

Introduction

The ever-growing digital realms of the Internet have over the last decades inarguably had massive impact on how the majority of society go about their daily lives. Even within said decades, we have seen drastic evolutions in how the Internet is being used. Thanks to the increasing popularity of websites with consumer-generated media (CGM), the general public can quickly and with ease catch up on the latest buzz and express their views to thousands or even millions of others.

With consumers taking their conversations online, effectively measuring, interpreting and acting on this online "Word-of-Mouth" (WoM) has shown to be a key factor in keeping a competitive and financial advantage for consumer centered brands. One example is the area of Product Recall Management, where it's imperative to quickly detect customers' problems and issues to determine if a product needs to be recalled from the market.

The challenge in monitoring online WoM is the sheer volume of information to be traversed. It is hardly practical nor efficient to manually oversee any large number of CGM websites, and even using traditional search engines will typically yield an unmanageable amount of information.

Our hypothesis is that we can use a system based on pattern recognition with machine learning to classify CGM as either problem descriptions or not problem descriptions, and additionally as informative problem descriptions or non-informative ones. This would, for instance, enable us to make an automated classifier

detecting when people have problems with a product.

Methods

To train and validate the classifier, we will be using sets of posts from online discussion boards that we manually classify beforehand (samples). A sample is presented to the classifiers in the form of a feature vector, each feature being a measurable characteristic of the sample. Since we are working with text classification, we'll use word frequencies or n-gram frequencies as features.

For the first classification task, to classify as problem description or not problem description, we implemented and investigated three well known algorithms for pattern recognition with machine learning: **Naive Bayes**, **k Nearest Neighbor** and **Self-Organizing Maps**. All three can in fact be called "naive", because they all assume that the words and their order are independent, i.e. they do not consider the semantics of the text. This means that they are relatively simple, but research has shown that they are nevertheless capable of producing

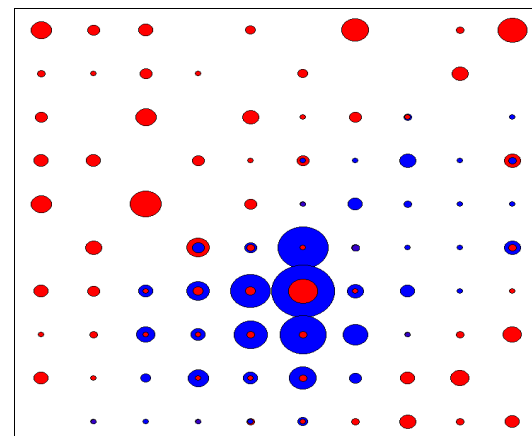


Figure 1: Problem posts (blue) and not problem posts (red) in a SOM.

impressive results. Each algorithm has its own set of parameters that influences the accuracy depending on the type of classification.

Of the three, Self-Organizing Maps is the odd one out. In its bare essence it's not a classifier, but a technique able to visualize the similarity of samples. It is a form of Multidimensional Scaling, in that we take the feature vectors that can be in thousands of dimensions, and scale it down to a low dimensional map.

For the second classification task, we use the technique that proved most accurate before and test on informative/non-informative samples.

Results

The results from the first classification task, where we classify a sample as a problem description or not a problem description, are impressive. As can be seen in figure 2, Naive Bayes reached an accuracy of almost 97%, with k Nearest Neighbor close behind at nearly 94%. We also observe that the maximum accuracies are reached after just 200-400 samples of training. Self-Organizing Maps boasts somewhat

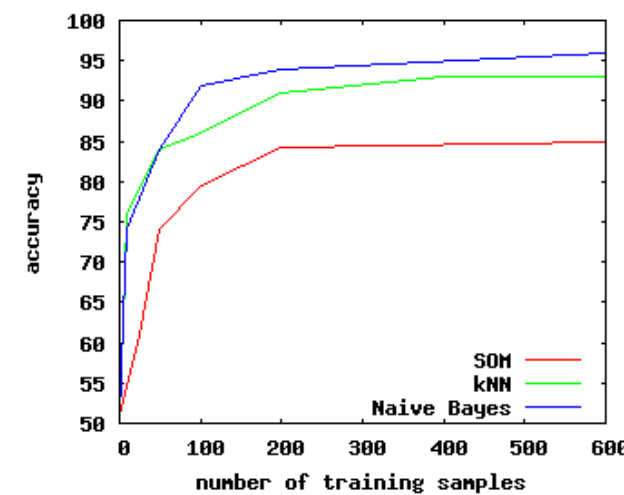


Figure 2: Classifier accuracy comparison

less accuracy for our classification, at around 85% with the parameters we found to be most favorable. It does share many of the same heuristics as k Nearest Neighbor, but some information is lost during the multidimensional scaling, resulting in some loss of accuracy as well.

The second classification task, to classify informative and non-informative problem descriptions, was unfortunately not as successful as the first, even using all our training samples. In sporadic test runs we would see 55% or even 60% accuracy, but the overall average was always below 55%. In comparison, simply guessing the classification yields 50%.

Conclusion

In our thesis we have shown that we can indeed use a system to classify texts from CGM websites (discussion board posts in our case) as problem descriptions and not problem descriptions using pattern classification with machine learning. The three algorithms we have used all produce accurate results on our online discussion board samples, especially Naive Bayes and k Nearest Neighbor.

Regarding the second part of our hypothesis, classifying based on a sample's informativeness, our results show that the same techniques do not yield satisfactory accuracies. It's apparent that the classifiers are not able to learn what distinguishes the classes based on word or n-gram frequencies alone. In reality, it's more measure informativeness on a linear scale, not as two distinct classes, and we can assume this adds to the problem. It's reasonable to believe that we would have to use different features, perhaps incorporating some semantic analysis to accurately perform a such classification.