AN INDUCTIVE RULE LEARNING TECHNIQUE FOR TEXT MINING IN QUESTIONNAIRES

Stephanie Chua[1] and Frans Coenen[2]

[1]UniversitiMalaysia Sarawak (UNIMAS), Malaysia, chlstephanie@fit.unimas.my
[2]University of Liverpool, United Kingdom, coenen@liverpool.ac.uk

ABSTRACT. This paper describes an inductive rule learning (IRL) technique for classifying questionnaires based on the natural language responses to the open-ended questions frequently found in questionnaire data. These responses are deemed to provide important information to the purpose of the questionnaire. Given that the responses are in the form of unstructured natural language text and that a collection of questionnaires can comprise thousands of returns, an automated approach for handling such text is desirable for analysis purposes. One common analysis task is the classification of questionnaires. For this purpose, an IRL technique is presented. An empirical comparison is also conducted to compare the presented technique with other established machine learning techniques. This IRL technique has been shown to be effective and efficient when applied to the classification of a collection of veterinary questionnaires.

Keywords: Machine Learning, Inductive Rule Learning, Classification

INTRODUCTION

A questionnaire is a form of document consisting of questions used to gather data from respondents, whether for the purpose of a survey or for research. Questions in a questionnaire can come in different formats but are largely divided into two categories: closed-ended questions and open-ended questions. In closed-ended questions, the respondents are presented with a set of options to select from and their answers are limited to the provided options. Open-ended questions, on the other hand, require the respondents to provide their own answers, usually in a natural language format. Answers to open-ended questions are usually deemed to provide vital feedbacks to the purpose of the questionnaire. Given that a collection of questionnaire can comprise large numbers, it is useful to be able to analyze the natural language element automatically. One common analysis taksis to classify the questionnaires into different classes based on the feedback gathered from the respondents. This can be to classify the feedback as “Good” or “Not Good” if the questionnaire is about a product or service, or to classify the feedbacks according to some pre-defined set of labels if the questionnaire is focused on a specific domain. Many techniques can be employed for the classification of questionnaires. One popular technique is machine learning, where the objective is to learn a model (classifier) from a set of pre-labeled documents (in this case, questionnaires) so that the learnt model can be used to classify previously unseen cases. Thus, the aim of this paper is to use a popular machine learning technique, Inductive Rule Learning (IRL), to learn a set of rules to be used as a classifier to classify a collection of questionnaire according to the nature of the natural language text responses to open-ended questions.

IRL is a generic term describing a machine learning technique used to derive rules from data. The advantage offered by IRL, over many other forms of machine learning (such as

Support Vector Machines (SVM), Neural Networks (NN) and probabilistic Naïve Bayes (NB)) is that the Disjunctive Normal Form (DNF) rules produced are expressive while at the same time being easily interpretable by humans. This advantage is of importance, particularly in applications where manual human intervention is essential in analyzing the classifiers, for example in the medical domain. Thus, the rules can be read, verified and modified so that they can be used for further studies. In the context of classification, the derived rules are typically of the form IF *condition* THEN *conclusion*; where the *condition* (also known as the antecedent) consists of a conjunction of features, while the *conclusion* (also known as the consequent) is the resulting class label associated with the condition. For example:

$$\text{IF } a \text{ AND } b \text{ THEN } x \ (a \wedge b \Rightarrow x)$$

where *a* and *b* are features that appear in a document, and *x* is a class label. This is simply interpreted as, if *a* and *b* occur together in a document, then classify the document as class *x*.

Rules do not normally include the negation of features. For example:

$$\text{IF } a \text{ AND } b \text{ AND NOT } c \text{ THEN } x \ (a \wedge b \wedge \neg c \Rightarrow x)$$

which is interpreted as, if *a* and *b* occur together in a document and *c* does not occur, then classify the document as class *x*. Intuitively, the inclusion of negation in rules should provide the rules with an additional capability to distinguish between classes. Therefore, this paper presents an IRL technique which is able to generate both rules with and without negation.

The rest of this paper is organized as follows. A brief review of related work is presented next. Then, the proposed IRL technique is presented. This is followed by a brief description of the dataset used. The experimental setup and evaluation is discussed next. Finally, a conclusion is given.

RELATED WORK

The IRL technique (without the inclusion of negation in rules)used in classification is a mature research field. Much previous research into IRL had been applied to the classification task. Perhaps the most popular of the IRL algorithms is the Repeated Incremental Pruning to Produce Error Reduction (RIPPER) (Cohen, 1995) algorithm. RIPPER is an IRL system which uses a “covering” algorithm to learn rules. Rules are generated by greedily adding features to a rule until the rule achieves 100% accuracy. This process tries every possible value of each feature and chooses the one with the highest information gain. Following this rule building phase, a rule pruning phase is applied whereby the generated rule is pruned using a pruning metric. Another rule induction method, Swap-1 (Weiss & Indurkhya, 1993), was used by Apté et al. (Apté et al., 1994) to induce rules for text classification. Weakest-link pruning was used to progressively prune the rule set so as to obtain a sequence of rule sets in a decreasing order of complexity. The best rule set was the one that resulted in the lowest observed true error rate in a test dataset.

IRL techniques which explicitly include a mechanism to generate rules with negation have also been seen in the literature. These include Olex Greedy (Rullo et al., 2007) and OlexGA (Pietramala et al., 2008).The Olex system was founded on the idea of using a fixed template that allows only one positive feature and zero or more negated features to generate rules. OlexGreedy used a greedy single stage rule learning process. OlexGreedy has the disadvantage of not being able to express co-occurrences based on feature dependencies. This is due to the adopted template approach that permits only one positive feature in the rule antecedent. This disadvantage was overcome by using conjunctions of terms (coterms), where conjunctions of positive features could be included in the rules generated. However, it was again reported that the rules generated using the improved version could not share common features in the rule antecedent. This meant that rules having the same positive feature in the rule antecedent could not be generated by OlexGreedy. Hence, OlexGA was proposed, which made used of a genetic algorithm to generate rules. However, the rules generated still used the same template of one positive feature (or coterm) and zero or more negated feature(s) (or coterms). This template is somewhat restrictive in that rules with flexible combinations of positive and negated features could not be generated. Despite this disadvantage, both the Olex systems performed better than other techniques such as C4.5,SVM, RIPPER and NB.

INDUCTIVE RULE LEARNING (IRL)

The proposed IRL technique aims to improve the effectiveness of classifiers, by using both positive and negated features, while maintaining the simplicity and effectiveness of the covering algorithm. In the covering algorithm, rules are learned sequentially one at a time based on the training dataset. The novel element of this proposed technique lies in the rule refinement component. Given a rule $a \Rightarrow x$; the rule may cover both positive and negative documents. Positive documents are documents in the training set that are correctly classified, while negative documents are those that are incorrectly classified. The rule refinement component is used to obtain a more specialized rule $a \wedge b \Rightarrow x$. Over fitting can be prevented by using stopping conditions for rule refinement, consisting of: (i) when a rule no longer covers negative documents, (ii) when the feature search space is empty or (iii) when the previous rule learnt has a higher or equal accuracy to that of the current rule learnt. Learning a rule without negation is straightforward; a conjunction of features that occur together in positive documents are used in the rule antecedent. However, to generate rules with negation requires the identification of features that can be advantageously negated. The proposed technique uses feature space division to divide the feature space into three sub-spaces comprising:

1. Unique positive (UP) features: features that appear only in positive documents.
2. Unique negative (UN) features: features that appear only in negative documents.
3. Overlap (Ov) features: features found in both positive and negative documents.

When a rule is refined, a feature from either one of the three sub-spaces can be selected to be added to the rule. Adding a UP or Ov feature will result in the generation of a rule with no negation. If a UN feature is added, then the feature is negated in the rule and hence, a rule with negation is generated. When adding a UP or UN feature, the feature with the highest document frequency is selected. When adding an Ov feature, the feature with the highest document frequency difference (i.e. positive document frequency minus negative document frequency) is selected. This will ensure that the refined rule covers the maximum possible number of positive documents.

Feature space division allows for the effective and efficient identification of positive or negated features to be used when refining rules. Eight rule refinement strategies were also devised to be used with these three sub-spaces. These strategies are able to generate rules both with and without negation. The UP, UN and Ov strategies refine a rule by adding a feature from the sub-space from which they are named. Given that a sub-space may be empty, the UP, UN and Ov strategies may lead to rule refinement being prematurely halted in the absence of any features to be added to a rule. Hence, two further strategies have been devised to address the empty sub-space problem: UP-UN-Ov and UN-UP-Ov. These strategies use a sequence of sub-space combinations and are labeled in the order that the sub-spaces are considered. In the UP-UN-Ov strategy, the UP features are considered first; if the UP sub-space is empty, then the UN features will be considered instead, and then the Ov features if the UN sub-space is also empty. The UN-UP-Ov strategy works in a similar manner, only interchanging the order of UP and UN. The Ov features are used as the last option because using an Ov feature will result in the coverage of at least one negative document. These five strategies described may generate different refined rules. The sixth strategy named as BestStrategy, therefore, chooses the best rule (using rule accuracy with Laplace estimation) from the rules generated from the first five strategies. Each of the first five strategies use a depth-first search. The final two strategies provide for a more exhaustive search through the possible rules generated. In the BestPosRule strategy, two versions of the original rule are generated, one by refining the original rule with a UP feature and another by a Ov feature. Each of these rules is further refined in the same manner until a stopping condition is met. The rule with the highest Laplace accuracy is selected to be added to the rule set. This strategy uses two sub-spaces for rule refinement and will generate rules without negation. The BestRule strategy is an extension of the BestPosRule strategy, where a third version of the original rule is generated by refining the original rule with a UN feature. The rule with the highest Laplace accuracy is added to the rule set. This strategy uses all three sub-spaces for rule refinement and may generate rules with negation. A summary of these strategies is given in Table 1.

Table 1. Summary of the Proposed Rule Refinement Strategies

Strategy	Description	Sample Rules
UP	Add a UP feature to refine a rule	$a \wedge b \Rightarrow x$
UN	Add a UN feature to refine a rule	$a \wedge \neg c \Rightarrow x$
Ov	Add an Ov feature to refine a rule	$a \wedge b \wedge d \Rightarrow x$
UP-UN-Ov	If UP is not empty, add a UP feature to refine a rule; Else If UN is not empty, add a UN feature to refine a rule; Else If Ov is not empty, add an Ov feature to refine a rule	$a \wedge b \Rightarrow x$
UN-UP-Ov	If UN is not empty, add a UN feature to refine a rule; Else If UP is not empty, add a UP feature to refine a rule; Else If Ov is not empty, add an Ov feature to refine a rule	$a \wedge b \wedge \neg c \Rightarrow x$
BestStrategy	Choose the best rule from the five rules generated by each UP, UN, Ov, UP-UN-Ov and UN-UP-Ov	$a \wedge b \wedge d \Rightarrow x$
BestPosRule	Generate two versions of the original rule; one refined with a UP feature and the other refined with an Ov feature. Choose the best rule between the two versions.	$a \wedge b \wedge d \wedge e \Rightarrow x$
BestRule	Generate three versions of the original rule; one refined with a UP feature, one refined with a UN feature and the other refined with an Ov feature. Choose the best rule between the three versions.	$a \wedge b \wedge \neg c \wedge \neg \mathrm{f} \Rightarrow x$

DATASET

The dataset used for evaluation purposes comprises a collection of questionnaires from the veterinary domain called the Small Animals Veterinary Surveillance Network (SAVSNET) dataset collected as part of the SAVSNET project (Radford et al., 2010). This project aims to determine the disease status of small animals (mostly cats and dogs). Each questionnaire is the result of a single veterinary consultation. The questionnaires have both closed (tabular) and open (free text) ended questions. We are interested in the free text data. These are notes made by a veterinary practitioner which include diagnostics, treatments and prescriptions. The questionnaire returns have been hand-annotated by domain experts. To construct a dataset useful for the purpose of this research, four annotations (class labels) were selected: aggression, diarrhea, pruritus and vomiting. The project had 27,072 questionnaire returns. However, 26,039 of them did not include the four class labels that were considered, leaving 1,033 questionnaires. Out of these, 89 questionnaires were found to be repetitions and thus, were removed. The remaining 944 questionnaires all had one of the four class labels. However, 116 of these had their open-ended questions left blank. This left us with only 828 questionnaires with free text data. The distribution of the class labels across these 828 questionnaires is as follows: Aggression (34), Diarrhea (308), Pruritus (350) and Vomiting (136).

EXPERIMENTAL SETUP AND EVALUATION

A generic framework for text classification was adopted for classifying the test collection of questionnaires. The framework included a number of processes: document preprocessing, text representation, IRL and classification. To evaluate a classifier the input dataset is usually split into a training set and a test set. The first step is to preprocess the training set. This will include sub-processes such as stop words removal and feature selection. In this research, the Chi-Square statistics (Sebastiani, 2002) is used as the feature selection technique. The preprocessed training set is then translated into a suitable representation. The text representation format used here is the bag-of-words (Radovanović & Ivanović, 2006) representation. Next, the IRL process as described earlier, is applied to the representation of the training set. A classifier is learnt from here and then applied to the preprocessed test set to evaluate its performance. The experiments that were conducted compared the use of the proposed IRL technique with that of JRip (an implementation of RIPPER), NaiveBayes (NB) (a probabilistic naive bayes algorithm) and Sequential Minimal Optimization (SMO) (an SVM algorithm) from the Waikato Environment for Knowledge Analysis (WEKA) machine learning workbench (Witten et al, 2011). In addition, an OlexGreedy and OlexGA plug-in to WEKA were also compared. The results obtained (micro-averaged F_1-measure for classification effectiveness and time in seconds taken to generate a classifier model) are shown in Table 2.

Table 2. Experimental Results (best two results are in bold)

Techniques	Micro-averaged F_1-measure	Time (seconds)
IRL + UP	0.794	0.29
IRL + UN	**0.832**	**0.24**
IRL + Ov	0.823	0.27
IRL + UP-UN-Ov	0.794	0.30
IRL + UN-UP-Ov	0.826	**0.21**
IRL + BestStrategy	0.822	0.57
IRL + BestPosRule	0.829	0.30
IRL + BestRule	0.827	0.96
JRip	0.796	16.64
OlexGreedy	0.765	9.23
OlexGA	0.820	95.78
NB	0.829	2.36
SMO	**0.852**	4.83

Comparing only between the different strategies used in the IRL technique, the best result was obtained when the UN strategy was used for rule refinement. In comparison with JRip, OlexGreedy, OlexGA, NB and SMO, IRL + UN was reasonably effective, coming second after SMO. As an SVM-based technique, SMO is expected to perform well, as SVMs are known to be one of the best techniques for classification. However, despite its good performance, it does not have the advantage of being easily interpretable by humans, an advantage that makes rule-based techniques desirable. When comparing only rule-based techniques (not including SMO and NB), the IRL + UN technique recorded the best results. The worst technique in this case was OlexGreedy. In addition to being an effective technique for classifying the collection of questionnaires, the IRL techniques using the rule refinement strategies took less than a second each to generate a classifier model. Among the other machine learning techniques, OlexGA took the longest time to generate a classifier model while NB was the fastest, although none were faster than the IRL technique.

CONCLUSION

This research paper has presented an IRL technique used for classification of questionnaires using the natural language text found in answers to open-ended questions. The IRL technique, using a number of rule refinement strategies, was found to be competitive with an SVM-based technique, deemed to be one of the best for text classification. Its performance against the other established machine learning techniques has shown that it is effective and efficient with respect to the SAVSNET dataset used for evaluation purposes.

ACKNOWLEDGMENTS

The corresponding author would like to thank Universiti Malaysia Sarawak for providing the funds to attend the conference in which this research paper is published.

REFERENCES

Apté, C., Damerau, F. J. and Weiss, S. M. (1994). Automated learning of decision rules for text categorization. *ACM Transactions on Information Systems*, 12(3), 233-251.

Cohen, W. (1995). Fast effective rule induction. *Proceedings of the 12th InternationalConference on Machine Learning (ICML)*, 115-123.

Pietramala, A., Policicchio, V. L., Rullo, P. and Sidhu, I. (2008). A genetic algorithm for textclassification rule induction. In Daelemans, W., Goethals, B. and Morik, K.(Eds.), *Lecture Notes in Computer Science: Vol.* 5212.*Machine Learning and Knowledge Discovery in Databases* (pp.188-203). Springer Berlin/Heidelberg.

Radford, A., Tierney, A., Coyne, K. P., Gaskell, R. M., Noble, P. J., Dawson, S., Setzkorn, C., Jones, P. H., Buchan, I. E., Newton, J. R. and Bryan, J. G. (2010). Developinga network for small animal disease surveillance.*Veterinary Record*, 167, 472-474.

Radovanović, M. and Ivanović, M. (2006). Document representations for classification of short web-page descriptions. *Proceedings of the 8th International Conference on Data Warehousing and Knowledge Discovery (DaWaK), Lecture Notes in Computer Science: Vol. 4081* (pp.544-553). Springer-Verlag.

Rullo, P., Cumbo, C. and Policicchio, V. L. (2007). Learning rules with negation fortext categorization.*Proceedings of the 2007 ACM Symposium on AppliedComputing*, 409-416.

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1), 1-47.

Weiss, S. M. and Indurkhya, N. (1993). Optimized rule induction. *IEEE Expert: Intelligent Systems and Their Applications*, 8(6), 61-69.

Witten, I. H., Frank, E. and Hall, M. A. (2011). *Data Mining: Practical Machine Learning Tools and Techniques*, Morgan Kaufmann.