Text Classification Using R
What makes a good post on StackOverflow?

Julian Aviles

1. Introduction

Text classification, also known as text tagging, is the process of categorizing text into predefined groups. This area of text mining is significant in many real-world applications. Some tasks, sucqh as manually labeling thousands of documents, are labor intensive and require specialized training. Using text classification techniques, we can eliminate some of these costs, while at the same time often reducing the error rate. Machine learning algorithms are often employed in text classificatin techniques due to their superior performance.

Prior research has been done in this area, for example that done by Bao and Ishii, which ensembles multiple models using k nearest neighbors, a relatively simple technique in machine learning [1]. An interesting point raised by the paper is that emsembling kNN with other commom machine learning algorithms tended to degrade performace. Another interesting application is that done by Leopold and Kindermann, in which they use Support Vector Machine technique as the model to use in classification [2]. Deep learning is a technique that has often shown superior performance in a variety of domains. It has been studied in application to text mining by use of ensembling, as Sung-Bae and Jee-Haeng did, with excellent results in accuracy [3].

2. Method

This project will focus on using machine learning techniques to classify user-submitted post on the website StackOverflow.com. The data set contains 60,000 observations of posts on StackOverflow, which are classified into three different types of posts, each with 20,000 observations. One group is high quality posts with a total of 30+ score and without a single edit. Next is low quality posts with a negative score, and multiple community edits, but still remaining open after those changes. Last we have low quality posts that were closed by the community without a single edit. The data set has the advantage of having perfectly balances classes, which will help us focus on choosing a model with good fit, instead of handling class imbalance issues.

The data set will be split into a training and validation set (80% training and 20% validation), so that we can train the model and check for over or under fitting. A variety of algorithms will be tried and ensembling will be used to try to improve performance. Ensembling will be done taking into account model type, for example kNN tends to degrade in performance when ensembled with other models, as mentioned above.

3. Data Analysis

To analyze the data, we will start by cleaning the text, which is in HTML format. We will remove elements from the HTML language such as tags, standardize the text by lowercasing and other procedures, tokenize, and stem. Next we will probably need to vectorize the text so that it can be fed into a machine learning model.Feature selection will need to be considered. Research has been done by Soucy and Mineau regarding feature selection by means of a scoring function to rank features and filter unimportant ones [4]. The paper presents five of these scoring functions, which may be incorporated in the cleaning of the data.

As for the models to be used, the current candidates are Naive Bayes, kNN, support vector machine, and deep learning models. I want to try applying each of these methods separately to gauge performance, then perhaps ensemble to create a model with superior accuracy. Naive Bayes classifiers are one of the

most common models in text classification, which is why it is being included in the potential models. K nearest neighbors is being considered because of the simplicity of the model structure. Deep learning has had state-of-the-art performance in many domains, which also makes it a good candidate. Support vector machines often have good performance in classification tasks so we will include it as well. Currently, I do not know how the model will look, but ultimately I think we will have model with good accuracy.

References

[1] Bao Y. and Ishii N., "Combining Multiple kNN Classifiers for Text Categorization by Reducts", LNCS 2534, 2002, pp. 340-347.

[2] Leopold, Edda & Kindermann, Jörg, "Text Categorization with Support Vector Machines. How to Represent Texts in Input Space?", Machine Learning 46, 2002, pp. 423 - 444.

[3] Sung-Bae Cho, Jee-Haeng Lee, Learning Neural Network Ensemble for Practical Text Classification, Lecture Notes in Computer Science, Volume 2690, Aug 2003, Pages 1032 – 1036.

[4] Soucy P. and Mineau G., "Feature Selection Strategies for Text Categorization", AI 2003, LNAI 2671, 2003, pp. 505-509 .