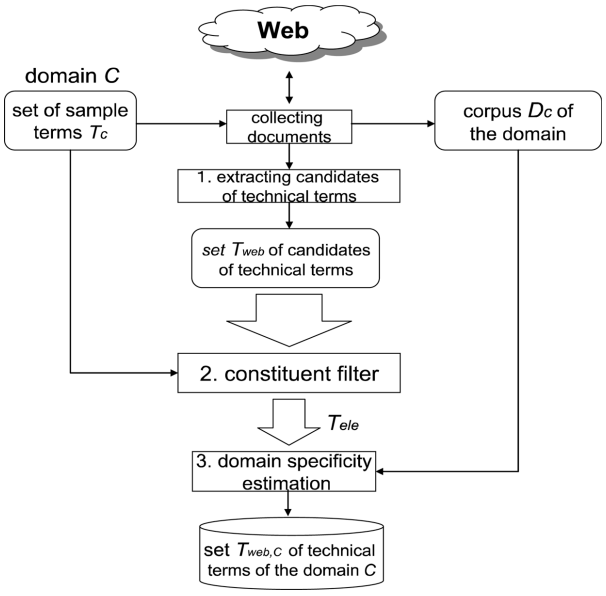

Fig. 3. Collecting novel technical terms of a domain from the Web.

3.2. Experimental evaluation

Table 3 compares the numbers of candidates of novel technical terms collected from the Web, with those after excluding terms which do not share constituent nouns against the sample terms of the given set T_C. As shown in the table, about 70 to 80% of the candidates are excluded, while the rate of technical terms within the remaining candidates increased. This result clearly shows the effectiveness of the constituent noun filtering technique in reducing the computational time of discovering fixed number of novel technical terms. Then, per domain, we randomly select 1000 of those remaining candidates, and estimate their domain specificity by the proposed method. After manually judging the domain specificity of those 1000 terms, we measure the precision/recall of the proposed method as in Table 4, where we achieved about 75% precision and 80% recall. Here, however, as candidates of technical terms, we simply collect compound nouns, where sometimes their term unit is not correct since the technical term candidate could be with a certain prefix or suffix. Considering this fact, Table 4 also gives the term unit correct rate for those with domain specificity "+" or "±." Finally, taking this term unit correct rate into account, we can conclude that, out of the 1000 candidates, we discovered about 100 to 200 novel technical terms that are not included in any existing lexicons of the domain. This result clearly supports the effectiveness of the proposed technique for the purpose of full-/semi-automatic compilation of technical term lexicons.

4. Concluding Remarks

This paper proposed a method of domain specificity estimation of technical terms using the Web. In the proposed method, it is assumed that, for a certain technical domain, a list of known technical terms of the domain is given. Technical documents of the domain are collected through the Web search engine, which are then used for generating a vector space model for the domain. The domain specificity of a target term is estimated according to

Table 3. Changes in number of technical term candidates with constituent filter

	before filtering		after filtering	
	# of candidates	# of tech. terms (estimated) (%)	# of candidates	# of tech. terms (estimated) (%)
electric engineering	24,460	1,272 (5.2)	6,623	848 (12.8)
optics	29,090	1,047 (3.6)	6,985	866 (12.4)
aerospace engineering	41,279	660 (1.6)	6,364	458 (7.2)
nucleonics	40,439	890 (2.2)	10,834	650 (6.0)
astronomy	29,240	1,170 (4.0)	5,491	659 (12.0)

Table 4. Precision/recall of collecting novel technical terms

(a) with threshold $a(\pm)$

	precision	recall	term unit correct rate
electric engineering	0.754(399/529)	0.828(399/482)	0.393(157/399)
optics	0.766(454/593)	0.875(454/519)	0.368(167/454)
aerospace engineering	0.797(408/512)	0.739(408/552)	0.402(164/408)
nucleonics	0.685(470/686)	0.953(470/493)	0.377(177/470)
astronomy	0.747(480/643)	0.945(480/508)	0.475(228/480)

(b) with threshold $a(+)$

	precision	recall	term unit correct rate
electric engineering	0.697(168/241)	0.853(168/197)	0.494(83/168)
optics	0.743(234/315)	0.932(234/251)	0.453(106/234)
aerospace engineering	0.666(277/416)	0.936(277/296)	0.502(139/277)
nucleonics	0.580(362/624)	0.981(362/369)	0.406(147/362)
astronomy	0.763(350/459)	0.888(350/394))	0.520(182/350

the distribution of the domain of the sample pages of the target term. Experimental evaluation results showed that the proposed method achieved mostly 90% precision/recall. We then applied this technique of estimating domain specificity of a term to the task of discovering novel technical terms that are not included in any existing lexicons of technical terms of the domain. Out of 1000 randomly selected candidates of technical terms per domain, we discovered about 100 to 200 novel technical terms.

Related techniques on domain specificity estimation of a term include the one based on straightforward application of similarity calculation of language models between the domain corpus and the target term. Although this technique seems mathematically well defined, it is weak in that it assumes a single language model per target term. In particular, when a target term appears in the documents of more than one domain, the technique proposed in this paper seems to be advantageous, since it independently estimates the domain of each individual document including the target term.

REFERENCES

1. Chung TM. A corpus comparison approach for terminology extraction. Terminology 2004;9:221–246.
2. Drouin P. Term extraction using non-technical corpora as a point of leverage. Terminology 2003;9:99–117.
3. Huang C-C, Lin K-M, Chien L-F. Automatic training corpora acquisition through Web mining. Proceedings of IEEE/WIC/ACM International Conference on Web Intelligence, p 193–199, 2005.
4. Joachims T. Learning to classify text using support vector machines: Methods, theory, and algorithms. Springer-Verlag; 2002.
5. Liu B, Dai Y, Li X, Lee WS, Yu PS. Building text classifiers using positive and unlabeled examples. Proceedings of the 3rd IEEE International Conference on Data Mining, p 179–186, 2003.
6. Liu B, Li X, Lee WS, Yu PS. Text classification by labeling words. Proceedings of the 19th AAAI, p 425–430, 2004.
7. Nakagawa H, Mori T. Automatic term recognition based on statistics of compound nouns and their components. Terminology 2003;9:201–219.
8. Utsuro T, Kida M, Tonoike M, Sato S. Towards automatic domain classification of technical terms: Estimating domain specificity of a term using the Web. In: Ng HT, Leong M-K, Kan M-Y, Ji DH (editors). Information retrieval technology: Third Asia Information Retrieval Symposium. AIRS 2006, Lecture Notes in Computer Science Vol. 4182, p 633–641. Springer; 2006.
9. Utsuro T, Kida M, Tonoike M, Sato S. In: Matsumoto Y, Sproat R, Wong K-F, Zhang M (editors). Computer processing of Oriental languages: Beyond the Orient: The research challenges ahead. Lecture Notes in Artificial Intelligence Vol. 4285, p 173–180. Springer; 2006.

AUTHORS (from left to right)

Mitsuhiro Kida received his B.E. degree in electrical engineering and M.Inf. degree from Kyoto University in 2004 and 2006. Since 2006, he has been involved in development activities of information technologies at Nintendo Co., Ltd.

Masatsugu Tonoike received his B.E. degree in information engineering and M.Inf. and D.Inf. degrees from Kyoto University in 2001, 2003, and 2007. Since then, he has been involved in research and development activities of information technologies at Fuji Xerox Co., Ltd.

AUTHORS (continued) (from left to right)


Takehito Utsuro (member) received his B.E., M.E., and D.Eng. degrees in electrical engineering from Kyoto University in 1989, 1991, and 1994. He was an instructor at the Graduate School of Information Science, Nara Institute of Science and Technology, from 1994 to 2000, a lecturer in the Department of Information and Computer Sciences, Toyohashi University of Technology, from 2000 to 2002, and a lecturer in the Department of Intelligence Science and Technology, Graduate School of Informatics of Kyoto University, from 2003 to 2006. He has been an associate professor in the Department of Intelligent Interaction Technologies, Graduate School of Systems and Information Engineering, University of Tsukuba, since 2006. From 1999 to 2000, he was a visiting scholar in the Department of Computer Science of Johns Hopkins University. His professional interests lie in natural language processing, information retrieval, text mining from the Web, machine learning, spoken language processing, and artificial intelligence.

Satoshi Sato is a professor in the Department of Electrical Engineering and Computer Science at Nagoya University. His recent work in natural language processing focuses on automatic paraphrasing, controlled language, and automatic terminology construction. He received his B.E., M.E., and D.Eng. degrees from Kyoto University in 1983, 1985, and 1992.

)