

Newly-Born Keyword Extraction under Limited Knowledge Resources based on Sentence Similarity Verification

Rachada Kongkachandra*, Chom Kimpan†, Thawatchai Suwanapong† and Kosin Chamnongthai*

*King Mongkut's University of Technology Thonburi
Bangkok 10140 Thailand, Tel/Fax: +66-2-4709070,
rdk@cs.tu.ac.th and kosin.cha@kmutt.ac.th

†The University of the Thai Chamber of Commerce, Bangkok 10400, Thailand
Tel/Fax: +66-2-2754892, E-mail: thawatchai.suw@utcc.ac.th

‡Sripatum University, Bangkok, 10900, Thailand
Tel: +66-2-5611721, E-mail: chom@spu.ac.th

Abstract—Most of keyword extraction systems are utilized the external knowledge bases such as lexicon, ontological databases, keyword lists and training corpus for extracting keywords. The drawbacks of using external knowledge bases are formatting incompatibility, domain constraint, and updating difficulty. This paper proposes a keyword extraction system, which uses only the knowledge within the document, to extract the keywords. The knowledge includes the title and content of document. All candidates are determined as keywords if they have the same meaning as seed keywords. These seed keywords are previously extracted by employing the document title. We syntactically parse the document title and then use all noun phrase combinations as seed keywords. These keywords are used as index to pick the relevant sentences in the document and then automatically converted into knowledge representation, called semantic graphs. Simultaneously, the candidates for being accepted as keywords are processed in the same steps as the seed keyword extraction. However, the candidates are altered from the whole sentences in the document instead of the document title as in seed keyword extraction. The similarity score of keyword semantic graphs and candidate semantic graphs are calculated based on conceptual graph operations. The experiments are arranged with 100 documents ten domains such as business, computer and psychology. We compared the experimental results to the author-defined keywords and two practical keyword extraction systems i.e. EXTRACTOR and KEA. The experimental results show the acceptable performance with 54.81%, 73.87% and 77.94% when they are compared to the keywords defined by author, EXTRACTOR and KEA, respectively.

I. INTRODUCTION

Keywords play important role in several applications such as text summarization, information retrieval and question answering system. The benefit of using keyword list as document representative can reduce reading time, assist the user to arrange the large amount of documents and facilitate users to communicate with machine. Unfortunately, extracting a keyword from the document is not so easy because of the following problems.

- 1) There is no standard keyword pattern.

- 2) Knowledge resources for keyword extraction are limited.

To solve the first problem, many researchers use text corpus for training the keyword patterns. They use various learning techniques such as statistics, machine learning and genetic algorithm. KEA [3,8], EXTRACTOR [5] and GenEX [6] are the practical keyword extraction systems based on the mentioned learning techniques, respectively. Although these keyword extraction systems yield the acceptable extraction performances, they will be suffered when they extract a newly-born keyword. The keyword patterns, collected in knowledge resources such as semantic ontology and keyword database, are essentially re-learned for knowing the newly-born keywords and the whole knowledge resources are also re-integrated.

Therefore, this paper presents the simplified keyword extraction system. The proposed keyword extraction system uses only document title and content as keyword resource for extracting keyword. Although, there are newly-born keyword occurred in the document, the proposed keyword extraction system can detect them if there are surrounding sentences explaining these newly-born keywords.

II. BASIC IDEA

To extract the new keyword, the surrounding information of the new word is required. In this paper, we use the surrounding sentence level information for verifying its meanings with the existing keywords. For example:

"COBOL is a programming language for simple computations on large amounts of data, designed by the CODASYL Committee in April 1960."

"Perl is a high-level programming language, started by Larry Wall in 1987 and developed as an open source project."

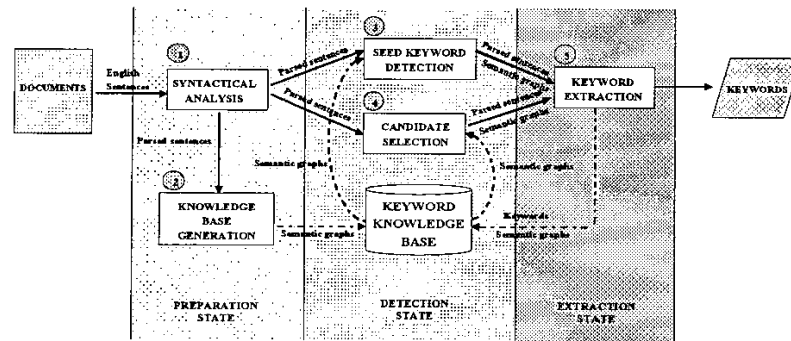


Fig. 1. The proposed Keyword Extraction System

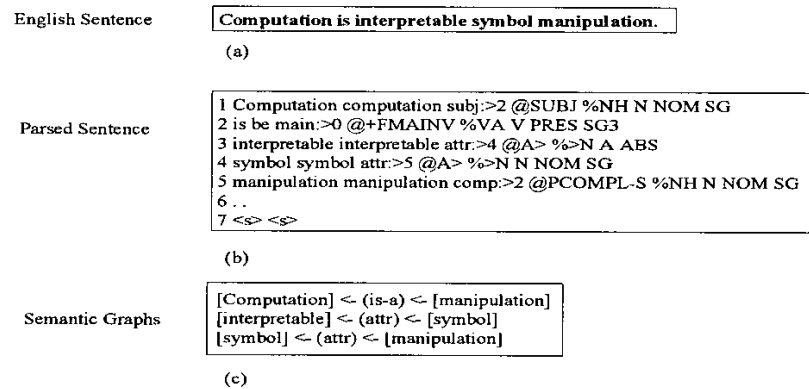


Fig. 2. Original Sentence and its Parsed Sentences

"Java" is a simple, object-oriented, distributed, interpreted, robust, secure, architecture-neutral, portable, multithreaded, dynamic, buzzword-compliant, general-purpose programming language developed by Sun Microsystems in 1995."

These three sentences are the definitions of "COBOL", "Perl" and "Java". Suppose that "COBOL" is the keyword in the domain of computer programming. "Perl" and "Java" are the new keywords in the same domain. Since "perl" and "Java" have the well-known meanings as "a shade of white the color of bleached bones" and "a Indonesian island" respectively, they are located in far away from the node "COBOL" in the semantic ontology. Therefore, they are rejected as the new keywords.

To solve this problem, instead of considering the meaning in word level, we propose to use the meaning of the whole sentence for extracting keyword.

III. SYSTEM OVERVIEW OF THE PROPOSED KEYWORD EXTRACTION SYSTEM

Figure 1 illustrates the overview of our proposed keyword extraction system. It is divided into three states i.e. preparation,

detection and extraction states.

A. Preparation State

The preparation state includes two processes i.e. syntactical analysis and knowledge base generation. This state is used for preparing knowledge base for the whole system. Since we propose the semantic-based method for extracting keyword, all the document is necessarily represented in form of knowledge representation for the subsequence semantic matching.

1) *Syntactical Analysis*: The document title and content are separately parsed and kept into parsed files. In this paper, we use the demo version of ENGCG[11], to produce the parsed sentences. The example result of this process is shown in Fig. 2(a).

2) *Knowledge Base Generation*: Each sentence in the parsed file from the previous process is then converted into the knowledge representation format. In this paper, conceptual graph is chosen because it can represent the relation of words in each sentence in readable form. Each conceptual graph, called semantic graph, consists of three sequential nodes. The three sequential nodes are ordered as one box, one circle and one box. The box represents the concept entity while the circle represents the concept relation between these two boxes. The

```

manipulation
[clearly] <- (man) <- [is]
[computation] <- (subj) <- [is]
[interpretable] <- (attr) <- [symbol]
[just] <- (meta) <- [manipulation]
[manipulation] <- (comp) <- [is]
[of] <- (mod) <- [manipulation]
[symbol] <- (attr) <- [manipulation]
[symbol] <- (attr) <- [tokens]
%%
semantic
[categories] <- (pcomp) <- [of]
[encompass] <- (mod) <- [language]
[in] <- (ha) <- [used]
[network] <- (pcomp) <- [in]
[networks] <- (obj) <- [encompass]
[networks] <- (subj) <- [do]
[of] <- (mod) <- [set]
[semantic] <- (attr) <- [categories]
[semantic] <- (attr) <- [network]
[semantic] <- (attr) <- [networks]
%%
symbol manipulation
[as] <- (copred) <- [is]
[being] <- (v-ch) <- [based]
[clearly] <- (man) <- [is]
[Completely] <- (ad) <- [different]
[different] <- (attr) <- [symbol-shapes]
[formal] <- (attr) <- [symbol]
[implementation] <- (mod) <- [symbol-shapes]
[in] <- (mod) <- [symbols]
[interpretable] <- (attr) <- [symbol]
[manipulation] <- (comp) <- [is]
[move] <- (mod) <- [symbols]
[of] <- (mod) <- [shapes]
[on] <- (mod) <- [operations]
[read] <- (subj) <- [move]
[symbol] <- (attr) <- [manipulation]
[symbol] <- (attr) <- [system]
[symbols] <- (pcomp) <- [of]
[symbols] <- (pcomp) <- [on]
[symbol-shapes] <- (subj) <- [could]
%%

```

Fig. 3. Example of Knowledge Base

arrow represents the direction of their relations. The examples of the semantic graphs for the sentence "Computational is interpretable symbol manipulation" are shown in fig. 2(b).

To automatically converted the parsed sentences into semantic graphs. We modify the transformation rules from [2], which are used based on context-free-grammar so that they can be used with the parsed sentences based on the dependency grammar.

B. Detection State

The detection state is used for detecting the seed keyword list and all possible word patterns voted for keyword candidate. In this paper, we assume that the possible keywords and candidates are "noun phrases" with the maximum three words. We used the English grammar mentioned to automatically determine the compound adjectives and nouns as possible noun phrases.

The two processes i.e. seed keyword detection and candidate selection are arranged in the same algorithm for finding the possible noun phrases.

1) *Seed Keyword Detection*: The seed keywords are then used as anchor in the last state for extracting the remaining keywords. The seed keywords are possible noun phrases generated from the document title. We assume that the authors commonly intend to entitle the document by using the important key words in the document.

Next, each keyword is used as index to search for the relevant semantic graphs in the knowledge base created in the first step. Figure 3 shows the example of our keyword knowledge base.

2) *Candidates Selection*: In this process, the whole document is employed for generating noun phrases. The keywords and stop words are then excluded from these noun phrases by matching them with keyword list, extracted from seed keyword detection, and stop word list, manually generated by user.

Similar to seed keyword detection, these candidates are then used as index to search for relevant semantic graphs in the knowledge base.

C. Extraction State

1) *Sentence-based Keyword Extraction*: To determine the semantic of keyword, the semantic graphs of candidates and keywords are apparently matched. The candidate is accepted to become a keyword if it has the word score over the specified threshold, computed by (1), (2) and (3), satisfying (4). Figure 4 displays our proposed matching method.

$$gpscore(g) = \frac{\sum_{n=1}^3 hit(n)}{3} \quad (1)$$

$$stscore(s) = \max(gpscore(g)) \quad (2)$$

$$wdscore(w) = \frac{\sum_{s=1}^{ns} stscore(s)}{ns} \quad (3)$$

where $hit(n)$ is the number of matched nodes between candidate and keyword nodes.

$gpscore(g)$ (Graph Score) is the average number of matched graph node of the g th semantic graph.

$stscore(s)$ (Sentence Score) is the maximum of $gpscore(g)$ of the s th sentence.

$wdscore(w)$ (Word Score) is the average $stscore(s)$ of the w th candidate.

n is the total number of nodes in each graph.

ns is the total number of sentences of the w th candidate.

The acceptance of the candidates are determined by the following criteria as shown in (4)

$$Accept(wdscore) = \begin{cases} 1 & \text{if } wdscore \geq T \\ 0 & \text{if } wdscore < T \end{cases} \quad (4)$$

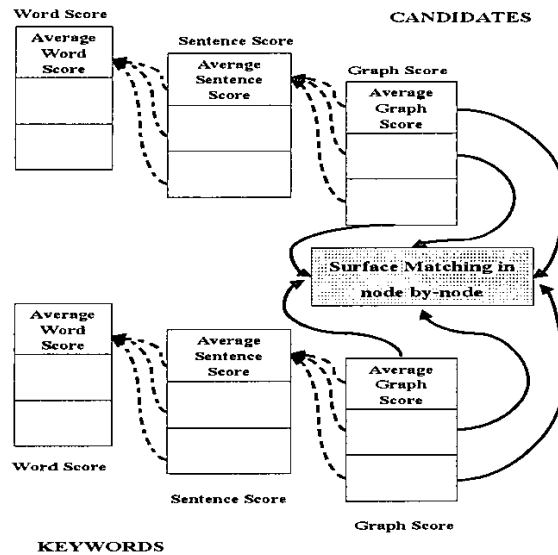


Fig. 4. Semantic Graphs Matching

TABLE I
EXPERIMENT SETTING

Con. guration	Practical Setting
Conventional Keyword Extractor	1. EXTRACTOR 2. KEA
Number of domains	10 domains
Number of documents	100 documents
Number of desired keywords	5 keywords

where T is the specified threshold

IV. EXPERIMENTS AND RESULTS

As we propose to develop the keyword extraction system under the limited knowledge resource by using only its document information for extraction. The configuration of our experiment is summarized in Table 1. The selected conventional keyword extraction systems are KEA [3,8] and EXTRACTOR [5-6]. They are the representative of statistic-based and machine learning-based keyword extraction systems. We use 100 documents in various ten domains for evaluating our proposed system performances.

Table 2 illustrates the results of the proposed keyword extraction system compared to the keywords defined by author, extracted by EXTRACTOR and KEA, respectively. We evaluate the performances based on the meaning similarity of our proposed keywords with the keywords extracted by author, EXTRACTOR and KEA. The number of desired keywords used in the experiments is five keywords per document.

From Table 2, our extracted keywords are matched to the author-defined keywords with only 54.81%. The cause of this effect is that our proposed extracted keywords are based on

TABLE II
THE OVERALL SYSTEM PERFORMANCE

The Experimental Results Compared with		
Author-de. ned	EXTRACTOR	KEA
54.81%	73.87%	77.94%

keyword extraction. It means that the keywords should occur in the document at least once. While author-defined keywords are based on keyword assignment. They possibly assign some words that are not shown in the document as keywords. When we compare the results with EXTRACTOR and KEA, which keyword extraction based approaches, it shows the higher performance with 73.87% and 77.94%, respectively.

V. CONCLUSION

This paper proposes the simplified keyword extraction system that uses only the document title and content to extract the newly-born keywords. The keywords and candidates are semantically matched based on sentence level consideration. The sentences explaining the keywords and candidates are converted into semantic graphs. Then, theses semantic graphs are matched based on the operations of conceptual graph. The proposed keyword extraction system can accurately extract the desired number of keywords with 54.81%, 73.87% and 77.94% compared to the keyword list defined by author and extracted from EXTRACTOR and KEA, respectively.

ACKNOWLEDGMENTS

This work is supported by the Thailand Research Fund, Thailand under the Royal Golden Jubilee Project.

REFERENCES

- [1] K. Barker and N. Cornacchia. 2000. Using noun phrase heads to extract document keyphrases. In *Canadian Conference on AI*, pages 4052.
- [2] Caroline Barriere. 1997. From a Childrens First Dictionary to a Lexical Knowledge Base of Conceptual Graphs. Ph.D. thesis, School of Computer Science, Simon Fraser University, July.
- [3] E. Frank, G. W. Paynter, I. H. Witten, C. Gutwin, and C. G. Nevill-Manning. 1999. Domain-specific keyphrase extraction. In *The Sixteenth International Joint Conference on Artificial Intelligence (IJCAI- 99)*, pages 668673.
- [4] H. Nakagawa. 2000. Automatic term recognition based on statistics of compound nouns. *Terminology*, 6(2):195210.
- [5] P. D. Turney. 1999. Learning to extract keyphrases from text. ERB 1057, National Research council, Institute for Information Technology.
- [6] P. D. Turney. 2000. Learning algorithms for keyphrase extraction. *Information Retrieval*, 2(4):303 336.
- [7] A. Voutilainen and J. Heikkilä. 1993. An english constraint grammar (ENGCG): a surface-syntactic parser of english. In *The Fourteenth International Conference on English Language Research on Computerized Corpora*, pages 189199.
- [8] I. H. Witten, G. W. Paynter, E. Frank, C. Gutwin, and C. G. Nevill-Manning. 1999. Kea:practical automatic keyphrase extraction. In *Proceedings of the Fourth ACM Conference on Digital Libraries*.