Toward an Enhancement of Textual Database Retrieval By using NLP Techniques*

Asanee Kawtrakul[1], Frederic Andres[2], Kinji Ono[2],
Chaiwat Ketsuwan[1], Nattakan Pengphon[1],
ak@beethoven.cpe.ku.ac.th , {andres,ono}@rd.nacsis.ac.jp

(1) NAiST[(1),] Computer Engineering Dept, Kasetsart University, Bangkok, Thailand
(2) NACSIS[(2)], Center of Excellence of the Ministry of Education, Tokyo, Japan

ABSTRACT: Improvements in hardware, communication technology and database have led to the explosion of multimedia information repositories. In order to provide the quality of information retrieval and the quality of services, it is necessary to consider both retrieval techniques and database architecture.

This paper presents the project named VLSHDS-Very Large Scale Hypermedia Delivery System. The quality of textual information search is enhanced by using NLP techniques. The quality of service over a large-scale network is provided by using AHYDS-Active HYpermedia Delivery System-framework.

KEY WORDS: Information Retrieval, Textual Database Retrieval, Multi-level indexing, Document Classification, Very Large Scale Hypermedia Delivery System, Natural Language Processing

บทคัดย่อ: การพัฒนาเทคโนโลยีฮาร์ดแวร์ เทคโนโลยีสื่อสาร และฐานข้อมูลได้นำไปสู่การเติบโตอย่างรวดเร็วของการจัดเก็บข้อมูลแบบหลายสื่อ เพื่อให้การบริการข้อมูลและการสืบค้นข้อมูลมีคุณภาพและประสิทธิภาพ เราจำเป็นต้องพิจารณาทั้งทางด้านเทคนิคการสืบค้นข้อมูลและสถาปัตยกรรมฐานข้อมูล

บทความฉบับนี้นำเสนอผลงานวิจัยภายใต้โครงการที่ชื่อว่า ระบบจัดส่งข้อมูลหลายสื่อขนาดใหญ่ (VLSHDS) ด้วยเทคนิคการประมวลภาษาธรรมชาติในระดับคำ และระดับวลี สามารถยกระดับคุณภาพของการสืบค้นข้อมูล ด้วยเทคโนโลยีของระบบจัดส่งข้อมูลหลายสื่อแบบแอคตีฟ สามารถเพิ่มคุณภาพการให้บริการข้อมูล

คำสำคัญ : การสืบค้นข้อสนเทศ การสืบค้นข้อมูลเอกสาร การสร้างดัชนีหลายระดับ การแยกประเภทเอกสาร ระบบจัดส่งข้อมูลหลายสื่อขนาดใหญ่

- Kasetsart University Research and Development Institute (KURDI), Kasetsart University, Thailand and National Center have granted this Project for Science Information Systems (NACSIS), Center of Excellence of the Ministry of Education, JAPAN and National Electronics and Computer Technology Center (NECTEC).
- This article is a reprint of the article appeared in the Proceedings of NECTEC Annual Conference 2000 : ECTI Technologies for New Economies, June 2000, pp. 280-290. This paper wins a best paper award in category of "the most impact to Thai society".

1. Introduction

Improvements in hardware, communication technology and database engines had led to the expansion of challenging interactive multimedia applications and services. Typical examples of applications include on-line news, digital libraries and web-based information involving multi-dimension multimedia document repositories. These systems combine various media content with hyperlink structures for user query or navigation. Most of them store contents inside the database systems supporting extenders in order to add application data types with their access methods. Moreover, there is no vertical integration between application plug-ins and the database kernel itself. This limitation is an underlying reason for further improvements [6,8,16,17,18]. AHYDS-The Active HYpermedia Delivery System is one of a new wave of database kernels [4,7,15] that facilitates the access to multimedia documents according to the user's requirement and application's features over a wide spectrum of networks and media [1].

The VLSHDS-Very Large Scale Hypermedia Delivery System is the project between NACSIS and NAiST [2,10] which is aimed to integrate both the quality of data management service and the quality of textual information retrieval. The VLSHDS platform is , then, based on AHYSD which provides a framework for open data delivery service, communication service, query execution service and supervision service. The quality of full text retrieval services has been enhanced in both precision and recall by integrating NLP techniques for document and query processing.
Section 2 gives an overview of the VLSHDS. The implementation of document processing, query processing and retrieving processing are described in section 3, 4 and 5 respectively. Section 6 gives the conclusion and briefs the next step of the project.

2. An Overview of the Very Large Scale Hypermedia Delivery Systems

The key architectural components in the VLSHDS platform used as textual database platform is shown in Figure 1. The system consists of a client/server three tiers architecture. At Client side, queries are sent to the server by using the AHYDS communication support [11]. At the server side, there are three main components: Document Processing, Query Processing and Retrieving Processing. The Document Processing based on the Extended Binary Graph (EBG) structure provides multilevel indices and document category as document representation. The Query Processing provides query expansion based on query guide. The Retrieval Processing computes the similarity between queries and documents and returns a set of retrieved documents with the similarity scores.

3. The Role of NLP in Document Processing

To enhance the performance of full text retrieval service, good representation of each document should be provided, i.e., multi-level indices and document category. Multi-level indices will increase the retrieval recall without the degradation of the system precision and document category will be used for pruning irrelevant document or increase precision while decreasing the searching time.

The primary problem in computing multi-level indices and category as document representation is a linguistic problem. The problems frequently found, especially in Thai documents, are lexical unit extraction including unknown word, phrase variation, loan words, acronym, synonym and definite anaphora.

Accordingly, to be more successful, NLP components, i.e., morphological analysis and shallow parsing should be integrated with statistical based indexing and categorizing

3.1 The Architecture of NLP based Document Processing

Figure 2 shows the overview of Thai document processing. There are two main steps: multilevel indexing and document categorizing. Each document will be represented as

$$D_i = <I_p, I_t, I_c, C_i>$$

Where I_p, I_t, I_c are the set of indices in phrase, single term and conceptual level, respectively

Ci is the category of an document i-th

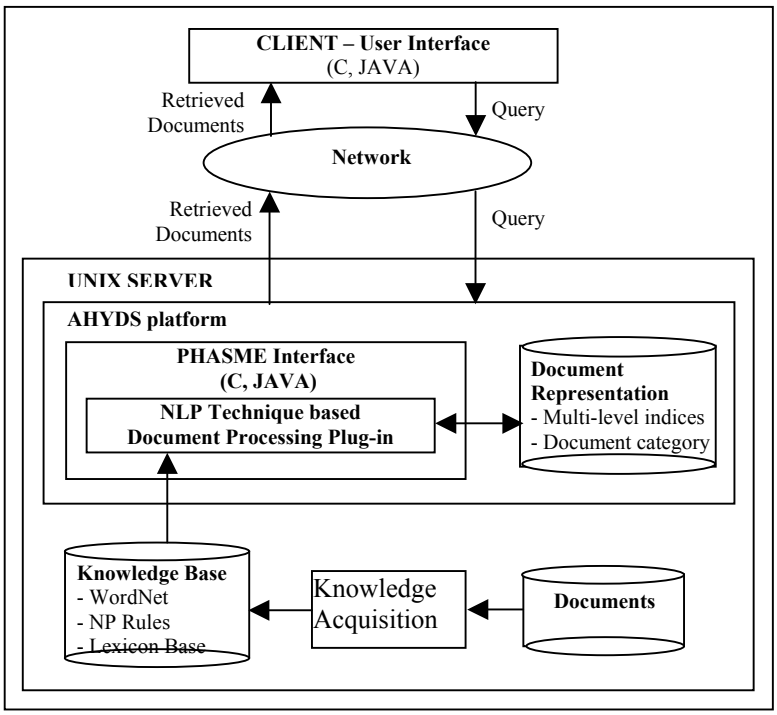

***Figure 1**: The Architecture of the VLSHDS Platform for Textual Document Retrieval*

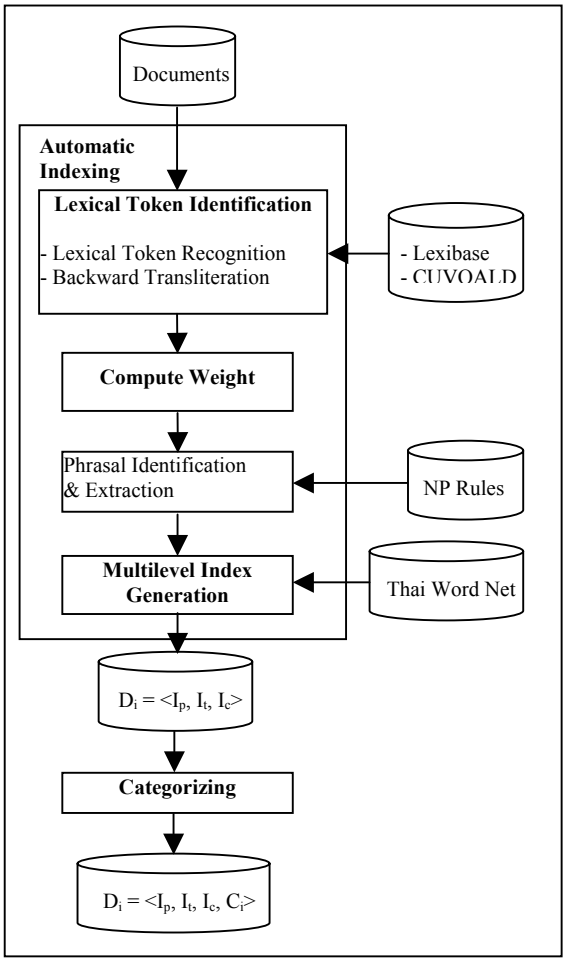

***Figure 2**: Overview of Thai document processing*

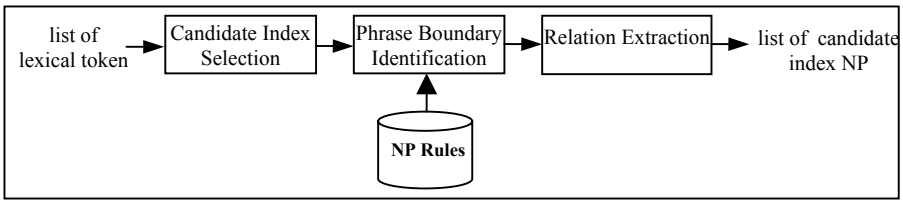

Figure 3: Phrase Identification and Relation Extraction

NP <- cn + cn + (cn)	กล้วยไม้ป่า
NP <- cn + {cn, pn}	ประเทศไทย
NP <- cn + v + (cn)	วิทยุกระจายเสียง
NP <- cn + (cn) + mod	น้ำปลาหวาน
NP <- cn + prep + cn	คุณค่าทางอาหาร
NP <- cn + v + v	ค่าใช้จ่าย
NP <- cn + num + cl	สัตว์สองเท้า
NP <- cn + NOM	อุตสาหกรรมการกลั่นสุรา
NOM <- prefix + vp	การกลั่นสุรา

***Figure 4**: Noun phrase rules*

3.2 Automatic Multilevel Indexing

As shown in Figure 2, automatic multilevel indexing consists of three modules: lexical token identification, phrase identification with relation extraction, and multilevel index generation. Each module accesses different linguistic knowledge bases stored inside the EBG data structure.

3.2.1 Lexical Token Identification

The role of lexical token identification is to determine word boundaries, and to recognize unknown words, such as proper names and loan words, i.e., technical terms in Thai orthography. Based on [12], lexical tokens are segmented and tagged with part of speech. Based on [13], loan word will be solved.

3.2.2 Phrase Identification and Relation Extraction

In order to compute multilevel indices, phrase identification and relation extraction are needed. The relation between terms in the phrase will be extracted in order to define indices in single term level and conceptual level. There are two kinds of relations: head-modifier and non-head-modifier (or compound noun). If the relation is head-modifier, the head part will be the single term-level index while the modifier will not be used as index. If the relation is compound noun, there is no single term-level index. The conceptual level indices, then, will be produced from the head of phrase or compound noun by accessing Thai wordnet.

The problems of this step are that (1) how to identify phrase without deep parsing for the whole text, (2) how to distinguish between noun phrase and sentence which may have the same pattern and (3) how to extract the relation between terms in phrase.

To solve the problems mentioned above, the algorithm of phrase identification and relation extraction consists of three main steps (see Figure 3):

The first step is the candidate index selection which is based on Salton's weighting formula [9], then, the second step is the phrase boundary identification by using statistical based NLP technique. This step will find phrase boundary for a set of candidate indices(provided from the first step) by using NP rules (see Figure 4)

At this step, we can describe as 4-tuple:

$\{T, N, R, NP\}$

where

T is the set of candidate terms which have weight $w_i > \theta$ (θ is a threshold)

N is the set of non-terminal

R is the set of rules in the grammar

NP is the starting symbol

The third step is the relation extraction. After the boundary of phrase(s) is identified, we need to compute the relation between a set of words in the phrase in order to find whether that NP is head-modifier NP or compound. If the frequency of each word of candidate NP has the same frequency (see Figure 5) then

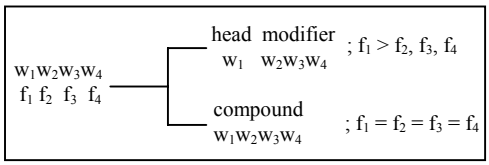

***Figure 5**: Relation Extraction*

relation of that NP is compound noun, otherwise relation of that NP is head-modifier pair. Head is the term(s) with highest frequency. Modifier is the term(s) with lower frequency.

The detail of the algorithm is given in Annex 1.

3.2.3 Multilevel index generation

At this step, each document D_i is summarized and represented in the EBG data structure as a vector of numeric weights, i.e.:

$$D_i =< I_{p_i}, I_{t_i}, I_{c_i} >$$

where

$$I_{p_i} =< W_{p1_i}, W_{p2_i}, ..., W_{p3_i} >$$
$$I_{t_i} =< W_{t1_i}, W_{t2_i}, ..., W_{t3_i} >$$
$$I_{c_i} =< W_{c1_i}, W_{c2_i}, ..., W_{c3_i} >$$

Phrasal level indices (I_p) consist of set of the phrases extracted by using noun phrase rules. Single term level indices (I_t) are the head of each index token in the phrasal level. Conceptual level indices (I_c) are the semantic concepts of each single term level index, given in Lexibase [14]. For example, the document concerns about มะนาว (lemon), may keep "มะนาวไข่" (A kind of lemon) as phrasal level and keep "มะนาว" (Lemon) as single term level and "พืช" (Plant) as conceptual level.

Figure 6 shows the process of Multilevel Index Generation consisting of phrase level, single term level and conceptual level for each document.

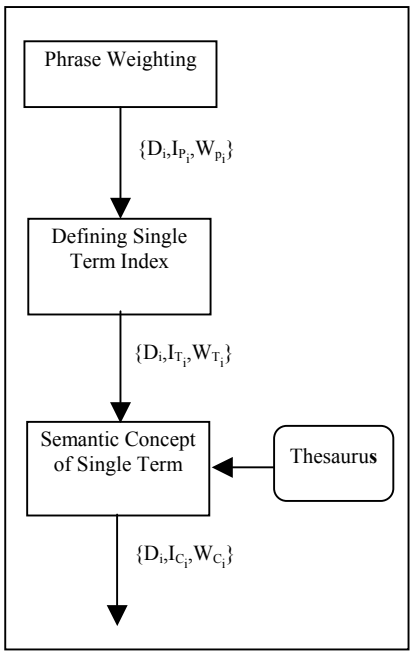

***Figure 6**: Multilevel Index Generation*

In each level of index we use Salton's Weight normalization [15] as shown below is used for computing weights.

$$V_k^m = \frac{tf_k^m \log \frac{N_d}{n_k}}{\sum_{j=1}^{l} tf_j^m \log \frac{N_d}{n_j}}$$

tf_k^m = Number of index terms k in document m
n_k = Number of documents that contain term k
N_d = Number of documents in the collection
l = Number of index terms

The parallel processing of the document is providing by the AHYDS engine using the EBG data structure. Each level of index of each document is computed dependently but independently from other document.

More details of the algorithm of multilevel index generation is given in Annex 2.

Figure 7 shows the comparison between multilevel indexing and traditional indexing system

Using multilevel indexing, "egg" would not be retrieved, while, in traditional IR, it will be retrieved which degrade the performance of the system.

3.2.4 Document Classification

Even though multi-level indices can cover a very wide range of document retrieval without degradation of system performance, document clustering for pruning irrelevant documents is still

$$D_i =< I_{p_i}, I_{t_i}, I_{c_i}, C_i >$$

necessary in order to increase precision and decrease searching time.

Text categorization or document clustering consists of two parts: a prototype learning process to provide prototypes for each cluster of documents and a clustering process, which compute the similarity between input document and prototype (see Figure 8).

Finally, document will be represented as

where

$$I_{p_i} =< W_{p1_i}, W_{p2_i}, ..., W_{pt_i} >$$
$$I_{t_i} =< W_{t1_i}, W_{t2_i}, ..., W_{tt_i} >$$
$$I_{c_i} =< W_{c1_i}, W_{c2_i}, ..., W_{ct_i} >$$
$$C_i =< W_{cat_{1_i}}, W_{cat_{2_i}}, ..., W_{cat_{n_i}} >$$

The algorithm of document clustering is summarized in Annex 3.

4. Query processing

In order to obtain those documents, which have the best match with a given query, we also need a "query guide". Query Guide is applied by using the cluster hypothesis and query expansions. Our method apply Word-Net for reconstructing query by adding more general term/concept. For example

Query = "น้ำดอกไม้." (proper name: the name of mango) and its general term (from Word-Net) is "มะม่วง" (mango). After expansion, the new query is "มะม่วง-น้ำดอกไม้" (The phrase contains of mango and its specified name).

5. Retrieval Processing

The following retrieval process is implemented for enhancing the performance of the system (see figure 9):

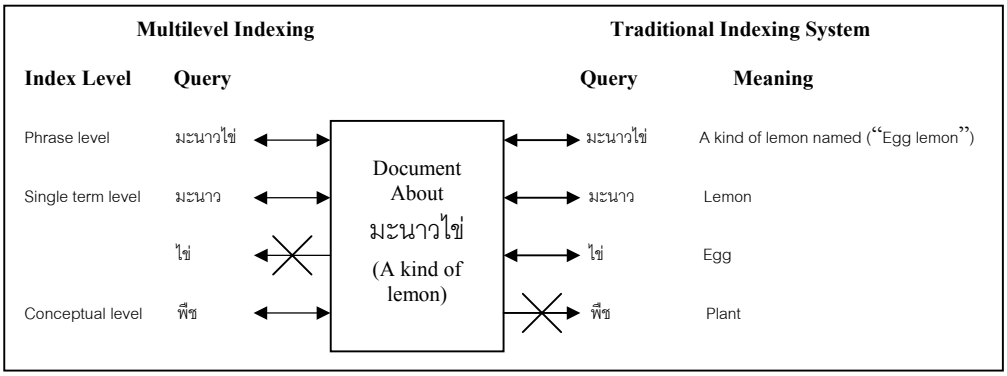

***Figure 7**: Example of how Multi-Level Indexing can enhance performance*

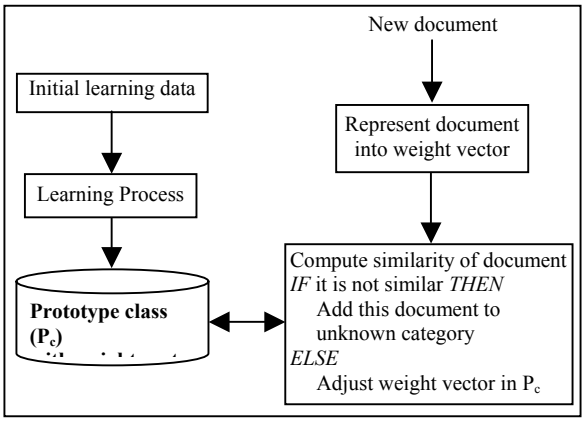

***Figure 8**: Text Categorization process*

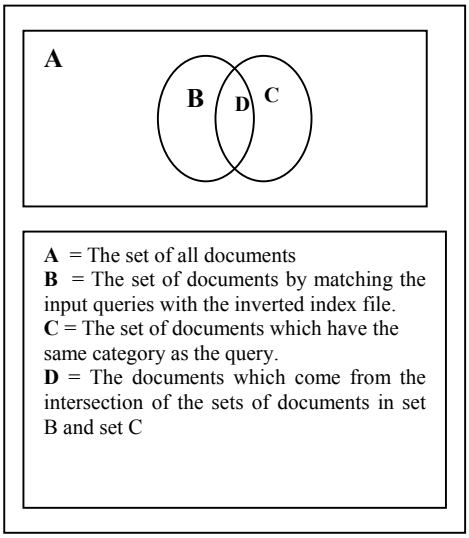

***Figure 9**. The Retrieved Documents*

1. Compute a candidate set of documents by matching the input queries with the inverted index file.
2. Select a candidate set of documents which have the same category as the query.
3. Calculate the similarity between the queries of the documents which come from the intersection of the sets of documents in 1 and 2.
4. Return a set of retrieved document with the similarity scores.

6. Conclusion and Future Work

Figure 10 shows the difference between a set of index with applying and without applying NLP techniques.

At the current state, knowledge acquisition is processed manually by linguists. Next step it will be provided by semi-automatically. The domain of documents is limits in computer area. However, it will be extended to cover agriculture and general news area.

References

[1] Andres F., "Active Hypermedia Delivery System and PHASEMA Information Engine" in Proc. First International Symposium on Advanced Informatics, Tokyo, Japan, 2000.

[2] Andres F., Kawtrakul A., Ono K. and al., "Development of Thai Document Processing System based on AHYDS by Network Collaboration", in Proc. 5th international Workshop of Academic Information Networks on Systems(WAINS), Bangkok, Thailand, December 1998.

[3] Andres F., and Ono K. "The Active Hypermedia Delivery System", in Proceedings of ICDE98, Orlando, USA, February 1998.

[4] Boncz, P.A. and Kerstern, M.L. "Monet: An Impressionist Sketch of an Advanced Database System" In Proc. IEEE BITWIT Workshop, San Sebastian (Spain), July 1995.

[5] E. Chaniak, "Statistical Language Learning", MIT Press, 1993.

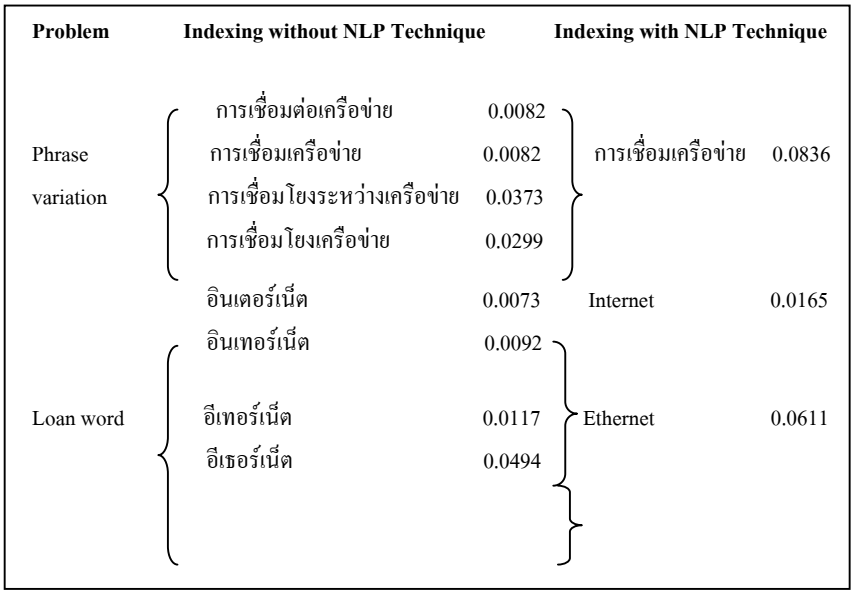

***Figure 10**. Examples of Applying NLP Technique to solving phrase variation and loan word*

[6] Geppert, A. Dittrich, K.R. "Constructing the Next 100 Database Management Systems: Like the Handyman or Like the Engineer ?" in SIGMOD RECORD Vol.23, No 1, March 1994.

[7] Geppers A., Scherrer S., and Dittrich K.R. "Kids: Construction of Database Management Systems based on Reuse", Technical report 97.01, Institutfur Informatik, University of Zurich, Switzerland, 1997.

[8] Grosky W.I. "Managing Multimedia Information in Database System" in Communication of the ACM December 1997, Vol 40., No 12, page 73-80.

[9] G. Salton, "Automatic Text Processing. The Transformation, Analysis, and Retrieval of Information by Computer", Singapore: Addison-Wesley Publishing Company, 1989.

[10] Kawtrakul A., Andres F., et.al.,. "A Prototype of Globalize Digital libraries: The VLSDHS Architecture for Thai Document processing." 1999. (on the process of submission)

[11] Kawtrakul A., Andres F., Ono K. and al., "The Implementation of VLSHDS Project for Thai Document Retrieval" in Proc. First International Symposium on Advanced Informatics, Tokyo, Japan, 2000.

[12] Kawtrakul A., et.al., "Automatic Thai Unknown Word Recognition", In Proceedings of the Natural Language Processing Pacific Rim Symposium, Phuket, pp.341-346, 1997.

[13] Kawtrakul A., et.al., "Backward Transliteration for Thai Document Retrieval", In Proceedings of The 1998 IEEE Asia-Pacific Conference on Circuits and Systems, Chiangmai, pp. 563-566, 1998.

[14] Kawtrakul A., et.al., "A Lexibase Model for Writing Production Assistant System" In Proceedings of the 2nd Symposium on Natural Language Processing, Bangkok, pp. 226-236, 1995.

[15] Seshadri P., Livny M., and Ramakrishnan R. "The Case for Enhanced Abstract Data Types" In Proceedings of 23rd VLDB Conference, Athens, Greece, 1997, pages 56-65.

[16] Subrahmanian V.S. "Principles of Multimedia Database Systems", Morgan Kaufmann, 1997.

[17] Teeuw W.B., Rich C., Scholl M.H. and Blaken H.M. "An Evaluation of Physical Disk I/Os for Complex Object Processing" in Proc. IDCE, Vienna, Austria, 1993, pp 363-372.

[18] Valduriez P., Khoshafian S., and Copeland G. "Implementations techniques of Complex Objects" in Proc. Of the International Conference of VLDB, Kyoto, Japan, 1986, pp 101-110.

Biography

Assoc. Prof. Asanee Kawtrakul,Ph.D., researcher, received bachelor degree (honor) and master degree in electrical engineering from Kasetsart University in 1976 and 1986, respectively. She received Ph.D in Information Engineering from Nagoya University in 1991. She has been the lecturer for Faculty of Engineering since 1983. She had been the project leader in several projects in the fields of Natural Language Processing, Text Retrieval Database, Speech Synthesis, Database Management System, and Geographical Information System.

Assoc. Prof. Frederic Andres,Ph.D., the National Institute of Informatics (NII), Japan. He was visiting Researcher at NACSIS since 1996 under the COE framework of the Japanese ministry of Education. He received his PhD and HDR degree in 1993 and in 2000 respectively

from the University of Paris VI (France) and University of Nantes (France). His research interests include distributed and heterogeneous multimedia information systems, and multimedia document retrieval. He was a scientist at Bull from 1989 to 1993, and system architect at Ifatec/Euriware from 1993 to 1996.

Annex 1

Algorithm Phrase Identification and Relation Extraction

Input: a list of lexical token $w_1, w_2, \ldots, w_n$
with set of POS tag information $T_i = \{t_1, t_2, t_3, \ldots, t_m\}$,
frequency f_i and weight W_i for each word

Output: set of candidate index NPs with head-modifier relation
or compound relation

Candidate Index Selection:
Selecting candidate index by selecting term which have weight $w_i > \theta$
(θ is an index threshold)

Phrase boundary identification:
FOR each candidate term *DO*
Apply NP rule to find boundary
IF can not apply rule directly
Consider weight of adjacent term w_{adj} *THEN*
IF adjacent term has weight $w_{adj} > \phi$
(ϕ is a boundary threshold) *THEN*
Extend boundary to this term
ELSE
IF this adjacent term in the preference list
Extend boundary to this term *THEN*

Relation Extraction:
FOR each candidate index phrase *DO*
To find internal relation we consider term frequency
in each phrase
IF the frequency of each word of candidate NP
has the same frequency *THEN*
Relation of this NP is *compound noun*
ELSE
Relation of this NP is *head-modifier pair* :
Head is the term(s) with highest frequency.
Modifier is the term(s) with lower frequency.

Annex 2

Algorithm Multilevel Index Generation

Input: 1. A list of lexicon token provided by Lexicon Token Identification and Extraction process w1, w_2,…, w_j with its frequency f_{wi} and weight w_{wi}.
2. A list of candidate Phrasal indices with head-modifier relation or compound relation.

Output: Index weight vector as document representation

$$D_i = \{I_{p_i}, I_{t_i}, I_{c_i}\}$$

Where

$$I_{p_j} = \{w_{p_1}, w_{p_2}, \ldots, w_{p_j}\}$$
$$I_{t_j} = \{w_{t_1}, w_{t_2}, \ldots, w_{t_j}\}$$
$$I_{c_j} = \{w_{c_1}, w_{c_2}, \ldots, w_{c_j}\}$$

Phrasal Level Indexing:

FOR each candidate Index NP *DO*
Recompute Phrase weights in whole documents
IF phrase weight > θ (θ *is* index threshold)
Keep sorted Phrase index token *THEN*

Single Term Level Indexing:

FOR each candidate phrase index NP *OR* each single term *DO*
IF tokens of candidate phrasal index
Extract the head of the token *THEN*
Recompute weight
ELSE
FOR each lexicon token that not appear in phrasal level *DO*
Recompute weight
Keep sorted Single term indices

Conceptual Level Index:

FOR each single term index *DO*
Find Semantic Concept of each single terms
Recompute weight
Keep Sorted Conceptual Level Index

Annex 3

Algorithm Document Clustering

Input: single term indices and phrase indices with their frequencies

Output: Document representation in $<I_{p_i}, I_{t_i}, I_{c_i}, C_i>$

Learning Process:

Define prototype class e.g. Computer, Agriculture, News etc.

FOR* each documents *DO

Compute weight vector of single term/phrasal indices by using

$$W_x^m = \frac{tf_x^m \log \frac{N_d}{n_x}}{\sum_{j=1}^{l} tf_j^m \log \frac{N_d}{n_j}}$$

W_x^m = Weight Vector of phrase indices x in document mth
x = k mean Single term index
x = p mean Phrasal index
tf_x^m = {0 if $f_x^m = 0$, $\log(f_x^m) + 1$ otherwise}
n_k = Number of documents that contain term *k*
N_d = Number of documents in the collection
l = Number of index terms

FOR each prototype class *DO*

Compute weight vector of single term indices by using Rocchio's algorithm is used [3, 5]:

$$W_{ck} = \begin{cases} 0 & \text{Otherwise} \\ W'_{ck} & \text{If } W'_{ck} > 0 \end{cases}$$

$$W'_{cx} = \beta \frac{1}{|R_c|} \sum_{i \in R_c} W_x^m - \gamma \frac{1}{|\bar{R}_c|} \sum_{i \in \bar{R}_c} W_x^m$$

W_{cx} = weight of term k in the prototype P_c for class.
x = k = Single Term index
x = p = Phrasal index
W_x^m = weight of term k indexing for each document
x = k = Single Term index
x = p = Phrasal index
R_c = set of training documents belonging to class c
$\bar{R}_c$ = set of documents not belonging to class c
(Note: $\beta = 16$, $\gamma = 4$ [Buckley et al., 1994])

Classification:

FOR new document input *DO*

Compute weight of phrase index and weight of single term index by using formula:

$$W_x^m = \frac{tf_x^m \log \frac{N_d}{n_x}}{\sum_{j=1}^{l} tf_j^m \log \frac{N_d}{n_j}}$$

W_x^m = Weight Vector of phrase indices x in document mth
x = k mean Single term index
x = p mean Phrasal index
tf_x^m = {0 if $f_x^m = 0$, $\log(f_x^m) + 1$ otherwise}
n_k = Number of documents that contain term *k*
N_d = Number of documents in the collection
l = Number of index terms

Compare weight with each Prototype Class by using the dot product formula

$$S(D_i, C) = \alpha \frac{\sum_{k=1}^{t} (W_p^m * W_{cp})}{\sqrt{\sum_{k=1}^{t} (W_p^m)^2 * \sum_{k=1}^{t} (W_{cp})^2}} + \beta \frac{\sum_{x=1}^{t} (W_k^m * W_{ck})}{\sqrt{\sum_{x=1}^{t} (W_k^m)^2 * \sum_{x=1}^{t} (W_{ck})^2}}$$

W_{ck} = Weight of Single term *k* in the Prototype P_c for class
W_k^m = Weight of Single term *k* in document mth
W_{cp} = Weight of Phrase *p* in Prototype P_c for class
W_i^m = Weight of Phrase *p* in document mth
α = [0, 1]
β = [0, 1]
$\alpha + \beta = 1$

FOR each C, *DO*

IF $S(D_i, C) > \theta_t$ *THEN*

Store document into Prototype P_c and adjust weight in Prototype Class.