Chapter 5

A Prototype System

In order to confirm the effectiveness of the stem rules and the minimum coverage, a prototype system on a Windows NT environment have been implemented. The system includes a rule base and an user interface and currently supports a medium-sized document database of electric engineering documents. The number of all documents is 40,000 ($|\mathcal{D}| = 40,000$), and 16717 keywords are extracted from the document database ($|\mathcal{K}| = \rho(\mathcal{D}) = 16717$). Basing on the prototype system, the effectiveness of reducing the number of refinement candidates and improving result of retrieval will be shown in this chapter.

5.1 Interface

The user interface of the prototype system is shown in figure 5.1, and figure 5.2. An user's query will be refined in an interactive way. The prototype system displays the list of refinement candidates with respect to user's query. The user can reference the support attached to the rules to decide the keywords he/she wants to be included in his/her query. The list displayed supports conjunction form and disjunction form. Keywords in a path of the structure mean a conjunction form, and keywords in a same list mean a disjunction form.

Figure 5.2, figure 5.3, and figure 5.4 show a real example of query refinement support. The original query submitted by an user is "digital communication" that has a large result set of 810 documents. The refinement candidates with respect to "digital communication" are displayed on the right frame of screen. According to each candidate, a number shows the size of result set returned by conjunctively adding the keyword to the original query "digital communication". If the user wants to adds conjunctively refinement candidate keyword "picture communication" to the original query, the user may put the number in the left side of keyword and the system forms the query {"digital communication", "picture communication"} that retrieve 79 documents each of which contains "digital communication" and "picture communication". The result of retrieval is displayed in the title list(figure 5.2). If the user wants to continually refine the query, the user may put directly the string of keyword and the refinement candidates of query {"digital communication", "picture communication"} will be dis-

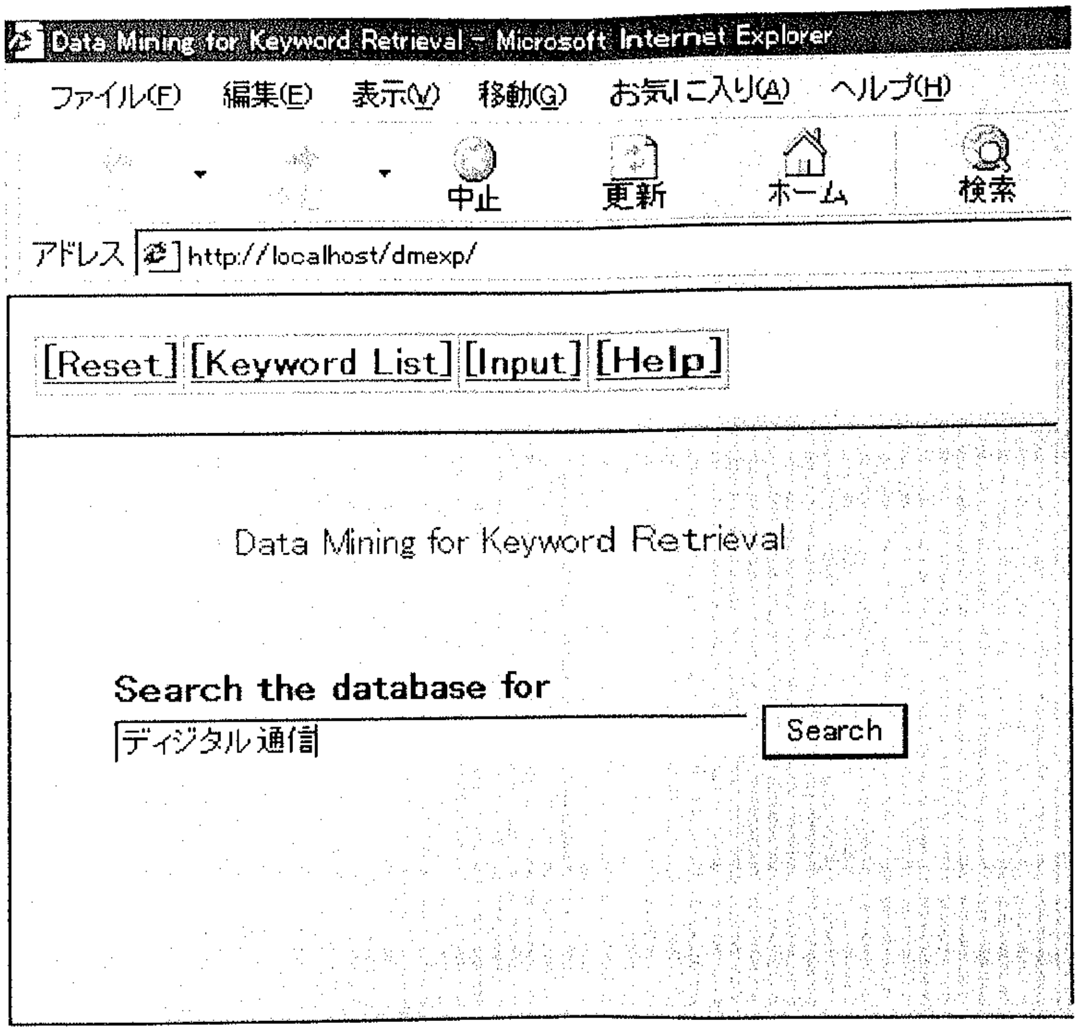
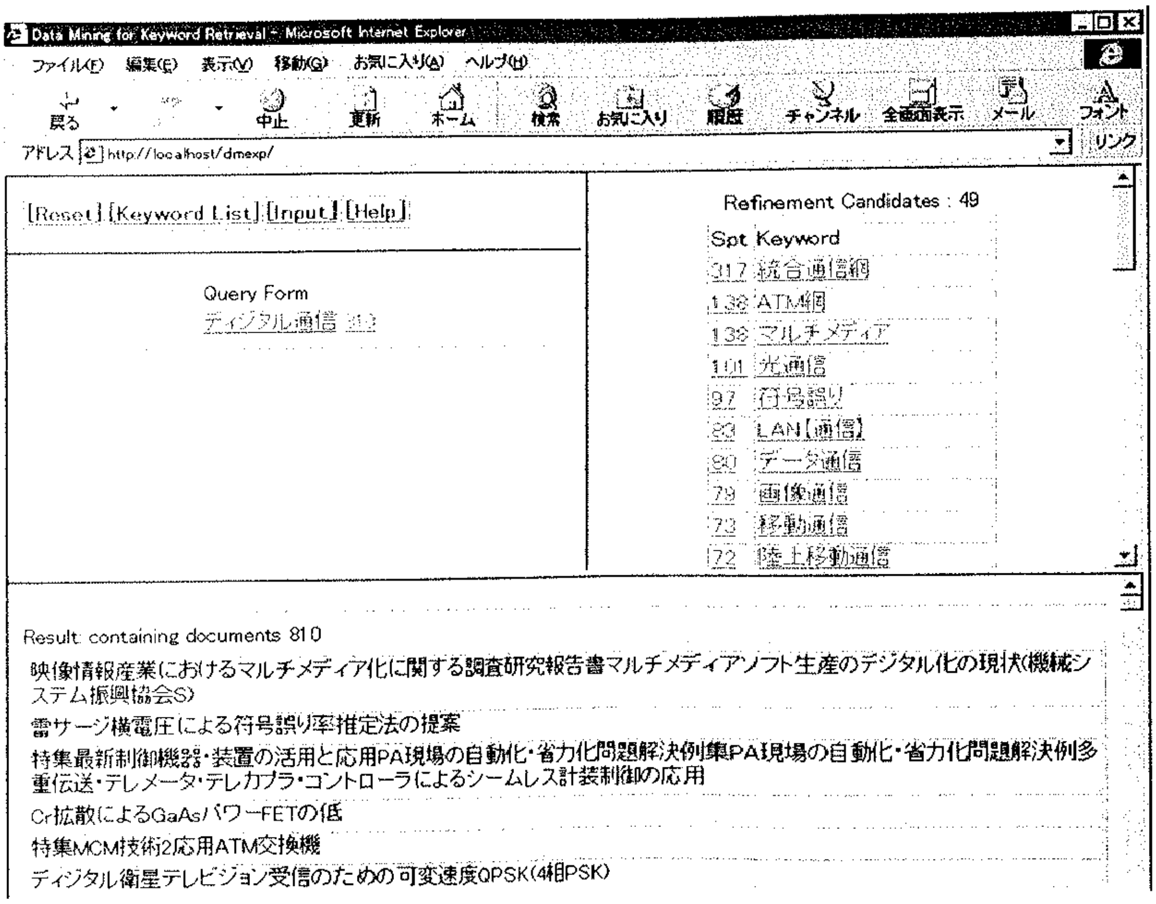

Data Mining for Keyword Retrieval - Microsoft Internet Explorer
ファイル(F) 編集(E) 表示(V) 移動(G) お気に入り(A) ヘルプ(H)
中止 更新 ホーム 検索
アドレス http://localhost/dmexp/
[Reset] [Keyword List] [Input] [Help]
Data Mining for Keyword Retrieval
Search the database for
ディジタル通信
Search

Figure 5.1: An example of query refinement support: Inputting the original query

played (figure 5.3). The path of query refinement is displayed in the left frame of screen and the user can put the string of keyword in path to return upper stage. This process will be carried out repeatedly until the user successfully refines his/her query (figure 5.4).


Data Mining for Keyword Retrieval - Microsoft Internet Explorer
ファイル(F) 編集(E) 表示(V) 移動(G) お気に入り(A) ヘルプ(H)
戻る
中止
更新
ホーム
検索
お気に入り
履歴
チャンネル
全画面表示
メール
フォント
アドレス
http://localhost/dmexp/
リンク
[Reset] [Keyword List] [Input] [Help]

Query Form
ディジタル通信 810

Refinement Candidates : 49

Spt	Keyword
317	統合通信網
138	ATM網
138	マルチメディア
101	光通信
97	符号誤り
83	LAN【通信】
80	データ通信
79	画像通信
73	移動通信
72	陸上移動通信

Result: containing documents 810

映像情報産業におけるマルチメディア化に関する調査研究報告書マルチメディアソフト生産のデジタル化の現状(機械システム振興協会S)

雷サージ横電圧による符号誤り率推定法の提案

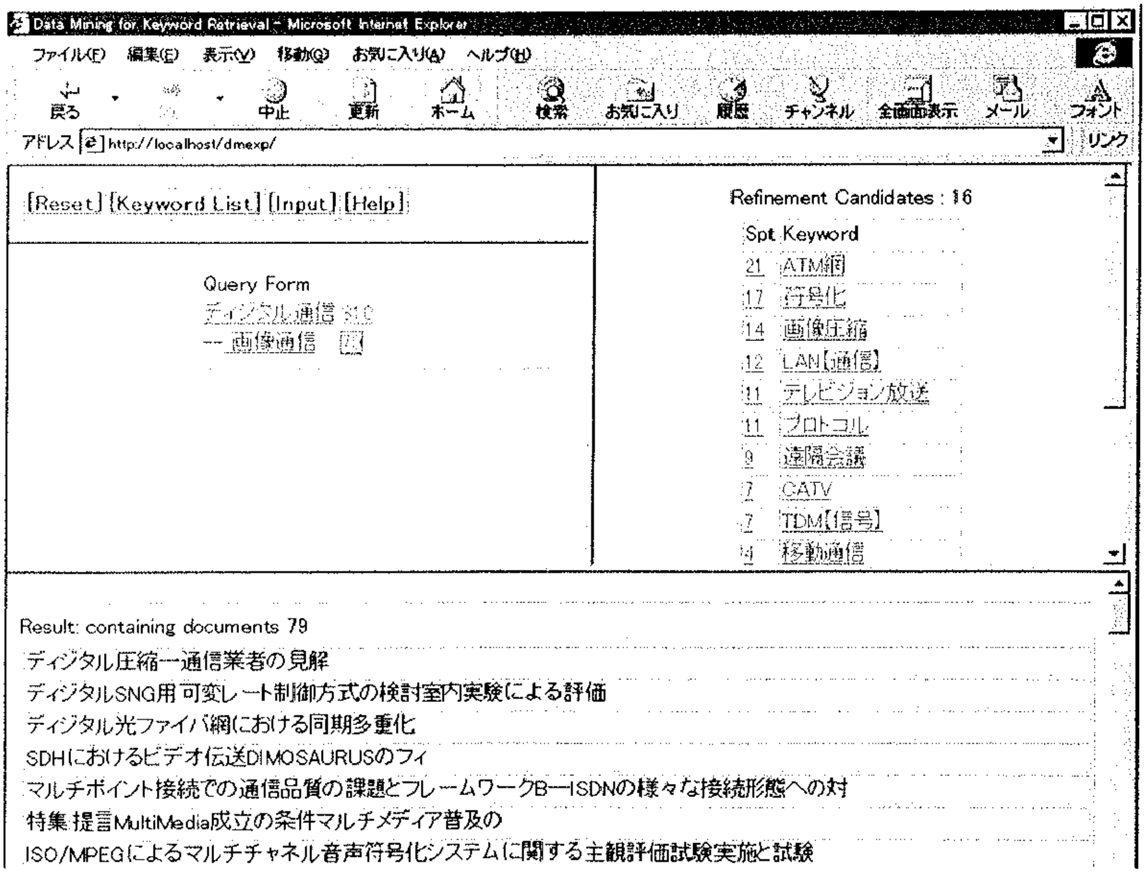
特集最新制御機器・装置の活用と応用PA現場の自動化・省力化問題解決例集PA現場の自動化・省力化問題解決例多重伝送・テレメータ・テレカプラ・コントローラによるシームレス計装制御の応用

Cr拡散によるGaAsパワーFETの低

特集MCM技術2応用ATM交換機

ディジタル衛星テレビジョン受信のための可変速度QPSK(4相PSK)

Figure 5.2: An example of query Refinement support: Showing the refinement candidates


Data Mining for Keyword Retrieval - Microsoft Internet Explorer
ファイル(F) 編集(E) 表示(V) 移動(G) お気に入り(A) ヘルプ(H)
戻る
中止
更新
ホーム
検索
お気に入り
履歴
チャンネル
全画面表示
メール
フォント
アドレス
リンク
[Reset] [Keyword List] [Input] [Help]

http://localhost/dmexp/

Query Form
ディジタル通信 310
-- 画像通信 79

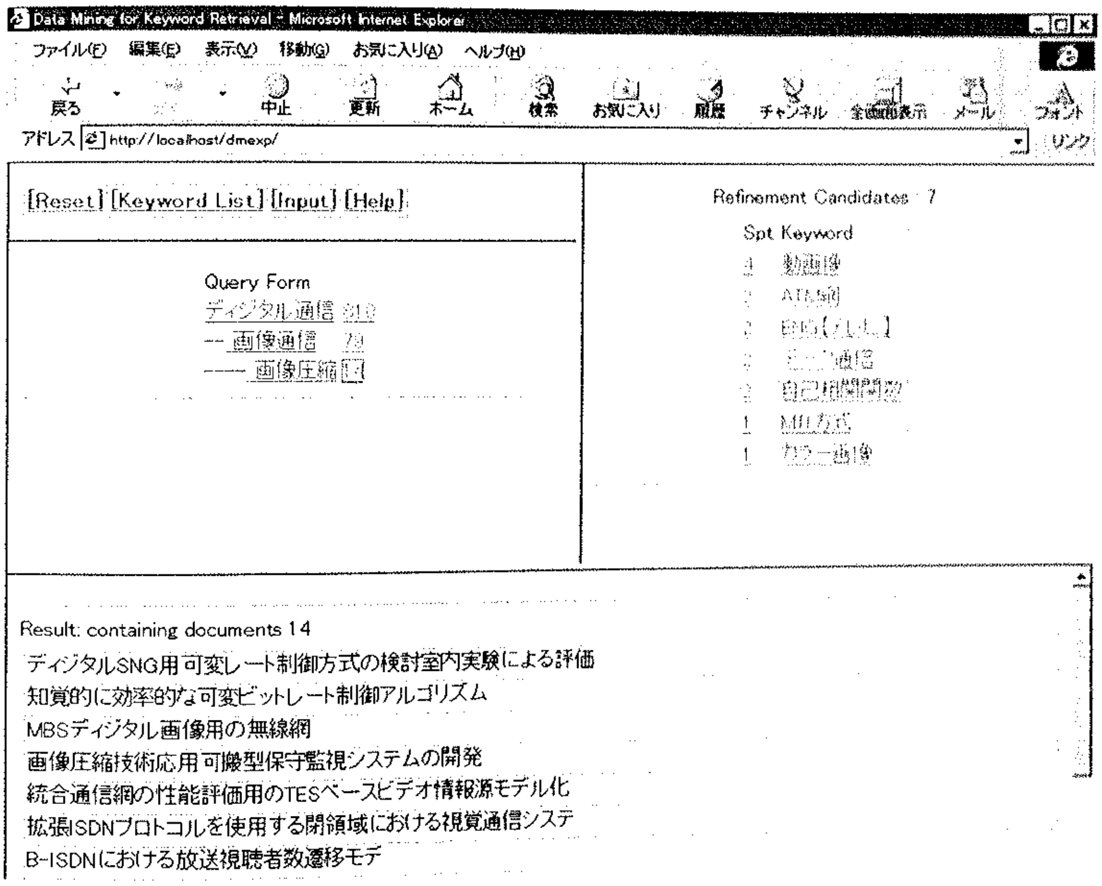
Refinement Candidates : 16

Spt	Keyword
21	ATM網
17	符号化
14	画像圧縮
12	LAN【通信】
11	テレビジョン放送
11	プロトコル
9	遠隔会議
7	CATV
7	TDM【信号】
4	移動通信

Result: containing documents 79

ディジタル圧縮―通信業者の見解
ディジタルSNG用 可変レート制御方式の検討室内実験による評価
ディジタル光ファイバ網における同期多重化
SDHにおけるビデオ伝送DIMOSAURUSのフィ
マルチポイント接続での通信品質の課題とフレームワークB―ISDNの様々な接続形態への対
特集 提言MultiMedia成立の条件マルチメディア普及の
ISO/MPEGによるマルチチャネル音声符号化システムに関する主観評価試験実施と試験

Figure 5.3: An example of query refinement support: One candidate chosen, and the next refinement candidates shown


Data Mining for Keyword Retrieval - Microsoft Internet Explorer
ファイル(F) 編集(E) 表示(V) 移動(G) お気に入り(A) ヘルプ(H)
戻る
中止
更新
ホーム
検索
お気に入り
履歴
チャンネル
全画面表示
メール
フォント
アドレス
リンク
[Reset] [Keyword List] [Input] [Help]

http://localhost/dmexp/

Query Form
ディジタル通信 310
-- 画像通信 79
---- 画像圧縮 [illegible]

Refinement Candidates 7

Spt	Keyword
4	動画像
2	ATM網
2	[illegible]
2	[illegible]通信
2	自己相関関数
1	MH方式
1	カラー画像

Result: containing documents 14

ディジタルSNG用可変レート制御方式の検討室内実験による評価
知覚的に効率的な可変ビットレート制御アルゴリズム
MBSディジタル画像用の無線網
画像圧縮技術応用可搬型保守監視システムの開発
統合通信網の性能評価用のTESベースビデオ情報源モデル化
拡張ISDNプロトコルを使用する閉領域における視覚通信システ
B-ISDNにおける放送視聴者数遷移モデ

Figure 5.4: An example of query refinement support: One candidate chosen, refined query decided

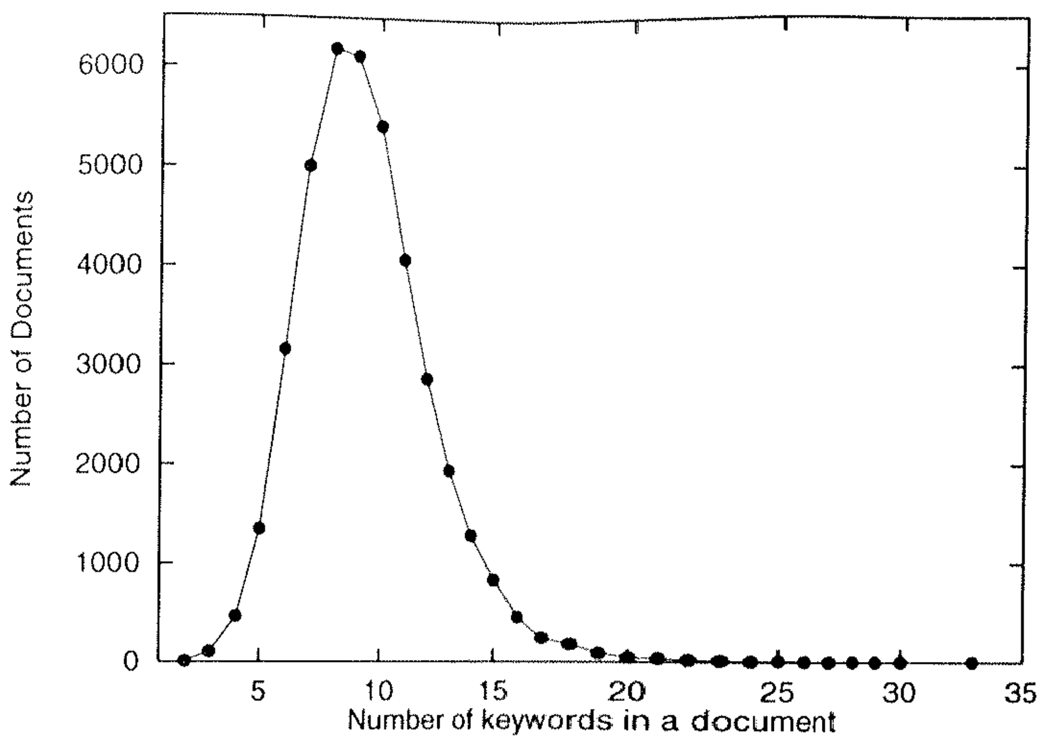


Figure 5.5: Statistics about number of keywords in a document

5.2 Data Set

The prototype system currently supports a medium-sized document database of electric engineering documents. The number of all documents is 40,000 ($|\mathcal{D}| = 40,000$), and 16717 keywords are extracted from the document database ($|\mathcal{K}| = \rho(\mathcal{D}) = 16717$). Though in chapter 3, the support of a keyword set q is defined as $Spt(q) = |\sigma(q)|$ documents, we use intuitively $|\sigma(q)|$ to stand for the support of q in the following.

We analyze the distribution of keywords in the document database.

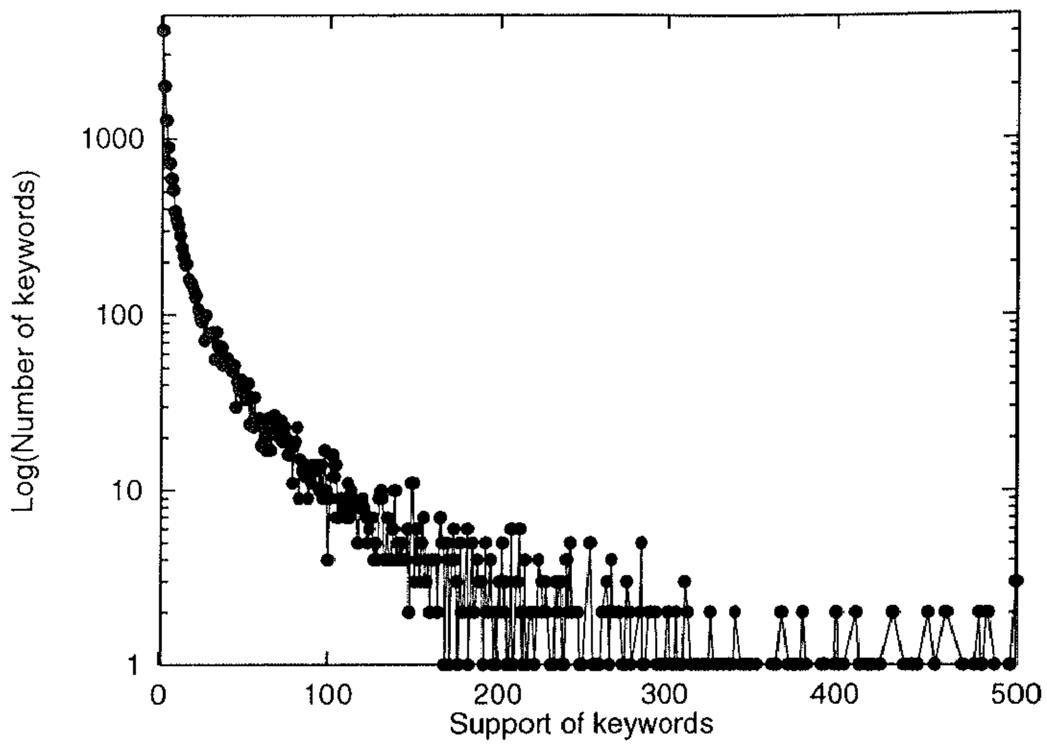


Figure 5.6: Statistics about support of keywords

The result is shown in figure 5.5, where the maximum number of keywords in a document is 33 and minimum number of it is 3. In other words, a document contains at least 3 keywords and at most 33 keywords. Most of the documents contain less than 15 keywords ($|\rho(d)| < 15$). Therefore, the rule base will be small because the stem rules are generated based on the combinations of $\rho(d)$.

The support of a keyword $Spt(k)$ is shown in figure 5.6. It can be seen that two-thirds of the keywords have supports less than 10 documents. These keywords will be not included in refinement candidates when minimum support is set to 10 by usual ARs. When a

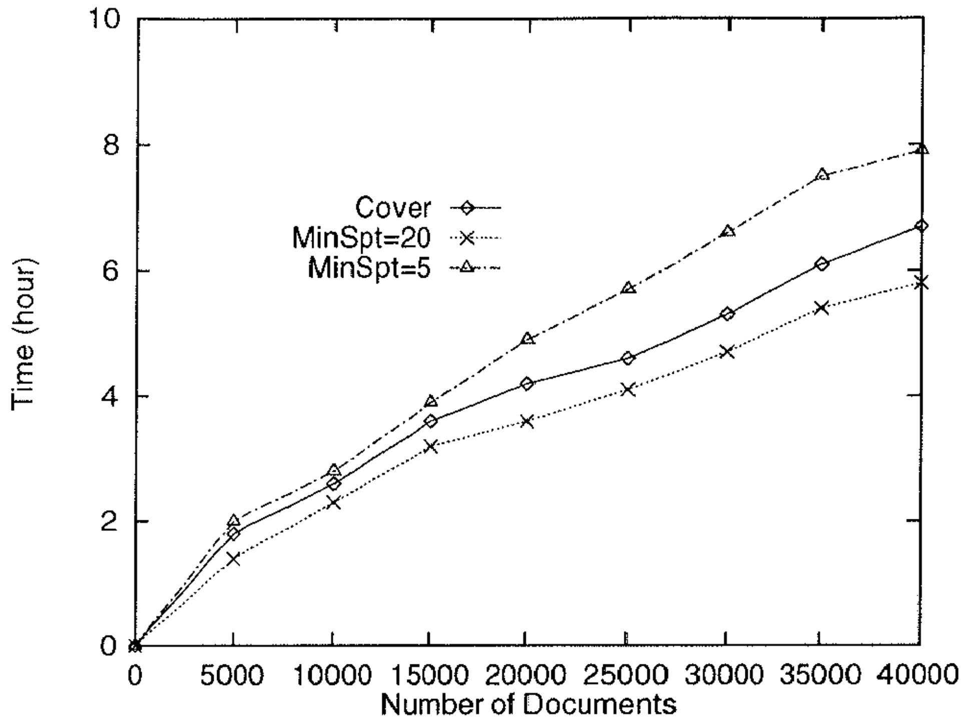


Figure 5.7: Time of generating rule base.

document only contains keywords with support being smaller than minimum support, the document will not be retrieved. Therefore, coverage is necessary that all documents have the chance retrieved when user refers the refinement candidates.

Figure 5.7 illustrates the time spent on generating rule base under the condition of setting minimum support to 5, 20 and the condition of coverage respectively. The environment of experiment is shown as follows.

OS:Windows NT

Database: SQL Server6.5

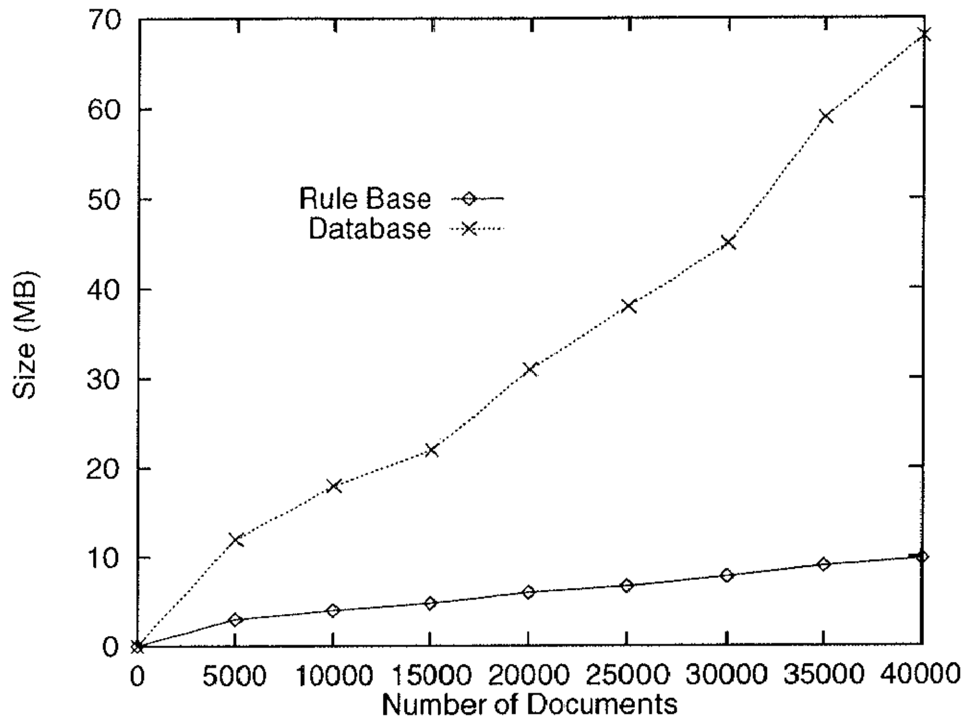


Figure 5.8: Size of rule base.

CPU: Pentium II 350

Memory: 256MB

Hard disk: 6GB IDE

Figure 5.8 shows the size of database and rule base respectively. The database includes title, keywords and relevant indexes of 40000 documents. The rule base is generated by the method of coverage. In [Frak92], a standard about evaluating storage efficiency is given as follows: "Storage efficiency is measured by the number of bytes needed to store data. Space overhead, a common measure of storage efficiency, is the ratio of the size of the index files plus the

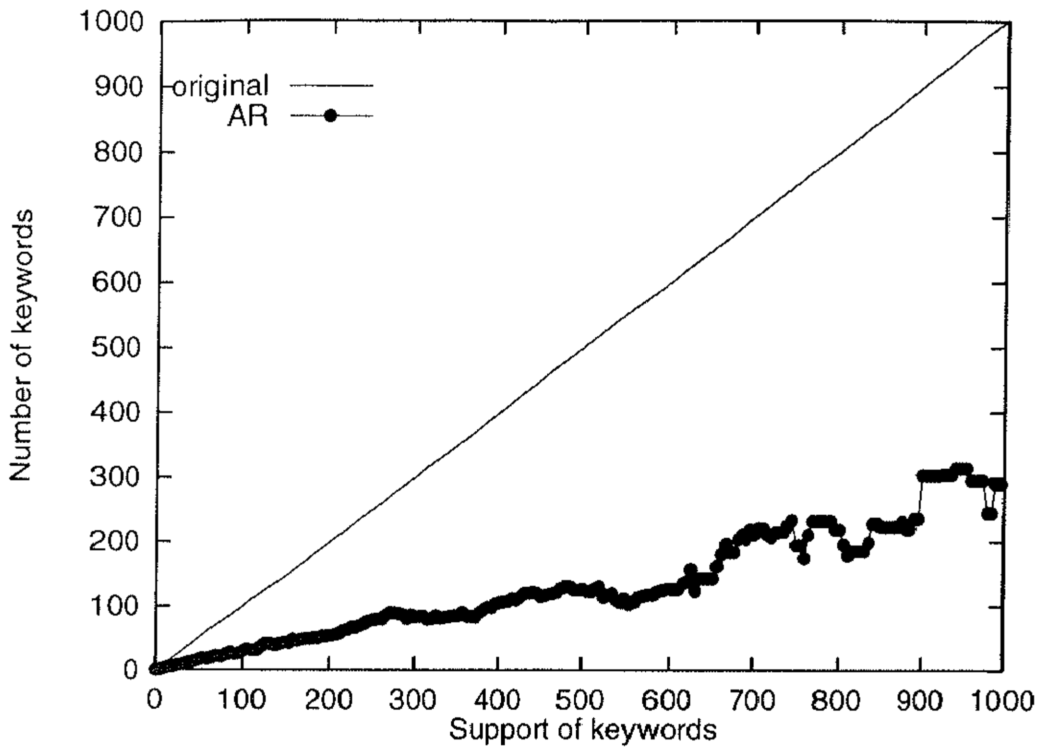


Figure 5.9: The average number of documents retrieved

size of document files over the size of the document files. Space overhead ratio of from 1.5 to 3 are typical for IR system based on inverted files". According to figure 5.8, the average space overhead ratio of rule base is about 1.2.

Lastly, we submit queries and investigate how many documents the queries will get. The "original" in figure 5.9 shows the average number of documents being retrieved without any assistance from the prototype system. Instead, the "AR" in figure 5.9 shows the average number of documents being retrieved using ARs. As can be seen, using ARs, the number of documents can be significantly

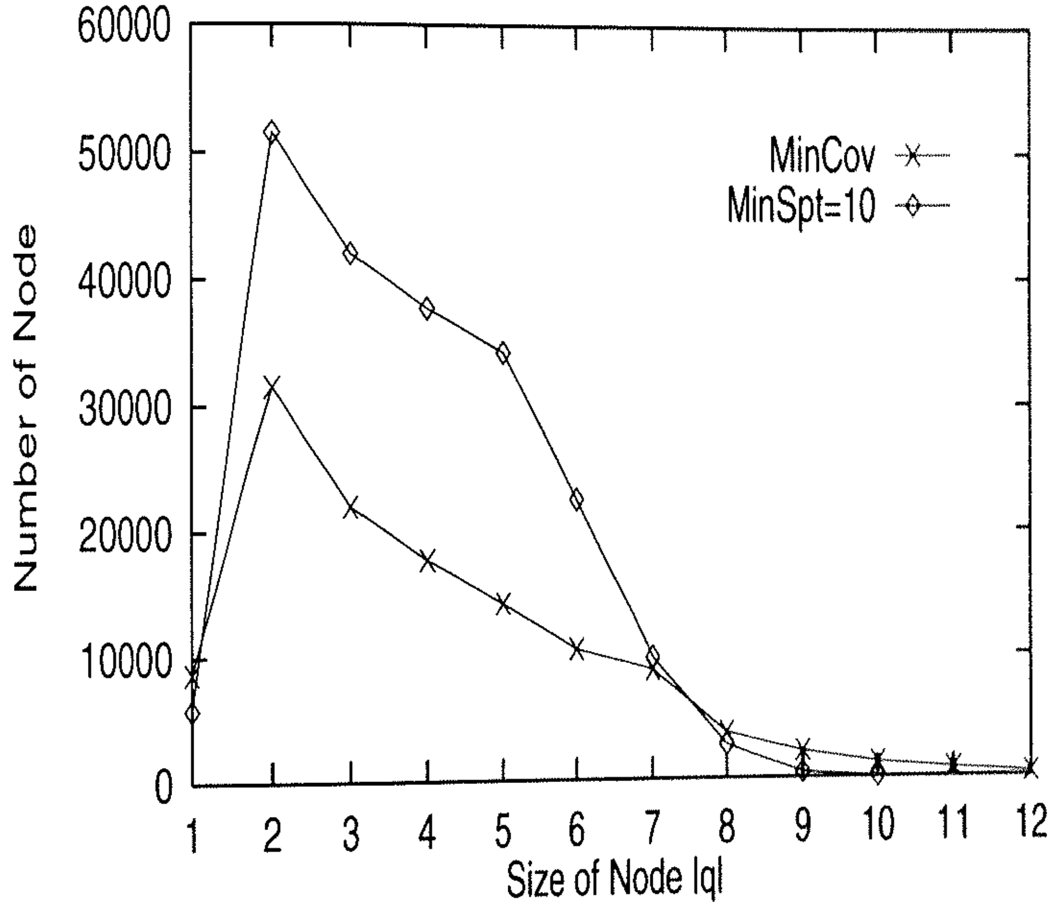


Figure 5.10: Number of rules generated.

reduced.

5.3 Evaluation

5.3.1 Number of Refinement Candidates

In this section, the effectiveness of reducing the refinement candidates by usual ARs, stem rule and minimum coverage is compared.

Usual ARs set thresholds($MinSpt$ and $MinCnf$) to a large

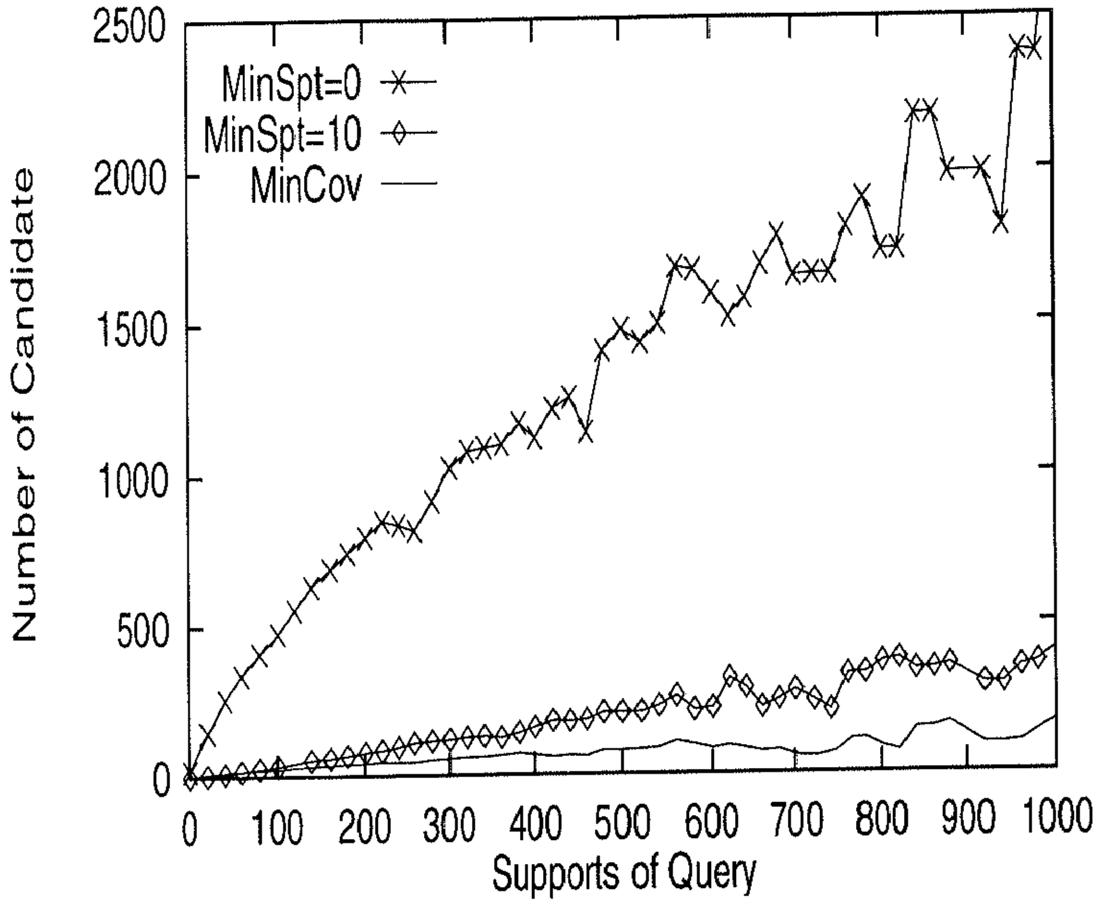


Figure 5.11: The supports v.s. the number of associated keywords.

value to reduce the number of rules. minimum support decide the number of Itemsets from which ARs are generated. In document databases, a large number of refinement candidates will be generated if a small $MinSpt$ is chosen. A large $MinSpt$ can reduce the number of refinement candidates but it may result that a set of refinement candidates does not cover original query.

The method of coverage uses minimum coverage as the condition of generating ARs instead of minimum support used in stem rule

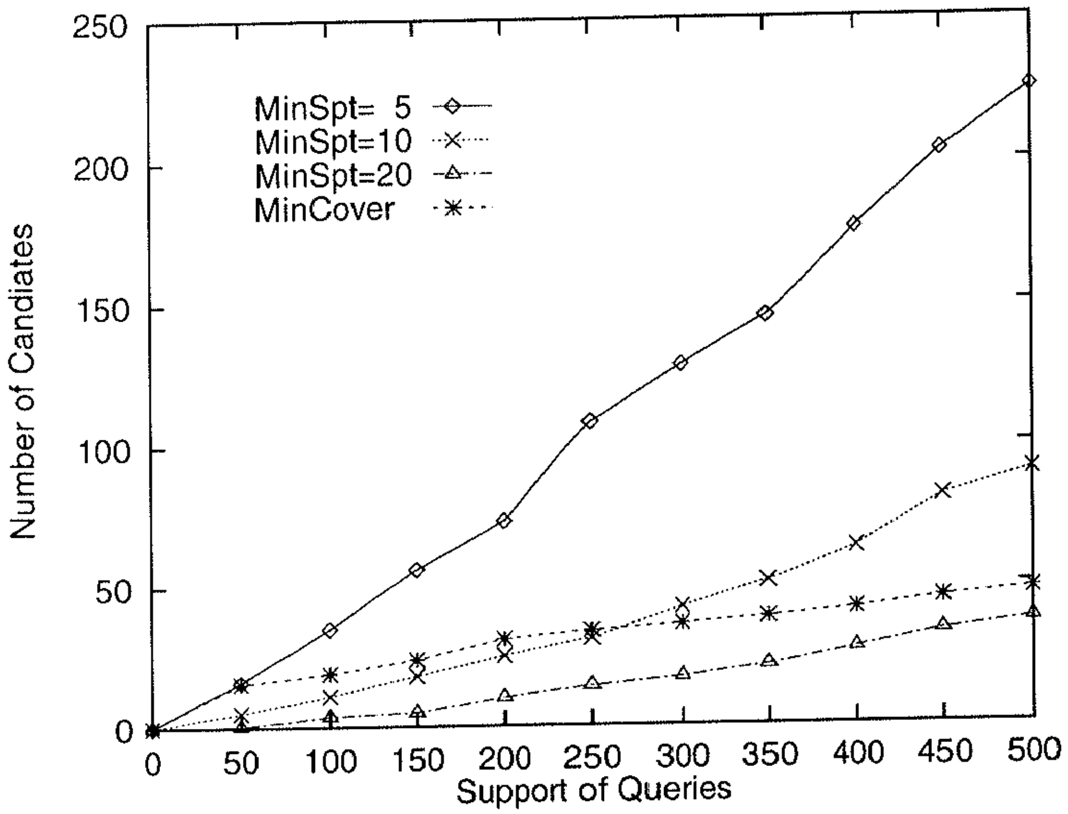


Figure 5.12: Number of relative keywords which large than minimum supports.

and usual ARs. Figure 5.10 shows that the number of Itemsets is also reduced using the condition of minimum coverage.

Figure 5.11 illustrates the numbers of the three kind of different refinement candidates with respect to supports of queries. In this figure, "MinSpt=0", "MinSpt=10" and "MinCov" are the relevant keywords based on co-occurrence, refinement candidates based on the stem rule and refinement candidates based on the minimum coverage, respectively. For an instance, when the support, $spt(kw)$, is set to 300, it can be seen that on average a keyword of support

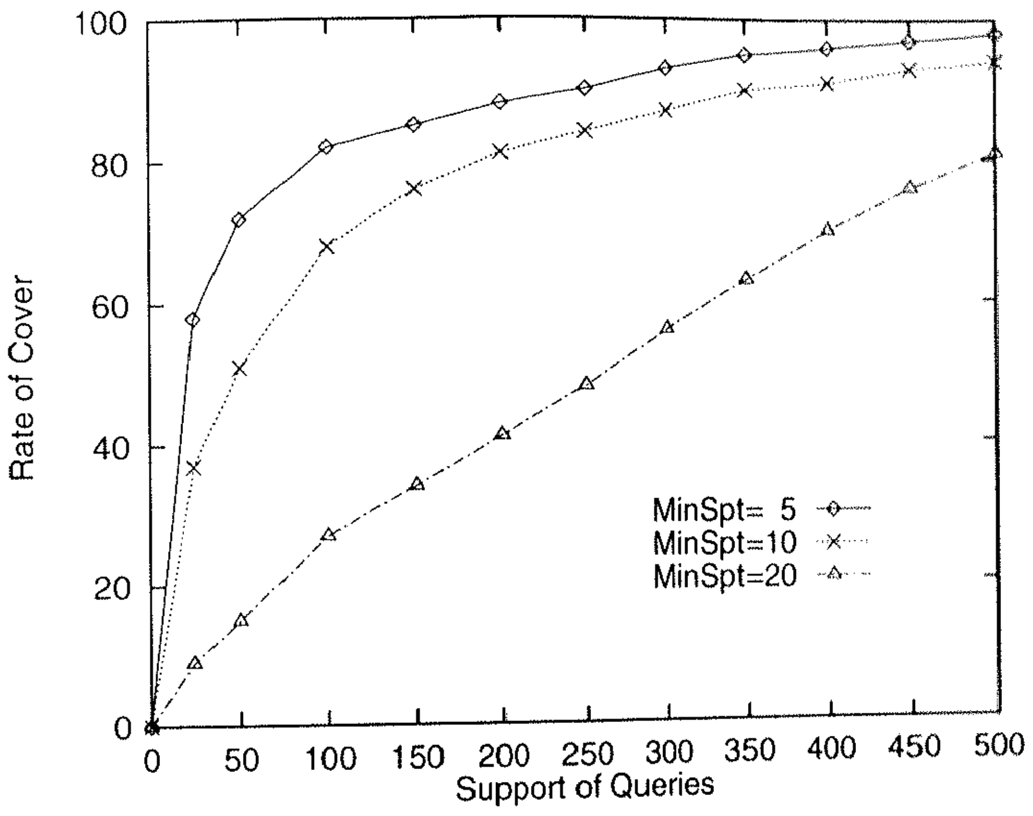


Figure 5.13: Rate of covering of refinement candidates.

300 will be associated with about 1000 other keywords. Using stem rules, the number of refinement candidates can be reduced to less than 10%. If coverage keyword candidates are used additionally, the number of refinement candidates is reduced to half of the refinement candidates generated by stem rules. This means that users can select from much less candidates to refine their queries.

Figure 5.12 also illustrates the numbers of the three different refinement candidates when keywords have certain supports. In this figure, the rules are computed under the condition of setting

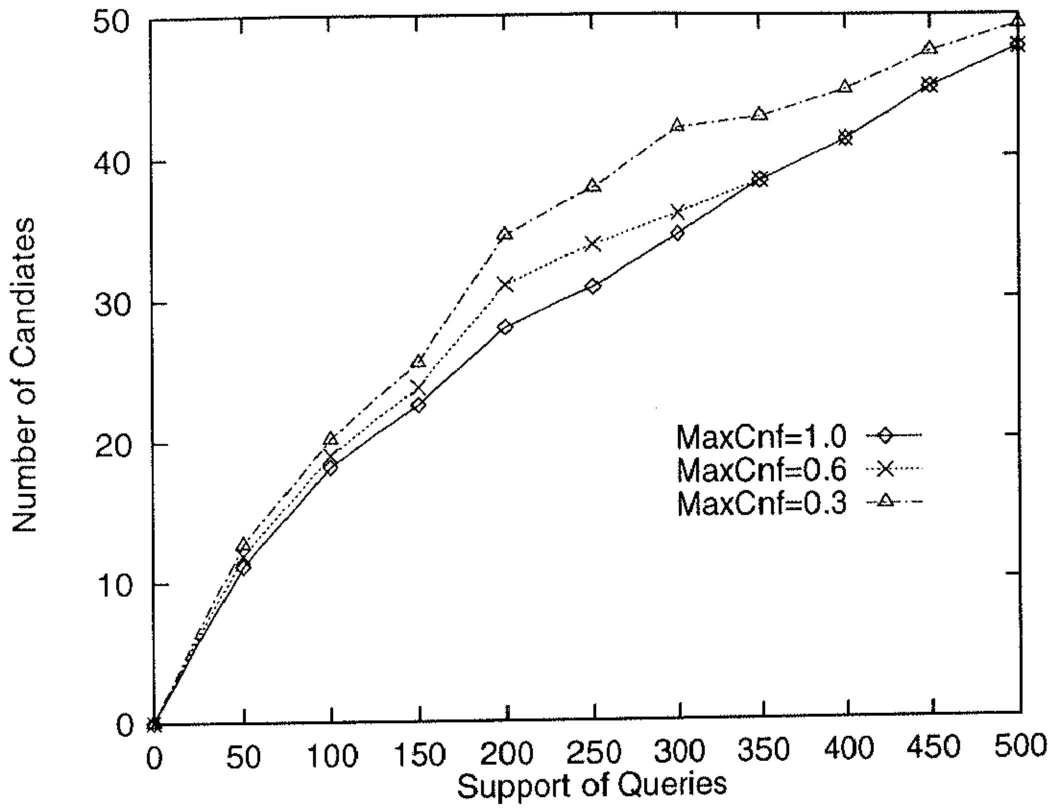


Figure 5.14: Effectiveness of various maximum confidence.

MaxCnf to 0.6 and setting *MinSpt* to 5, 10, 20 respectively and the condition of coverage.

The number of candidates under the condition of *Spt*= 5 is more than one under the condition of coverage. The number of candidates under the condition of *Spt*=10 is less than one under the condition of coverage within certain range. But The number of candidates under the condition of *Spt*=10 increases quickly outside certain range.

A large threshold of minimum support can reduce the number of refinement candidates but it may result that a set of refinement candidates does not cover all documents of original query. Figure 5.13 illustrates the rate of coverage of the three different condition of setting MinSpt to 5, 10, 20 respectively. It can be seen that the set of refinement candidates generated under the condition of minimum support can not cover original query. In order to increase the rate of coverage, a small value of minimum support have to be given. But this would increase the number of refinement candidates.

In the figures above, maximum confidence is a certain value 0.6. Figure 5.14 illustrates the number of refinement candidates under the three different $MaxCnf$ set to 1, 0.6, 0.2 respectively and the condition of coverage. The number of refinement candidates based on the condition of coverage have a little variety.

5.3.2 Result of Retrieval

As suggested in [Buck95], we also submit queries and investigate the relationship between precision and recall.

Two classical factors, *recall* and *precision* ([Naga90]), are often used to evaluate an information retrieval system. The two factors are defined as follows.

Let q be a query submitted by an user, and let D_t be an appropriate set of documents that should be needed for the user in

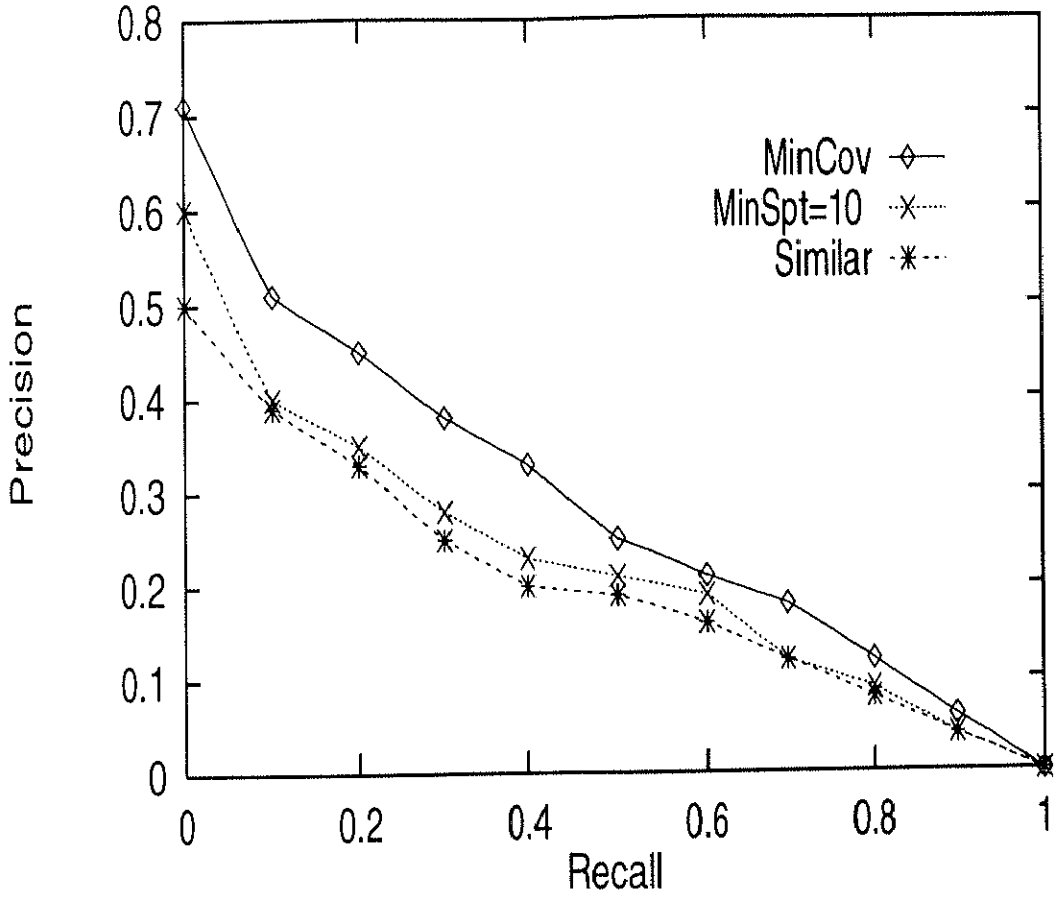


Figure 5.15: Comparison of Recall and Precision.

document database. Suppose that the system executes the query q, which retrieves a set of documents denoted $\sigma(q)$. The recall and precision are defined as $|\sigma(q) \cap D_t| / |D_t|$ and $|\sigma(q) \cap D_t| / |\sigma(q)|$, respectively.

A simulation of calculating recall and precision is supposed as follows.

Suppose that an author of a document is a user and D_r, reference documents listed by author, is a target set of documents that user wants to have. Let $\mathcal{D}$ be all documents in database. Then, $D_t = D_r \cap \mathcal{D}$ is a set of documents that should be needed for user in document database.

Figure 5.15 shows the superiority of this approach using ARs over both traditional data mining approach and the approach using similarity. In this figure, "Cover", "MinSpt" and "Similar" stand for the retrieval of using coverage approach, traditional data mining approach with minimum support and the approach of using similarity, respectively. It can be seen that "Cover" always has higher recall and precision than other two approaches because this system guarantees the coverage to be 100% and an user can interactively choose what he/she wants in order to refine his/her query. Fixing recalls, the precision of our approach are up to 15% higher than those of the other two. Fixing the precision, the recalls of our approach are up to 20% higher than those of the other two approaches.

Recently, [Véle97] independently notes the same problem we are targeting [Chen97] that too many documents will be retrieved by a query which is under-specified or contains ambiguous terms. The solution proposed in [Véle97] is an algorithms called RMAP which provides suggestions to refine the user query by calculating ranks of keywords.

In Comparison with the experiments in [Véle97], this system guarantees the coverage to be 100%. That is, we greatly reduce the refinement candidates a user can choose to refine his/her query without lost any documents he/she wants to access. On the other hand, unlike [Peat91, Salt90] which automatically refine an user query, our interactive interface enhances the precision because the user interactively chooses what he/she want to refine his query.