The Performance Evaluation of A Knowledge-based Information Retrieval System

Dongwook Shin a, Yongun Yoon, and Keysun Choi b

aDepartment of Computer Engineering, Chungnam National University, 220 Kungdong, Yusong-Gu, Taegon, 305-764, South Korea
bDepartment of Computer Science, KAIST, 373-1 Kusong-Dong, Yusong-Gu, Taejon, 305-701, South Korea

This paper aims at measuring the performance of an information retrieval system named HYKIS. HYKIS (HYbrid Knowledge-based Information retrieval System) is a knowledge-based information retrieval system that allows a naive user to retrieve relevant documents effectively, without any help of a domain expert. According to an experiment with the CACM data set, it turns out that HYKIS achieves much higher recall and precision rate, compared to a thesaurus-based retrieval system.

1. Introduction

This paper aims at measuring the performance of an information retrieval system HYKIS that has been proposed in [3-5]. HYKIS (HYbrid Knowledge-based Information retrieval System) is a knowledge-based information retrieval system that allows a naive user to retrieve relevant documents effectively, without any help of a domain expert. For accomplishing this, it has a hybrid knowledge base that lets most of heuristics be represented in a simple and structured manner. As knowledge can be categorized by domain independent and domain dependent one, it separates the store of the knowledge into two bases: domain independent knowledge base (Ibase) and domain dependent knowledge base (Dbase). Ibase is used as a graph-based thesaurus that permits “is-a” relation between index terms. Dbase describes domain-specific information, such as synonym relationship, some term relationships other than “is-a” relation, and some heuristics for navigating Ibase.

HYKIS is only a hybrid knowledge-based information retrieval system among those which have been proposed in the literature [2]. In HYKIS, the query evaluation procedure, QEP retrieves the documents relevant to a query given as Boolean form, interacting with Ibase and Dbase. According to an experiment with the CACM data set, it turns out that HYKIS achieves much higher recall and precision rate, compared to a thesaurus-based retrieval system.

2. Two Knowledge Bases in HYKIS

The knowledge base of HYKIS is composed of the aforementioned two bases: Ibase and Dbase. Ibase is concerned of the domain independent knowledge, while Dbase deals with domain dependent knowledge. The system overview of HYKIS comprising two bases is depicted in Figure 1.

Ibase plays the role of thesaurus consisting only of index terms, and can be used in the general domains or general topics. HYKIS usually has only one Ibase, since Ibase possesses all of common classification knowledge between terms, whereas it has many Dbases, each of which corresponds to a domain in HYKIS. Each Dbase allows to convey the relationship other than “is-a” and several heuristics of the domain expert and librarian for retrieving proper information items.

The Retrieval module of the Query Evaluation Procedure (QEP) evaluates a user query, interacting with these bases. Using two kinds of knowledge together, QEP produces a suitable classification schemes for the searcher’s interesting topic. And it accesses the document sets, via the inverted file that keeps the relationship between index terms and documents. Finally, the Ranking module ranks the selected documents and takes some of the highest ranked documents as a result.
The following two subsections discuss these bases briefly such as how knowledge is represented.

2.1. Ibase for Domain Independent Knowledge

Basically, Ibase is a hierarchical thesaurus viewed as a DAG (Directed Acyclic Graph). Among the various inter-term relation such as “is-a”, “synonym”, “antonym”, “part-and-whole”, and “affinity” relation, Ibase only offers the “is-a” relation, since it is heavily concerned about common sense knowledge of a human expert or a knowledge engineer. In Ibase, the nodes constitute a set DES of index terms. A direct edge in Ibase represents a relation $R$ between two adjacent term in DES. The notation $c_i R c_j$ indicates that the index term $c_i$ is more general than the index term $c_j$. Note that $R$ is irreflexive, asymmetric, and transitive. There exists a dummy node in DES as a root of Ibase from which all other nodes are connected. If $c_i$ is in DES, then $c_i$ appears only once in Ibase and there exist at least one $c_j$ in Ibase such that $c_i R c_j$ or $c_j R c_i$.

Ibase offers six operators for constructing or navigating the thesaurus:

1. def_descriptor($C_i$),
2. def_isa($C_i, C_j$),
3. del_descriptor($C_i$),
4. del_isa($C_i, C_j$),
5. find_all_c($C_i, T, X$),
6. find_partial_c($C_i, Depth, T, X$).

The first operator lets the term $C_i$ in DES be generated in Ibase, while the second makes a “is-a” relation from $C_i$ to $C_j$. The latter two operators play the similar roles as the former two, except that they perform the deletion of terms or “is-a” relations instead of insertion. The fifth one gathers all subterms below $C_i$ and assigns them to $X$, while the sixth one finds subterms of $C_i$ whose distance from $C_i$ is less than or equal to $Depth$. These operators, at the same time, extract the subtrees involving only the terms in $X$ and assign them to $T$. The first four operators are mainly used for constructing a thesaurus, whereas the last two are employed for searching related terms in retrieval stage.

2.2. Dbase for Domain Dependent Knowledge

In contrast to Ibase, Dbase lets the domain dependent heuristics of domain experts be represented. In our system, Dbase is used as the front-end to Ibase that helps to convert the arbitrary terminology given by an IR user to the index terms implied in Ibase and reflects domain dependent knowledge. The usefulness of Dbase is easily demonstrated, considering that it is extremely difficult for an average user to compose a query made up only of index terms without the aid of an expert.

For obtaining better retrieval result, Dbase is able to have heuristics for helping information seekers produce proper index terms and identifying the search depth of Ibase. Dbase allows to represent several heuristics that a domain expert
and librarian have as following rules:

rule 1. assertions for synonyms,

rule 2. domain restriction for a multiple meaning term (polysemy),

rule 3. generation rules for complex compound phrases,

rule 4. navigation rules for affinitive relationships other than “is-a”,

rule 5. conversion rules from non-index terms to the corresponding index terms on a given topic,

rule 6. determining the depth of search of an index term in Ibase.

These rules are described by the six operators, respectively as:

1) def_syn($C_i, C_j$),
2) def_poly($C_i, D_j$),
3) def_compound($C, (C_1, \ldots, C_n)$),
4) def_affin($C, (C_1; \ldots; C_n)$),
5) conv_index($C_i, D_j$), and
6) def_depth($D_i, n$), where $C_i$ is a general term and $D_j$ is an index term.

3. Query Evaluation of HYKIS

The aim of The Query Evaluation Procedure (QEP) is to retrieve the relevant documents in the aid of Ibase, Dbase and the inverted file, when an input query is given as Boolean form.

First, the query interacts with the proper Dbase by the Dbase handler. The Dbase handler repeats the replacement of the arbitrary terms in the current query by the index terms described in Ibase.

Secondly, taking the refined query consisting only of index terms, the Ibase handler searches the narrower terms of each index term and makes the appropriate search commands into the inverted file. Here, if the search depth of an index term $t$ is constrained as $n$ in Dbase, it finds the terms whose depths are within $n$ from $t$.

Thirdly, the inverted file system extracts the relevant document set for each operator and fourthly, Join/Union module does join, union and negation for the connectives “and”, “or”, and “not” operators. Finally the ranking algorithm computes the distances of the query and the retrieved document sets, and takes highest ranked ones as the final result.

4. Analysis of HYKIS

HYKIS has been implemented in Sicstus Prolog, mainly because the query evaluation resembles the SLD-resolution in Prolog.

We conducted an experiment with the CACM data set that is able to measure the performance of HYKIS. The CACM data set consists of a thesaurus named CRCS (Computing Reviews Classification Structure) [1], more than three thousand documents, sixty four sample queries, and the sets of relevant documents for the queries. Single-term indexing was employed with the stop and stemming list given in the data set and, as the result, an inverted file was created. The Dbase has been made appropriately, referring to CRCS.

Figure 2 shows that HYKIS achieves much higher recall/precision rate, comparing with a thesaurus-based retrieval system. In the average, HYKIS produces nearly 40 percent recall and precision rate, whereas the thesaurus-based system shows only 12 and 18 percent in recall and precision, respectively. In fact, these two systems do not make use of the thesaurus well, because we made an indexing scheme based on single-term, while the thesaurus uses controlled vocabulary. Thus, the thesaurus-based retrieval often degenerates into the simple retrieval by the exact match of terms and application of Boolean operators.

As depicted in Figure 2, in some queries (q3 and q9), the knowledge of Dbase operates negatively in precision, because it helps to find a greater number of inappropriate documents than relevant ones. It also imply that only the proper knowledge made by a domain expert is able to improve both recall and precision rate.

We also notice that poor stemming and single-
term indexing prevent a retrieval system from
finding proper documents. In a query $q_6$, HYKIS
produces low retrieval rate. It is not because
HYKIS works badly for the queries, but because
single-term indexing limits the search of relevant
documents. That is, by separating “message” from “passing” in the term “message passing”,
each single word does not imply the real mean-
ing of “message passing” fully. Hence, we expect
that a better stemming method and the use of
the controlled vocabulary instead of single-term
indexing can improve the performance of HYKIS
in some degree.

Recall

\[
\begin{array}{cccccccccc}
q1 & q2 & q3 & q4 & q5 & q6 & q7 & q8 & q9 & q10 \\
60.0 & 30.0 & 33.3 & 25.0 & 33.3 & 3.57 & 17.7 & 26.3 & 51.6 \\
60.0 & 30.0 & 33.3 & 25.0 & 33.3 & 3.57 & 17.7 & 26.3 & 51.6 \\
\end{array}
\]

Precision

\[
\begin{array}{cccccccccc}
q1 & q2 & q3 & q4 & q5 & q6 & q7 & q8 & q9 & q10 \\
60.0 & 40.0 & 26.7 & 20.0 & 40.0 & 40.0 & 54.5 & 52.5 & 27.3 \\
60.0 & 40.0 & 26.7 & 20.0 & 40.0 & 40.0 & 54.5 & 52.5 & 27.3 \\
\end{array}
\]

\[\] the performance result in a thesaurus based
information retrieval system
\[\] the performance result in HYKIS

Figure 2 The performance comparison
of the two systems

5. Concluding Remarks

We already proposed a knowledge-based model
to information retrieval that utilizes most of
knowledge in IR environment, especially query
refinement and retrieval stage, and presented a
system HYKIS (HYbrid Knowledge-based Infor-
mation retrieval System), based on this model.
Accordingly, it not only enables an information
searcher to make a suitable query for his inter-
est without the aid of an expert or librarian, but
also leads to better retrieval effectiveness. It can
be accomplished by utilizing domain dependent
information as well as domain independent one
appropriately.

We implemented HYKIS in Sicstus Prolog and
demonstrated that it worked well as we predicted.
According to an experiment with the CACM
data set, it turns out that HYKIS achieves much
higher recall and precision rate than those of a
thesaurus-based retrieval system.

REFERENCES

computing reviews classification system - Fi-
nal version,” Communications of the ACM,
2. R. Yager (ed), International Journal of Intel-
ligent Systems, Vol. 4, No. 3, 1989, pp. 221-
402.
3. Y. Yoon, K. Choi, G. Kim, and D. Shin,
“A Hybrid knowledge-based Approach to In-
formation retrieval,” The Euromicro Journal:
Microprocessing and Microprogramming, Vol
4. Y. Yoon, An Information Retrieval Model
based on a Hybrid Knowledge-base, Ph.D.
dissertation, Computer Science Department
KAIST, January 1993.
5. D.W. Shin, Y.U. Yoon, and K.S. Choi,
“HYKIS - An Information Retrieval System
based on a Hybrid Knowledge base,” (To ap-
pear in the Second Internation Conference
on Information and Knowledge Management)