Automatic categorization of questions from Q&A sites

Eduardo Cunha Campos
Computer Science Department
Federal University of Uberlândia
Uberlândia, Brazil
eduardocampos@mestrado.ufu.br

Marcelo de Almeida Maia
Computer Science Department
Federal University of Uberlândia
Uberlândia, Brazil
marcmaia@facom.ufu.br

ABSTRACT

Q&A sites are attracting growing interest of software developers. The categorization of questions in terms of user concerns would open new opportunities to extract valuable information from millions of posts.

This paper presents a comparison between different classification algorithms to find the one that best classifies questions from Q&A sites, such as, Stack Overflow. In the classification process, we used the following classification algorithms: Naive Bayes, Multilayer Perceptron, Support Vector Machine, K-Nearest Neighbors, J4.8 Decision Tree and Random Forests.

We conducted an experimental study with Stack Overflow questions with posts equally divided into three domain categories: *How-to-do-it*, *Need-to-know* and *Seeking-something*. The attributes were extracted from a textual analysis of the title and body of each question. We considered a total of 8 attributes to get the data for each question. We found a classifier with an overall success rate of 84.16% and 92.5% on *How-to-do-it* category.

Categories and Subject Descriptors

I.5.4 [**Pattern Recognition**]: Applications – *Text processing*

General Terms

Pattern Recognition, Data Mining

Keywords

Q&A sites, Classification algorithms, Bayes

1. INTRODUCTION

Developers often deal with various technologies and need to exchange knowledge with each other to find answers to their problems. A useful source of information is Q&A sites. Stack Overflow has been widely used by developers to post their programming questions and receive answers for them.

SAC'14 March 24-28, 2014, Gyeongju, Korea.

The automatic categorization of the questions according to the concerns of the questioner may have many applications:

- Interesting *How-to-do-it* questions could be selected to provide a cookbook on desired topic;
- Some *Need-to-know* conceptual questions could be selected to compose a plugin for an Integrated Development Environment (IDE) on best practices for different topics;
- Some *Seeking-something* questions could be selected to generate the state of the art about tools, books and tutorials on a desired topic;

However, the manual categorization task of the questions is tedious and labour-intensive.

The remainder of this paper is organized as follows. Section 2 shows the experimental setting used to conduct this study. Section 3 presents and discusses the results. Section 4 concludes the paper.

2. EXPERIMENTAL SETTING

In this section, we state our research goal and define the criteria used to build the dataset. Furthermore, we give a brief description of each classification algorithm considered in this study and present how the attributes were defined. Finally, we expose our evaluation to estimate the success rate of the classifiers.

2.1 Goal

The objective of this work is to investigate the success rate of different classification algorithms to find the one that best classifies the set of questions from Stack Overflow.

2.2 Classification Algorithms

In the classification process, we considered six classification algorithms widely used in the data mining area:

- **Naive Bayes (NB)**: These classifiers assume that all the attributes are independent and that each contributes equally to the categorization. A category is assigned to a project by combining the contribution of each feature [5].
- **Multilayer Perceptrons (MLPs)**: They are an important class of neural networks. Typically, the network consists of a set of sensory units that constitute the *input layer*, one or more *hidden layers* of computation nodes, and an *output layer* of computation nodes [3].

- **Support Vector Machines (SVMs)**: These classifiers split the problem space into two possible sets by finding a hyper-plane that maximizes the distance with the closest item of each subset [5].
- ***K*-Nearest Neighbors (KNN)**: This algorithm is a lazy classifier because it does not induce a categorization model from training data. The category for the new instance is selected from the categories of the K most similar instances [5].
- **J4.8 Decision Tree (J4.8)**: J4.8 is an algorithm for construction of a decision tree that allows the manipulation of both discrete and continuous attributes [8].
- **Random Forests (RFs)**: They are trained in a supervised way. At run-time, a test sample is passed down all the trees of the forest, and the output is computed by averaging the distributions recorded at the reached leaf nodes [4].

2.3 Attribute definition

We define 8 attributes to characterize the questions. Each attribute refers to the number of times that a keywords set appear in the title and body of the question. Table 1 represents the relationship between the attributes and their respective keywords.

Table 1: Attributes and their respective keywords

Attribute	Keywords
howQtd	"how"
debugQtd	"exception(s)", "error(s)", "debug", "debugging", "fail", "failed", "warning", "notice", "fault", "problem", "matter", "wrong", "incorrect", "notification", "trouble", "denied", "breakpoint", "unhandled"
whatQtd	"what", "why", "which", "meaning", "significance", "difference(s) between", "how much", "how many", "how difficult"
isThereQtd	"is there", "are there", "exist(s)"
possibleQtd	"wonder", "should", "possible", "feasible", "uncertain"
lookingQtd	"looking for", "looking forward", "searching for", "seeking", "tool(s)", "book(s)", "tutorial(s)", "resource(s)", "where", "looking at", "searching forward", "searching at"
adviceQtd	"when", "advice", "recommendation", "recommend", "guideline", "guide", "suggestion", "suggest", "opinion", "ideas"
optimalQtd	"optimal", "efficient", "best", "better", "reliable", "elegant", "appropriate", "safest", "security", "fast", "quickly", "suitable", "robust", "performant", "reasonably", "viable", "practicable", "smoother", "lightweight"

2.4 Stack Overflow Dataset

We downloaded a release of Stack Overflow's public data dump and import the data into a relational database in order to classify the Stack Overflow questions.

We selected from the constructed relational database, a batch of 100 questions and manually classified them. We repeated this process until we get 40 questions in each of the three categories:

- *How-to-do-it*: Providing a scenario and asking about how to implement it [6];
- *Need-to-know*: Questions regarding possibility or availability of something. These questions normally show the lack of knowledge or uncertainty about some aspects of the technology [6];
- *Seeking-something*: The questioner is looking for something (e.g. book, tutorial, tool), searching for a quality solution (e.g. reliable, efficient) or need a recommendation (e.g. an advice, an opinion).

In the next step, we generated the ARFF(Attribute-Relation File Format) file, containing the attribute information for each classified question.

2.5 Evaluation

2.5.1 Cross-validation

We chose to use the cross-validation method because it is suitable to compare the performance of two or more different algorithms and find out the best algorithm for the available data. Moreover, it is indicated when the amount of labeled data is relatively small. In all the tests conducted in this study, we used 10-fold cross-validation [7].

2.5.2 Feature Selection

The attribute reduction is one of the key processes for knowledge acquisition. Some attributes may be irrelevant or redundant to the mining task and they can causing confusing for mining algorithm employed [1].

Information Gain and Chi-square are among the most effective methods of Feature Selection for classification [1]. In this study, we used the Weka [2] software to perform Information Gain Filter in the attributes.

Table 2 shows the most relevant attributes ordered by Information Gain Value. The higher the value of Information Gain, the better the attribute classifies the sample. The remaining original attributes are least relevant to classify the sample. The Information Gain Value for them was zero.

Table 2: Ranked Attributes: Information Gain

Information Gain Value	Attribute
0.583	howQtd
0.161	optimalQtd
0.117	isThereQtd
0.101	lookingQtd

3. RESULTS AND DISCUSSION

Figure 1 shows a bar chart comparing two different test scenarios. "Test 1" represents the scenario considering all attributes defined in this work. "Test 2" represents only the attributes that were selected with Feature Selection. We can observe that some classification algorithms increased their success rate when it was done the "Test 2": NB, SVM, KNN and J4.8.

Table 3 shows the confusion matrix of NB Classifier in the "Test 2". The main diagonal of the matrix represents

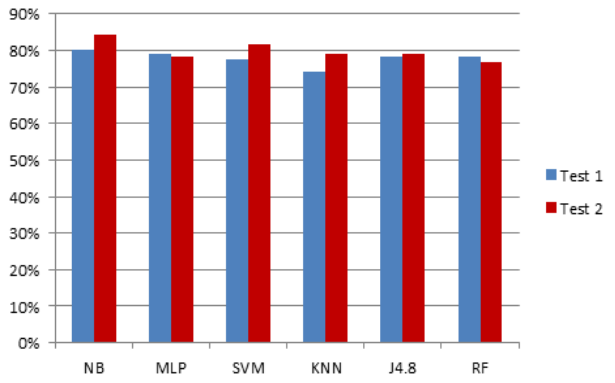

Figure 1: A bar chart comparing "Test 1" (8 attributes) and "Test 2" (4 attributes)

the correctly classified instances by classifier. We can state that the NB classifier has greater difficulty in differentiating between questions of *Need-to-know* category and *Seeking-something* category.

Table 3: NB Confusion Matrix (4 attributes)

a	b	c	<– classified as
37	2	1	a = How-to-do-it
3	32	5	b = Need-to-know
3	5	32	c = Seeking-something

In the "Test 2", the NB classifier achieved a success rate of 92.5% on questions of *How-to-do-it* category and a success rate of 80% on questions of *Need-to-know* and *Seeking-something* categories.

Table 4 and Table 5 show the classification results obtained with the Weka [2] software and have the same column structure: Classifier, Success rate (%), Correctly Classified Instances (Correct) and Incorrectly Classified Instances (Incorrect). Table 4 represents the first test scenario ("Test 1"), whereas Table 5 represents the second ("Test 2").

Table 4: 8 attributes - All attributes

Classifier	Success rate (%)	Correct	Incorrect
NB	80	96	24
MLP	79.1667	95	25
SVM	77.5	93	27
KNN	74.1667	89	31
J4.8	78.3333	94	26
RF	78.3333	94	26

Table 5: 4 most relevant attributes

Classifier	Success rate (%)	Correct	Incorrect
NB	84.1667	101	19
MLP	78.3333	94	26
SVM	81.6667	98	22
KNN	79.1667	95	25
J4.8	79.1667	95	25
RF	76.6667	92	28

We observed that the NB classifier increased their accuracy in "Test 2". This suggests the existence of bad or correlated attributes that were confusing the classifier in "Test 1". The higher success rate was obtained with NB classifier (84.1667%). Table 5 shows the results obtained in "Test 2".

4. CONCLUSIONS

We carried out an experiment using 120 Stack Overflow questions, equally divided into three domain categories: *How-to-do-it*, *Need-to-know* and *Seeking-something*. All questions considered in this study were selected and classified manually by us. We defined 8 attributes to perform textual analysis of the title and body of each question.

As our sample is relatively small, we used Cross-validation (10 folds) to avoid overfitting and increase the accuracy of the success rate of the classifiers. The Information Gain Filter revealed that the most important attributes for this problem are: "howQtd", "optimalQtd", "isThereQtd" and "lookingQtd". We can state that some classifiers increased their success rate when the Feature Selection was done: NB, SVM, KNN and J4.8.

The results showed that the higher success rate was obtained with NB classifier (84.1667%). Furthermore, this classifier achieved a success rate of 92.5% on questions of *How-to-do-it* category. As future work, we can still continue to improve the classification accuracy, and we will investigate how the categories can be used in new applications.

ACKNOWLEDGMENTS

This work was partially supported by FAPEMIG grant CEXAPQ-2086-11 and CNPQ grant 475519/2012-4.

5. REFERENCES

[1] G. Forman. An extensive empirical study of feature selection metrics for text classification. *Journal of Machine Learning Research*, 2003.

[2] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The weka data mining software : An update. *SIGKDD Explorations*, pages 10–18, 2009.

[3] S. Haykin. *Neural Networks: A Comprehensive Foundation*. Prentice Hall, PTR Upper Saddle River, NJ, USA, 1998.

[4] V. Lempitsky, M. Verhoek, A. Noble, and A. Blake. Random forest classification for automatic delineation of myocardium in real-time 3d echocardiography. pages 449–449, 2009. Springer Verlag.

[5] M. Linares-Vasquez, C. McMillan, D. Poshyvanyk, and M. Grechanik. On using machine learning to automatically classify software applications into domain categories. In *Empirical Software Engineering*, pages 7–8, 2009. Springer US.

[6] S. Nasehi, J. Sillito, F. Maurer, and C. Burns. What makes a good code example?: A study of programming q&a in stackoverflow. In *ICSM*, pages 25–34. IEEE Computer Society, 2012.

[7] C. Park and D. Kim. Cross-validation. In *Progress in neurological surgery*, pages 1–12, 2012.

[8] L. Sehgal, N. Mohan, and P. S. Sandhu. Quality prediction of function based software using decision tree approach. In *ICCEMT*, pages 43–44, 2012.