http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 5 (6): 1023-1027, 2006
ISSN 1812-5638

Efficient Information Extraction Using Machine Learning and Classification Using Genetic and C4.8 Algorithms

[1]A. Christy and [2]P. Thambidurai
[1]Sathyabama Deemed University, India
[2]Pondicherry Engineering College, Pondicherry, India

Abstract: With the amount of information available on the internet growing at phenomenal rate, research in improving the effectiveness and efficiency of information extraction and knowledge discovery has become crucial. Text mining is one of the most important ways of extracting meaningful information from a large collection of text documents, leaving aside the information which is not useful to the ordinary user. In this study, we propose a method for automatically extracting key elements from a collection of text documents by extracting a set of features using a machine learning technique. We have used the Genetic algorithms for classifying the features those are selected by the machine learning technique. We also compared the results produced by the Genetic algorithm with 10 folds cross-validation at C4.8, Rain Forest, Raintree and NB Tree methods and we have found C4.8 has produced better precision and recall and also the Genetic algorithm is an effective classifier and is quite competitive even though the concept increases in complexity.

Key words: Parsing, feature set extraction, C4.8, precision, recall, cross over, mutation

INTRODUCTION

Information extraction is a process which takes unseen texts as inputs and produces fixed-format, unambiguous data as output. IE application analyses texts and presents only the specific information from them that the user is interested in. IE systems are difficult and knowledge intensive to build and tied to particular domain and scenarios. Due to this, IE in Text mining is always limited to a particular corpora and the unstructured information from the texts can be converted into a structured format, which can be then queried using standard query languages or using natural language processing techniques.

Much of the work in information extraction deals with extracting information by training with a bag of words. The standard vector space model of text represents a document as a sparse vector that specifies a weighted frequency for each of the large number of distinct words or tokens that appear in a corpus. Some of the work deals with the usage of Natural language processing using a Hidden Markov Model (HMM) or a conditional Random field (CFR) based on the Veterbi algorithm (Raymond and Bunescu, 2005). Some of the IE systems treat the text as a sequence of un interpreted tokens, where the pattern matching technique or rule-based technique is adopted to retrieve the information.

Abutridy *et al.* (2004) has proposed a Genre-based IE for automatically extracting the fragments of key elements from text documents. In this study, we have used natural language processing combined with machine learning technique, in addition to the method specified by Abutridy *et al.* (2004) for automatically extracting key elements from the technical abstracts and to represent them in the form of rule-like structures, as they form the significant role in any document. It is always found that authors use only a specific set of features (verbs) for indicating these events. We have classified the features into various events as they occur based on their probabilities using Genetic algorithm, as they are considered as efficient in search and optimization. The performance of the Genetic algorithm as classifier is compared with C4.8, Rain Forest, Rain tree and NBTree method

SYSTEM ARCHITECTURE

The possible system architecture for our work is depicted in Fig. 1. Machine learning is the area of Artificial Intelligence that examines how to write programs that learn. With Machine learning the computer makes a prediction and then based on the feedback as to whether it is correct, learns from this feedback. It learns through examples, domain knowledge and feedback. When a

Corresponding Author: A. Christy, E2 Sai Vikas, 140 Velachery Road, Sembakkam, Chennai-73, India Tel: 044-32912963

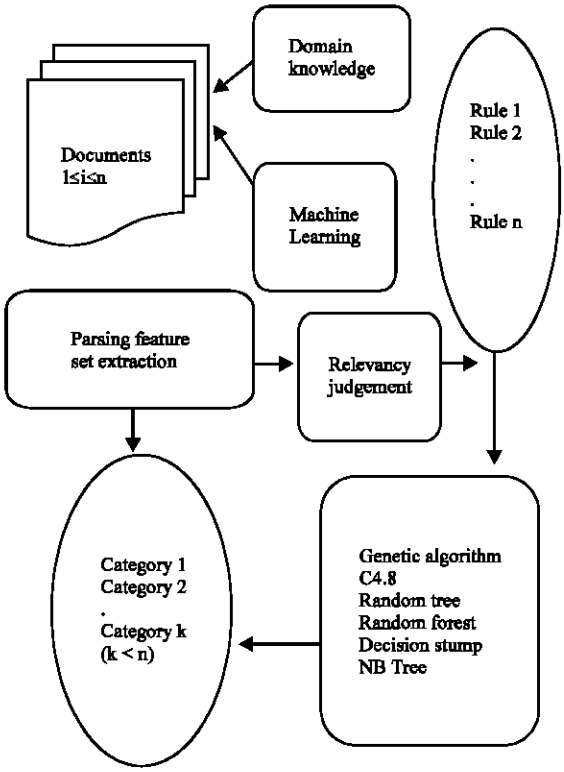

Fig. 1: Possible system architecture

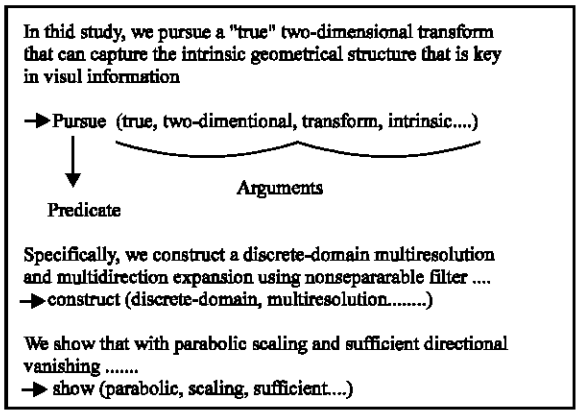

Fig. 2: Identification of predicates and arguments

similar situation arises in the future, this feedback is used to make the same prediction or to make a completely different prediction.

In order to retrive the key elements (features), we have parsed the sentence, removed the stop words and used the trigram model to identify the previous token, current token and next token to identify the verb. If the current token is a verb, it is considered as its predicate action and the subsequent features as its arguments and it is explained as shown in Fig. 2

If the current token is a verb and the next token is not a prepositional phrase, then the current token is identified as the feature to be extracted. The previous token considered as valid in our approach are Noun, Pro and Prep. The possible next tokens can be noun, pronoun, adjective, prep and det.

Initially the system is trained with a bag of features and further the features are extracted and the frequency of the features thus extracted are found using the tf.idfi function to retrieve the most frequently used feature. We have found the keywords which falls within the threshold value 0.301030-2.0000 has produced good recall. The features are checked if the values falls within the range and if not, it is proceeded by retrieving the next verb from the sentence.

CLASSIFICATION TECHNIQUES

In Data mining, Machine learning is often used for Prediction or Classification. Classification systems deal with methods for assigning a set of input objects to a set of decision classes and their accuracy depends on how well a set of features can present a problem. In Pattern recognition and Machine learning finding an acceptable set of features necessary for recognition and classification is a key part of designing efficient, implementable systems.

We have classified the features which we have extracted into various classes aim, methodology and conclusion i.e., present is often used as a keyword for specifying the aim and also in certain cases it is used for specifying the conclusion, whereas propose always specifies aim.

We have classified the features into the classes based on the probability of their occurrences distributed over the classes. We have used Genetic algorithm for classification. Compared with classical search-and-optimization algorithms, GAs are much less susceptible to getting struck in local suboptimal regions of the search space because they perform global searches by exploring multiple solutions in parallel (Abutridy *et al*., 2004). Given a population of potential problem solutions (individuals), evolutionary computing expands this population with new and potentially better solutions. We have used the C4.8, Random Tree, Random Forest, Decision stump and NBTree for classification.

GENETIC ALGORITHMS

John Holland introduced Genetic algorithms, which are adoptive search techniques named for the similarity between their operations and the mechanics of

population-Genetics models of natural systems (Vafari and De Jong, 1998). These techniques are used for finding the fittest models from a set of models to represent the data.

The best of these, as defined by a fitness function are then input to the next generation. Algorithms differ in how the models is represented, how different individuals in the models are combined and how the fitness function is used. A complete search of all possible individuals would yield the best individual or solution to the problem using the predefined fitness function. Since the research space is quite large (infinite), what a Genetic algorithm does is to prune from the search space individuals who will not solve the problem. In addition, it creates new individuals who probably will be much different from those previously examined. Since GA do not search the entire space, they may not yield the best result. However, they can provide approximate solutions to difficult problems

C4.8: The ID3 technique to building a decision tree is based on information theory and attempts to minimize the expected number of comparisons. The concept used to quantify information is called entropy. Entropy is used to measure the amount of uncertainty or surprise or randomness in a set of data. When all data in a set belong to a single class, there is no uncertainty. In this case, the entropy is zero. The C4.8 algorithm is the improved version of ID3 and it permits numeric attributes, deal sensibly with missing values and prune to deal with noisy data.

Random tree and random tree forest: A Random Tree Forest is an ensemble (collection) of decision trees whose predictions are combined to make the overall prediction for the forest. Decision tree forest models often have a degree of accuracy that cannot be obtained using a large, single-tree model. They use the out of bag data rows for validation of the model. This provides an independent test without requiring a separate data set or holding back rows from the tree construction. They can handle hundreds or thousands of potential predictor variables and the sophisticated and accurate method of surrogate splitters is used for handling missing predictor values. Surrogate splitters are predictor variables that are not as good at splitting a group as the primary splitter but which yield similar splitting results; they mimic the splits produced by the primary splitter. Surrogate splitters are used to classify rows that have missing values in the primary splitter. They function both when the tree is being built and later when the tree is used to score additional datasets

Number of trees in forest specifies how many trees are to be constructed in the decision tree forest. It is recommended that a minimum value of 100 be used. Minimum size node to split is a node in a tree in the forest will not be split if it has fewer than this number of rows in it. Maximum tree levels specify the maximum number of levels (depth) that each tree in the forest may be grown to. Some research indicates that it is best to grow very large trees, so the maximum levels should be set large and the minimum. The following are some of the paths created by the tree in our approach:

```
frequency < 1.8
    feature = present: aim (3/2)
    feature = propose: aim(1/0)
    feature = use: aim (2/1)
    feature = provide
        frequency < 0.78: :method(1/0)
    frequency < 0.78: method(1/0)
    frequency >= 0.78: aim (1/0) ……..
```

Decision stump: Stumps are the simplest special case of decision trees which consists of a single decision node and two prediction leaves (i.e., it is a decision Tree with only one split).

The decision stump learner works in time proportional to the number of training examples. It also requires memory that is proportional to the number of classes, number attributes and number of values. This can be very large for continuous attributes.

NBTree: NBTree is a hybrid of decision tree and Naive Bayes classifiers. The decision tree nodes contain univariate splits as regular decision trees and the leaves contain Naïve Bayes Classifiers.

Decision trees are easy to visualize while Naïve Bayes classifiers are easy to understand and the induction of these algorithms is extremely fast requiring only a single pass through the data if all attributes are discrete. A univariate (single attribute) split is chosen for the root of the tree using some criterion (e.g., gini index, gain ratio). The data is provided then according to the test and the process repeats recursively for each and every child. After a full tree is built, a pruning step is executed, which reduces the tree size.

Naïve-Bayes uses Baye's rule to compute the probability of each class given the instance, assuming the attributes are conditionally independent given the instance.

EXPERIMENTS

We have performed experiments on 200 abstracts collected from www.computer.org containing 2 data sets

Table 1a: Comparison of precision, recall and f-measure of data set 1

	Aim			Methodology			Conclusion		
Method adopted	Precision	Recall	f-measure	Precision	Recall	f-measure	Precision	Recall	f-measure
C4.8	0.311	0.609	0.412	0	0.000	0.000	0.300	0.261	0.279
Random tree	0.250	0.435	0.317	0	0.000	0.000	0.063	0.043	0.051
Random forest	0.273	0.391	0.321	0	0.000	0.000	0.292	0.304	0.298
Decision stump	0.500	0.043	0.080	0	0.000	0.000	0.365	1.000	0.535
NB tree	0.719	1.000	0.836	1	0.556	0.714	0.870	0.870	0.870

Table 1b: Comparison of precision, recall and f-measure of data set 1

	Aim			Methodology			Conclusion		
Method adopted	Precision	Recall	f-measure	Precision	Recall	f-measure	Precision	Recall	f-measure
C4.8	0.000	0.000	0.000	0.373	1.000	0.543	0.000	0.000	0.000
Random tree	0.700	1.000	0.824	0.750	0.682	0.714	1.000	0.563	0.714
Random forest	0.833	0.714	0.769	0.708	0.703	0.739	0.765	0.813	0.788
Decision stump	0.368	1.000	0.538	1.000	0.091	0.167	0.000	0.000	0.000
NB tree	0.833	0.714	0.769	0.656	0.955	0.778	1.000	0.563	0.720

Table 2a: Comparison of precision, recall and f-measure of data set 1 after 10-folds cross validation

	Aim			Methodology		
Method adopted	Precision	Recall	f-measure	Precision	Recall	f-measure
C4.8	0.333	0.190	0.242	0.340	0.727	0.464
Random tree	0.263	0.238	0.250	0.207	0.273	0.235
Random forest	0.211	0.190	0.200	0.182	0.273	0.218
Decision stump	0.354	0.810	0.493	0.300	0.136	0.187
NB tree	0.235	0.190	0.211	0.200	0.318	0.246

Table 2b: comparison of precision, recall and f-measure of data set 2 after 10-folds cross validation

	Aim			Conclusion		
Method adopted	Precision	Recall	f-measure	Precision	Recall	f-measure
C4.8	0.311	0.609	0.412	0.300	0.261	0.279
Random tree	0.250	0.435	0.317	0.063	0.043	0.051
Random forest	0.273	0.391	0.321	0.292	0.304	0.298
Decision stump	0.269	0.304	0.286	0.333	0.522	0.407
NB tree	0.289	0.478	0.361	0.238	0.217	0.227

related to information retrieval and image processing producing the documents in the form of rule-like structures as shown in Fig. 3a. Since we select the features based on the threshold value, our system scores 84% accuracy in retrieving the rule-like structures, which is an improvement over the traditional approaches bag of words, natural language processing and pattern matching. We selected the set that produced the best result on average-the lowest error rate-and the keyword generated during its best run become input for the classification system.

From the features we have extracted from the text documents, the first two-thirds of the features we have chosen as the training data set and the remaining as the test data set. We have trained the dataset to build up the decision trees for the evaluation procedure and the comparison with other methods are shown in the Table 1a and b results after applying 10 folds cross validation is shown in Table 2a and b. The results after cross validation shows C4.8 having the highest f-measure.

For Genetic algorithm technique, by choosing a crossover probability 0.99, 0.8, 0.7, 0.6 and mutation 0.01 we have found that cross over point 0.99 has performed good results and the accuracy for the first training data set is 0.566038 and the accuracy in Test data is 0.316072

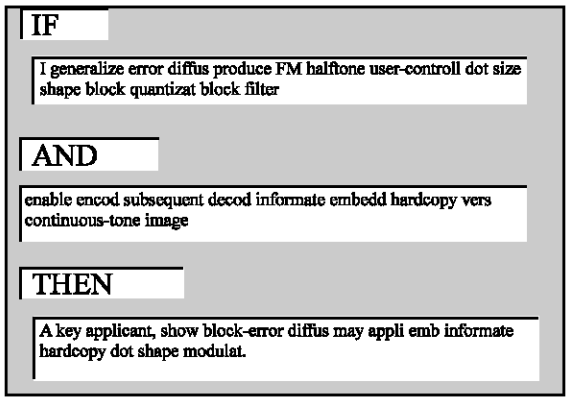

Fig. 3a: Rule extracted from a text documents

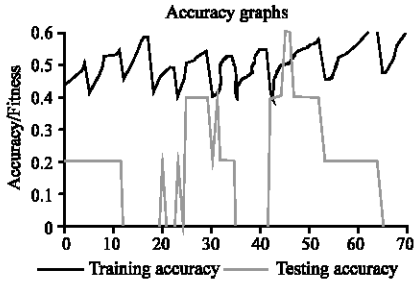

Fig. 3b: Accuracy of training data with testing data set 1

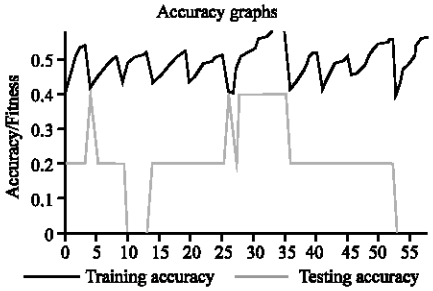

Fig. 3c: Accuracy of training data with testing data set 1

the best size of tree as 37 and average genome score as 0.27640, which is close to the value produced by C4.8 (Fig. 3b).

In the second data set, the accuracy in training data is 0.574074 and the accuracy in Test data is 1.319117 with the best size of tree as 25 and average genome score as 0.303032 and we have found that the average genome score is close to the value produced by C4.8 algorithm and the accuracy in Genetic algorithm is better than the results produced by C4.8 (Fig. 3c).

CONCLUSIONS

We have shown the extracting information from text documents using Machine learning technique based on a limited threshold can improve the performance of the system. Also, we have shown the classification of features extracted using Genetic algorithm has produced the accuracy close to C4.8 algorithm. As information extraction is limited to a particular corpora, the usage is limited. Also in extracting the aim, methodology and conclusion using the features, the features used for identifying aim and conclusion are same in different domain whereas the feature for methodology varies in different domain. This work can be further implemented by considering different interrelated topics.

REFERENCES

John Abutridy, C. Mellish and S. Aitken, 2004. Combining Information extraction with Genetic algorithm for text mining. IEEE Intelligent Syst., 19: 22-30.

Raymond, J., M. Razvan and Bunescu, 2005. Mining knowledge from text using information extraction. ACM. SIGKDD Explorations, 7: 3-10.

Vafarie, N.K. and De Jong, 1998. Feature space transformation using Genetic algorithms, IEEE Intelligent Sys., 13: 57-65.