December 2005

Applying Association Rule Discovery to Select Laws and Articles for Lawsuit

Waiyamai Kitsana
Kasetsart University

Pongsiripreeda Teerawat
Kasetsart University

Applying Association Rule Discovery To Select Laws and Articles for Lawsuit

Waiyamai Kitsana
Facullty of Engineering, Kasetsart University
Bangkok 10900, Thailand
fengknw@ku.ac.th

Pongsiripreeda Teerawat
Facullty of Engineering, Kasetsart University
Bangkok 10900, Thailand
fengknw@ku.ac.th

Abstract

Data Classification and Association Rule Discovery are two important data mining techniques. In this paper, we apply the two techniques to find appropriate laws and articles for lawsuits. We propose a data classification method using a classifier generated from association rule discovery technique. We utilize the classifier to help selecting laws and articles to be tried for a lawsuit. Our experiment shows that the classifier generated from a proposed method yields better accuracy than one generated from a general data classification method. In addition, our method increases efficiency in multi-group and multi-level classification.

Keywords: Data classification, Association rule discovery, Data mining, Knowledge Discovery, Lawsuits

1. Introduction

Knowledge discovery from a very large database (KDD) [Waiyamai et al. 2000; Chen et al. 1996; Fayyad et al. 1996], or data mining, is a branch in computer science that gains high popularity nowadays. With KDD, large data is examined, analyzed, and organized into a knowledge base, which is used for retrieving information that cannot be discovered from an ordinary database method; for instance, finding relationship of data, and predicting an incoming phenomenon. The data mining techniques can add values in several areas; for example, using KDD in business to analyze products and markets, then promote a right product at a right timing, and categorizing groups of customers to increase service efficiency. These benefits will bring competitive advantages to an organization.

Both Data Classification [Chan et al. 1995; Gehrke et al. 1998] and Association Rule Discovery [Agrawal et al. 1993; Feldman et al. 1995] are important techniques of data mining. Classification rule mining aims to discover a small set of rules in the database that forms an accurate classifier. Association rule mining finds all the rules existing in the database that satisfy some minimum support and minimum confidence constraints. The main difference between two techniques is that Classification rule mining gives an exact number of results, while Association rule mining has no result limits. In this paper, we apply both techniques in law work by building a system to select appropriate laws and articles for a lawsuit.

To try a case, a lawyer has to investigate and judge from the details of each law. Current Thai laws are divided into Constitutions of Thai Kingdom and Acts, both of which have immense contents. It is a hard work to finish reading or memorizing all of them. A problem occurs when a lawyer examines or judges a lawsuit; he or she must select appropriate laws and articles from these constitutions and Acts. Generally, lawyers rely on their own experience in combination with searching similar cases and using them as a reference to

consider laws and articles. To do so, special skills and an amount of time are required but good results are not always guaranteed.

Several full-text-search [Witten et al. 1999] engines were built to facilitate lawyers on searching laws. The engines work by selecting old files that contain a searching word or phase. Until now, a result is somewhat unsatisfactory because a user cannot determine an exact word or portion of phase to produce an effective search result. General words or phases provide overwhelming search results, whereas detailed words or phases show few or no results. In addition, a user still wastes time in scanning several cases after search results until finally finds the one that matches his needs.

In Using Data Mining Technique to Select the Laws and Articles for Lawsuits [Pongsiripreeda et al. 2000], we utilized data classification technique in building a system to automatically select appropriate laws and articles in an objective to solve the addressed problems. We used 30,000 cases in the experiment. From the results, we summarize the system limitations as followed:

1) Word separation technique in preparation process still lacks enough efficiency to produces high accuracy search results.
2) The system cannot provide more than one law per each lawsuit.
3) The system can not simultaneously accommodate multi-level classification.

In Integrating Classification and Association Rule Mining [Liu et al. 1998], the experiment has shown that association rule mining techniques are applicable to classification tasks. In the spirit of this property, we propose a data classification method using a classifier generated from Association Rule Discovery technique. A law classifier model is built from a set of discovered association rules. We refer to this subset of rules as class association rules, or CARs.

Experimental results show that the classifier generated from the proposed method is more accurate than one generated from a common classification system. In addition, our data classification method helps to solve the addressed limitations in the classification systems.

Our method consists of three steps:

1) Pre-processing input data
2) Generating all the class association rules (CARs)
3) Building a law classifier model based on the generated CARs

2. Pre-processing Input Data

2.1 Input Data

The system's input, lawsuit data, is acquired from Thailand's Ministry of Justice. The data is exclusively formatted by BRS Search software. So far, 30,000 lawsuits filed during 1946 and 1998 A.C., are employed. Each lawsuit has indicated laws and articles used. On average, a lawsuit uses 1-2 laws, each of which refers to 2-3 articles. After collecting all data, we categorize laws into 20 groups and articles into 2200 groups.

2.2 Separating words and phases

Contents in each lawsuit are separated into small words and phases based on Thai dictionary's standards. In this process, we utilize Suffice Array technique [Bentley 1989] in breaking sentences into small pieces.

Suffix array technique is a pervasive technique to find association of data. We utilize the technique to find groups of longest character sets that appear repeatedly in the data. The

results, or groups of characters, are words and phases that we will apply in the next step, generating CARs.

Let A as a string, or array of characters, and S as suffix array of A. S is an array of pointers, which will point at each character of A. Therefore, member of S is sub-string that starts at the different character of A.

For example: A = banana
We will have

S[0] = banana
S[1] = anana
S[2] = nana
S[3] = ana
S[4] = na
S[5] = a

We then rearrange all members of S in ascending order. The result will be:

S[5] = a
S[3] = ana
S[1] = anana
S[0] = banana
S[4] = na
S[2] = nana

With rearranged S, we then have a group of longest character sets appearing repeatedly in the data, or A, that is *ana* as contained in S[3] and S[1].

```
1  FOR  m = 1  TO Count(S) - FNUM   DO
2       sl = 999;
3       FOR  i = m  TO   m + FNUM   DO
4            FOR   c = 1   TO   Length(Si)   DO
5                 IF   Si [c] <> Si+1 [c]   THEN
6                      IF   c<sl   THEN   sl = c
7                      break;
8                 END
9            END
10      END
11      IF  sl > LNUM  THEN ReturnPhase
        ( S 1..sl )
12  END
```

Figure 1: Algorithm for searching phases

Algorithm for finding phases from a group of arranged strings is shown in Figure 1.

- S is an array of rearranged sub-strings
- FNUM is a threshold or smallest frequency of a repeating phase required for a next process
- LNUM is a smallest number of characters in a phase
- sl is a variable used to count number of repeating characters in two comparing sub-string

The program compares a string for Count(S) – FNUM times to find a repeating set of characters (line 1). In each loop, the variable *sl* is initialized to a possible maximum number, 999 (line 2). Then each string is compared with next adjacent strings (sub-loop in line 3). This works by comparing each character from the beginning in two strings (line 4 to 9). When the difference occurs between both strings, the position of distinction is kept in *sl*. After finishing the loop in line 2, *sl* is the highest number of distinction position. If *sl* is higher than LNUM, a character from 0 to *sl* position will be selected as a phase in a dictionary.

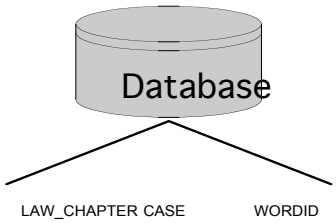
We apply the technique above with input data by categorizing data into 70 groups. Each group has 100 lawsuit data supported. Then, we load lawsuit data of each group into memory to create an array of character (A). In this process, A is a set of concatenated characters, whose size is approximately 200 Kbytes.

We use an array of pointer S pointing to each member of A. After we reorder S with a quick sort method, the output is small, 200K sub-strings. Using algorithm in Fig. 1, we find a group of longest characters or phases.

The most time-consuming process is comparing string. To improve the speed, we first prepare string A with Thai dictionary's spelling check and let array of pointers S pointing to only each first character of a word in A. This method decreases the size of S by 70 percent and the process is much faster.

After loading and processing all 70 groups, we then have 23,722 phases, which will be utilized as a dictionary for a next step.

2.3 Transforming Data



LAW_CHAPTER	CASE	WORDID
01_249	2489_0002	17566
01_249	2489_0002	14690
01_249	2489_0002	14690
01_249	2489_0002	9830
01_249	2489_0002	20880
01_249	2489_0002	5189
01_249	2489_0002	9231
01_249	2489_0002	21113
01_249	2489_0002	682
01_249	2489_0002	5138
01_249	2489_0002	11008
01_249	2489_0002	5189
01_249	2489_0009	15651
01_249	2489_0009	21122
01_249	2489_0009	5189
01_249	2489_0009	1532
01_249	2489_0009	21549
01_249	2489_0009	1913
01_249	2489_0009	284
01_249	2489_0009	23144
01_249	2489_0009	7889
01_249	2489_0009	22295
01_249	2489_0009	2846

Figure 2: Structure of transformed data

The phase dictionary from 2.2 is kept in a database and each phase of the dictionary has an ID specified. After gathering lawsuit data into a database, we use the phase dictionary to separate the contents. The outputs are IDs referenced to words and phases appeared in a lawsuit, law codes, and article codes tried. Structure of the database is shown in Figure 2. It consists of law code, law article, case ID, and word or phase ID.

3. Generating Class Association Rules (CARs)

Using Association Rule Discovery technique, we can find rules representing data association. The rules discovered must satisfy some MinSupt and MinConfd constraints. We call these rules as association rules and their format is as follows:

{item1, item2} → item3 Supt% Confd%

- Supt (Support) is percent of items applying to the rule
- Confd (Confidence) is percent of items corresponding to the rule out of all items applies to the left-hand-side.
- MinSupt (Minimum Support) is a smallest support number to accept the rule
- MinConfd (Minimum Confidence) is a smallest confidence number to accept the rule

The rule implies that, with a support of Confd percent, when item1 and item2 exist in the data, item3 will also exist.

Class Association Rules (CARs) are rules generated from Association Rule Discovery process. However, CARs show the relationship of data to the category. CARs' format is as follows:

{item1, item2} → class Supt% Confd%

We apply the rule in law work using 70% of prepared data to teach the system to learn classifying data, and find association rules of phases and laws used in lawsuits. In this case, the following rule will return laws used in a lawsuit:

Phase → law Supt% Confd%

We set a left-hand-sided parameter as a phase, not a set of phase, because the current system employs almost 30,000 phases, causing a tremendous processing time in finding rules. In addition, phases consist of several small words and are considering lengthy. Therefore, we disregard the process of finding rules for set of phases.

In this experiment, we set minimum support and minimum confidence to 0, due to the fact that data is highly dispersed and confidence value is low. Nevertheless, rules found from this step will be selected and filtered before being utilized for generating a classifier in the next step.

For example, we form the relationship of phase W, law code L, and article C to the table as follows:

Phase	Law	Article
W1	L1	C1

W1	L1	C2
W1	L2	C3
W2	L3	C4
W2	L3	C4

Each row represents a transaction. The experiment is required to classify input into two types of output: law code, and article. This constraint forces the system to generate two sets of rules: law-level rules and article-level rules. From the table, we can find following rules.

Rules	Supt	Confd
W1 → L1	40.00%	66.67%
W1 → L2	40.00%	33.33%
W2 → L3	40.00%	100.00%

Law-level CARs

Rules	Supt	Confd
W1 → C1	20.00%	33.33%
W1 → C2	20.00%	33.33%
W1 → C3	20.00%	33.33%
W2 → C4	40.00%	100.00%

Article-level CARs

We can calculate Supt value of rule W1→L1 from number of transaction having W1 and L1, 2, divided by number of all transaction, 5, then multiplied by 100. The result, or support value, is 40%. Also, we can calculate Confd value of rule W1→L1 from number of transaction applies to the rule, 2, divided by number of transactions having W1, 3, then multiplied by 100. The result, or confidence value, is 66.67%.

```
1    f = 0
2    FOR   w = 1   TO   Count(Word) DO
3         GetDataOfWord(w)
4         FOR   i = 1   TO   Count(Data)   DO
5           IF  (i > 1) and (Class_i <> Class_i-1)
                THEN
6                   supt = f / Count(All)*100
7                   conf = f / Count(Data) * 100
8               ReturnRule( W → Class , supt,
                    conf)
9                   f = 0
10            ELSE
11                  f = f+1
12            END
13        END
14   END
```

Figure 3: Algorithm for generating CARs

Generating CARs has the algorithm as in Figure 3. Initially, a program sets a frequency variable, f, to 0 (line 1). Then, for a number of phases, or Count(Word), in dictionary a program executes the following procedures (line 2). The program calls function GetDataOfPhase (line 3) to retrieve an array of law code and article code used in the specified phase code, or w. The array is ordered by law code and article code. In the algorithm, a unique law code is referred as Class. (Article code is ignored only in this case as a purpose to illustrate an easy process of the algorithm.) Consequently, for a number of Count(Word) times, the program counts frequency of law code, or f, used in a phase (from line 4 to 13), by counting number of repeating class. The inner loop works by checking the different class in the array. When a new class is found (line 5), the information of previous class, which consists of a support value, confidence value and a rule, will be returned to the system (line 6 to 8) and the frequency value of the new class will start as 0 (line 9). The rule created (line 8) is considered a new rule in law-leveled CARs.

Generating article-leveled CARs follows similar algorithm as in Figure 3. The difference is that the variable class used in line 5 and 8 becomes article code instead of law code.

4. Generating Classifier from Class Association Rules (CARs)

We use CARs generated from a prior step to create a classifier. To do so, we arrange CARS of each phase in sequence from the highest precedence value. Precedence value is considered from confidence and support value, using the following logic:

If rule1's confidence value is more than rule2's, rule1 has a higher precedence value.

If rule1's confidence value is equal to rule2's and rules1's support value is higher than rule2's, rule1 has a higher precedence value.

If both rules' confidence and support values are equal, the prior rule has a higher precedence value.

As CARs are kept in the database, we use a SQL statement below to arrange CARs in order from the highest precedence.

```
SQL:  SELECT Rules FROM CARs
      ORDER BY Confd DESC, Supt DESC
```

As a result, the reordered CARs is a classifier, which we refer as Class Association Rules Sorted with Precedence, or CARsp.

5. System Implementation

When a new case enters a system to predict laws and articles to be tried, the system will follow 2 steps:

Step 1) Separate contents with a phase dictionary created from 2.2. The output is a group of numbers referenced to each phase of a case.

Step 2) Use CARsp to determine rules from every phase. The rules will be ordered by precedence value executed by an SQL statement below.

```
SQL:  SELECT Rules FROM CARs
      WHERE Word IN (W1, W2, W3, …)
      ORDER BY Confd DESC, Supt DESC
```

The rule with highest precedence value, or the first entry from SQL, will show a law to be considered for a lawsuit.

6. Experimental Summary

We use the rest of prepared data, or 30% of total lawsuits, to test system accuracy and compare the results with those of the old system using general data classification technique (Decision Tree [Gehrke et al. 1998]). We have two tests for law-level and article-level results. Law-level test will classify data into 4 groups of law while article-level test will classify data into 70 groups of article. The experimental result is shown in graphs in Figure 4.1 and Figure 4.2. Both graphs' Y-Axis represents percent of classification accuracy while X-Axis represents law and chapter code, respectively.

The graphs in Figure 4.1 and 4.2 show that the new classifier generated from a new method yields a better result than the one generated from general data classification method. In law-level test, the result indicates that the new classifier has higher accuracy in every law group. Meanwhile, in article-level test, the accuracy percents of both classifiers are relatively close. However, the new classifier can classify more articles than the old one can. The old classifier can only give 0% accuracy when applied to classify data into more than 20 chapters.

Comparing the accuracy of both systems in chapter level, the graph in Fig 4.2 indicates that both systems produce a consistent result. Both systems give low and high accuracy with the same articles. High accuracy with some articles can be explained by containing some specific or outstanding phases, while low accuracy with some articles results from not containing any specific words or phases, which make difficulty in distinguishing an article from the others.

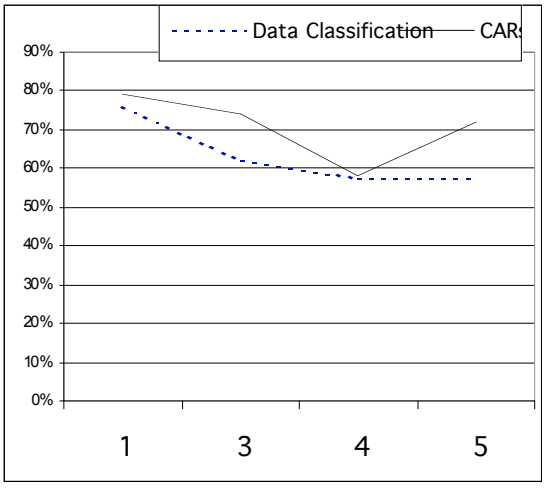


Figure 4.1: Graph comparing law-level accuracy between a classifier generated from general data classification technique and one generated from law-level CARs.

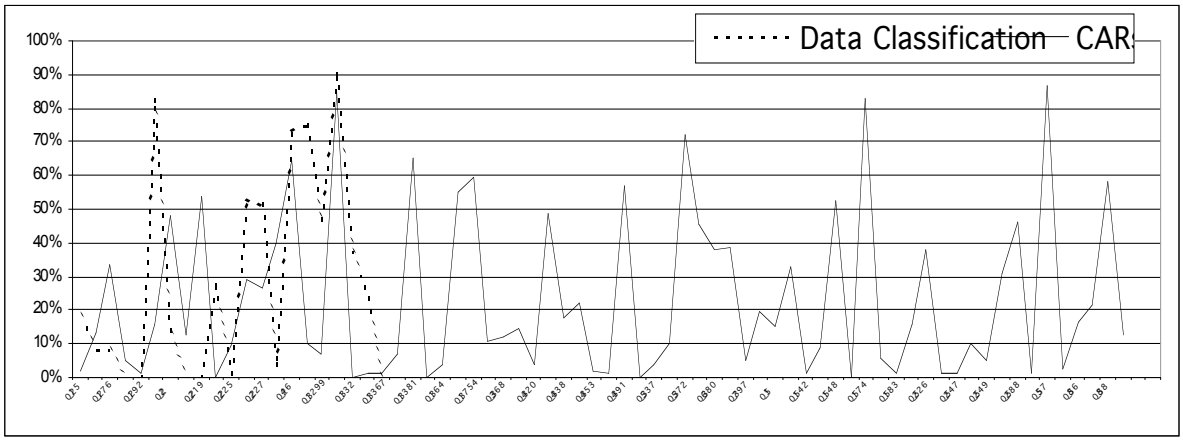


Figure 4.2: Graph comparing article-level accuracy between a classifier generated from general data classification technique and one generated from law-level CARs.

There are two assumptions why the new system yields better accuracy than the old system.

First, integrating Association Rule Discovery technique in generating a classifier, and changing method to separate phases instead of words. To support our assumptions, we retest and adapt the old system, which utilizes general Data Classification technique, to apply the new method of separating phases. The result is shown in Figure 5. Y-Axis represents percent of accuracy from classification process and X-Axis represents law chapters.

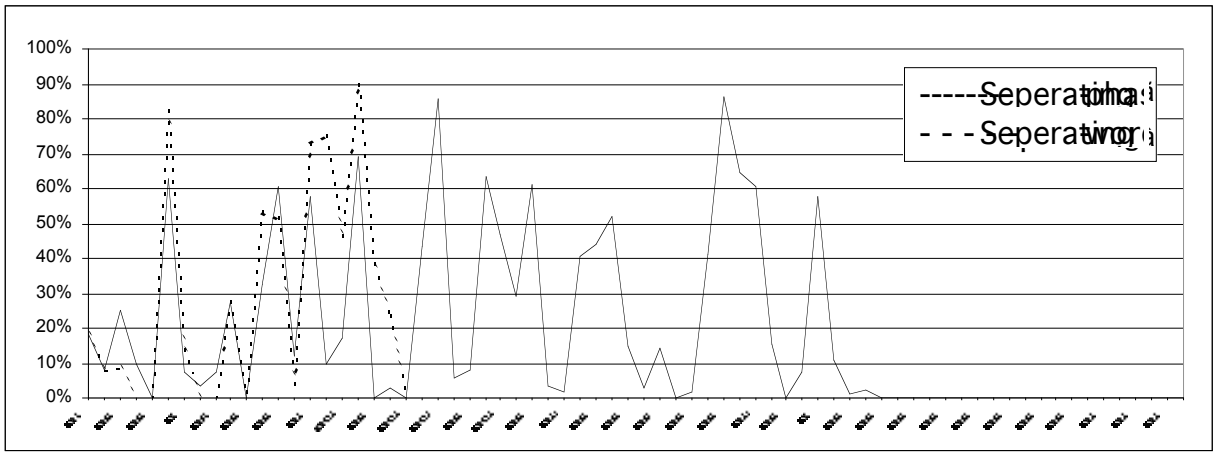


Figure 5: Graph comparing accuracy between the classifier generated from a separating-word method and the one generated from a new separating-phase method

The graph in Figure 5 shows that the new method yields relevant accuracy to the old one when compared in chapter-level. However, the new method can classify more chapters.

Generally, each lawsuit needs more than 1 law and chapter to be considered. General Data Classification technique cannot accommodate this requirement. After changing the method to use CARs to generate a classifier, we can solve this problem. Instead of picking only a single rule with highest precedence value from CARs, the system can select 5 or 10 rules each time. This method allows the system to predict the classification more than 1 groups. Also, it gives better accuracy, as shown in Figure 6.

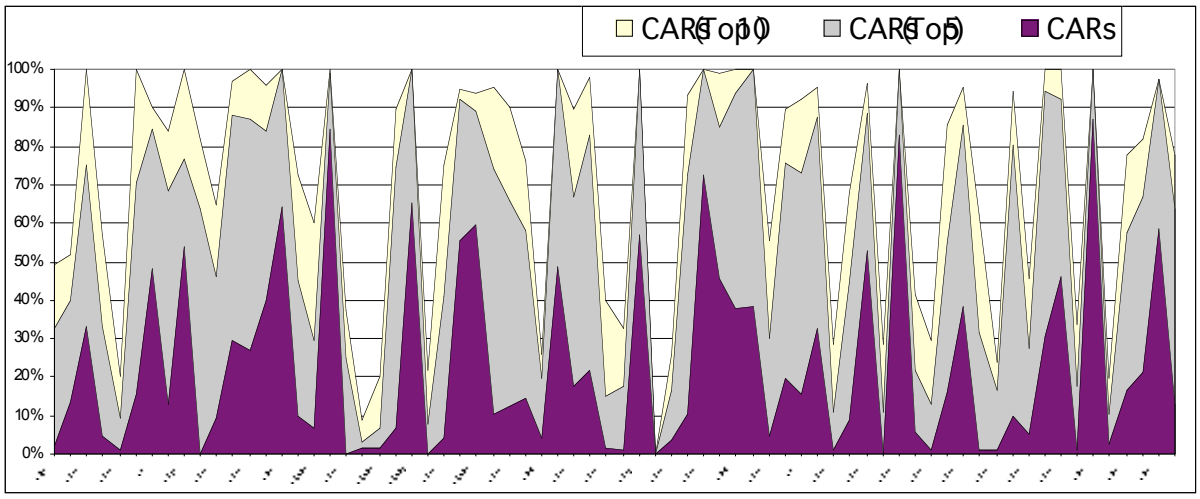


Figure 6: Graph comparing percents of accuracy when classifying results by 1, 5, and 10

Another aforementioned problem about multi-level classification, which cannot be solved from a general Data Classification technique, forces the old system to generate two classifiers and use them separately to process the data twice. For the new system, two classifiers generated from CARs can process the input at one time, given that when a law-level precedence value is too low, an article-level rule will be automatically selected.

The comparison result of different classifiers is shown in a graph in Figure 7. X-Axis represents classifiers while Y-Axis represents average accuracy percent.

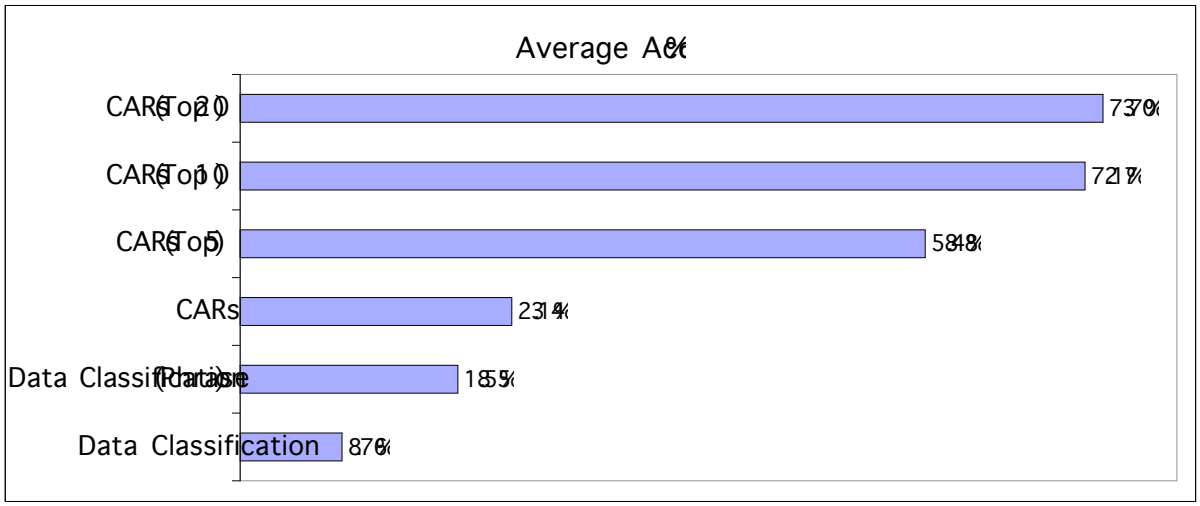


Figure 7 Graph comparing accuracy percents of several classifiers

7. Summary

The graph in Figure 7 clearly shows that this experiment can improve efficiency and accuracy of the technique of generating a classifier. In addition, the new system can solve the problems and limitations of the old system as followed:

1. Output limitation, or number of group of data the system can categorize. The old system can predict only 20 from 70 articles, while the new system can predict every article.
2. The old system's inability to predict more than 1 article for each lawsuit. The new system, using the new generator from CARs, solves this problem by setting the number of article as output for each search. This technique also increases the average system accuracy.
3. Multi-leveled classification. As the old system cannot process law-leveled and article-leveled prediction at the same time, the new system can do so with a single search. Using two separate CARs, the new system will check if law-leveled results have too low accuracy percent, the system will select only article-leveled results.

In this experiment, the system focuses on automatically predicting Thailand's laws and articles. Therefore, the design and technique are exclusive and specific. To implement the technique to other systems or environments requires several procedural and technical modifications. In addition, even though our proposed system still does not give highly accurate result, there is a room of improvement by either using the new technique to generate a classifier or modifying the data preparation process to increase the system's efficiency.

8. References

Waiyamai, K., and Lakhal, L. 2000. "*Knowledge Discovery Very Large Database using Frequent Concept Lattices*". Springer-Verlag Lecture Notes in Artificial Intelligent (1810), 2000, pp. 437-445.

Chen, M.S., and Han, J., and Yu, P.S. Data Mining: "*An overview from a database perspective*". IEEE Trans. Knowledge and Data Engineering, 8:866-883, 1996.

Fayyad, U.M., and Piatetsky-Shapiro, G. and Smyth, P. and Uthurusamy, R. "*Advance in knowledge Discovery and Data Mining*". AAAi/MIT Press, 1996.

Witten, I.H., and Moffat, A. and Bell, T.C. "*Managing Gigabytes: Compressing and indexing Documents and Images*". Morgan Kaufmann Publishers, 1999.

Chan, P.K., and Stolfo, S.J. "*Learning arbiter and combiner trees from partitioned data for scaling machine learning*". In Proc. 1st.Int.Conf. Knowledge Discovery and Data Mining (KDD'95), 1995, pp. 39-44.

Gehrke, J. and Ramakrishnan, R. and Ganti, V. "*Rainforest: A framework for fast decision tree construction of large datasets*". In Proc. 1998 Int. Conf. Very Large Datacase, 1998, pp 416-427.

Agrawal, R., and Imielinski, T. and Sawami, A. "*Mining Association Rules between sets of items in large database*". SIGMOD'93, 1993, pp 207-216.

Feldman, R. and Dagan, I. "*Knowledge Discovery in Textual Database*" In Proceedings of the International Conference on Knowledge Discovery and Data Mining (KDD'95), 1995.

Pongsiripreeda, T., and Waiyamai, K. "*Using Data Mining technique to select the laws and articles for lawsuits (in thai language)*". The National Computer Science and Engineering Conferrence (NCSEC'2000). 2000.

Liu, B. and Hsu, W. and Ma, Y. "*Integrating Classification and Association Rule Mining*". In Proceedings of the International Conference on Knowledge Discovery and Data Mining (KDD'98), 1998.
Bentley, J.L. "*Programming Pearls*". Addision-Wesley, 1989, pp. 164-167.