Chapter 14

The Responsio email management system

M. Kockelkorn¹ & T. Scheffer²
¹Strato AG, Berlin, Germany
²Humboldt-Universität zu Berlin, Department of Computer Science, Berlin, Germany

Abstract

Many individuals, organizations, and companies have to answer large amounts of emails. Often, most of these emails contain variations of relatively few frequently asked questions. The Responsio email management system addresses the problem of predicting which of several frequently used answers a user will choose to respond to an email. Our approach effectively utilizes the data that are typically available in this setting: inbound and outbound emails stored on a server. We map the problem to a semi-supervised classification problem. We briefly discuss a case study based on emails sent to a corporate customer service department.

1 Introduction

Companies allocate considerable economic resources to communication with their customers. A continuously increasing share of this communication takes place via email; marketing, sales and customer service departments as well as dedicated call centers have to process high volumes of emails, many of them containing repetitive routine questions. It appears overly ambitious to completely automate this process; however, any software support that leads to a significant productivity increase is already greatly beneficial.

Our approach to support this process is to predict which answer a user will most likely send in reply to an incoming email, and to propose this answer to the user. The user, however, is free to modify – or to dismiss – the proposed answer. We utilize the available data (stored inbound and outbound emails) and transform the problem into a semi-supervised text classification problem.

The Support Vector Machine [1] is generally considered to be one of the most accurate text classification algorithms. Among the known algorithms that utilize
unlabeled data, the multi-view approach [2] and the transductive SVM [3] are distinguished by desirable theoretical properties and promising empirical results.

2 Email answering by semi-supervised text classification

We consider the problem of predicting which of the given standard answers $A_1$, \ldots, $A_n$, a user will reply to an email. In order to learn a predictor, we are given a repository \{ $x_1$, \ldots, $x_m$ \} of emails received in the past, and a repository \{ $y_1$, \ldots, $y_k$ \} of outbound emails. Typically, these repositories contain at least hundreds, but often (at least) thousands of emails stored on a corporate email server. The problem would become a supervised classification problem if we knew, for each inbound email, which of the standard answers had been used. Unfortunately, for two reasons, this is not the case.

Firstly, it is not trivial to identify which outbound email has been sent in reply to a particular inbound email; neither the emails themselves nor the internal data structures of email clients contain explicit references. Secondly, when an outbound email does not exactly match one of the standard answers, this does not necessarily mean that none of the standard answers is a correct prediction. The user could have written an answer that is semantically equivalent to one of the answers but uses a few different words.

What is the appropriate utility criterion for this problem? Our goal is to assist the user by proposing answers to emails. Whenever we propose the answer that the user accepts, he or she benefits; whereas, when we propose a different answer, the user has to manually select or write an answer. Hence, the optimal predictor proposes the answer which is most likely given $x$ (i.e. maximizes $P(A_i | x)$). Keeping these characteristics in mind, we can pose the problem as follows.

**Problem Setting** Given is a repository of inbound emails and a repository of outbound emails in which instances of standard answers $A_1$, \ldots, $A_n$ occur. There is no explicit mapping between inbound and outbound mails. The task is to generate a predictor for the most likely answer to a new inbound email.

We use the following heuristic to identify cases in which an outbound email is a response to a particular inbound email. The recipient has to match the sender of the inbound email, and the subject lines have to match up to a prefix (“Re:” for English email clients). Furthermore, the outbound email has to be sent while the inbound email was visible in one of the active windows. This rule enables us to identify some inbound emails as positive examples for the answers. For many emails, however, we will not be able to identify the corresponding answer. These emails are unlabeled examples in the underlying classification problem.

We can reasonably assume that no two different standard answers are semantically equivalent. Hence, when an email has been answered by a standard answer $A_i$, we can conclude that it is a negative example for all $A_j$. Thus, we have a small set of positive and negative examples for each $A_i$; additionally, we have a large quantity of emails for which we cannot determine the appropriate answer.
We have thus reduced the problem to a semi-supervised text classification problem. In order to apply the SVM, we furthermore transform this problem with $n$ different classes (standard answers) into $n$ binary classification problems.

3 Responsio email management system

The key design principle of Responsio is that, once it is installed and the standard answers are entered, it does not require any extra effort from the user. The system observes inbound and outbound emails, but does not require explicit feedback. Responsio is an add-on to Microsoft Outlook. The control elements (shown in fig. 1) are loaded into Outlook as a COM object.

When an email is selected, the COM add-in sends the email body to a second process which first identifies the language of the email, executes the language specific classifiers and determines the posterior probabilities of the configured answers. The classifier process notifies the add-in of the most likely answer which is displayed in the field marked. When the user clicks the “auto answer” button (circled in fig. 1), Responsio extracts the name of the sender, identifies the gender, and formulates a salutation line followed by the proposed standard answer. The system opens a reply window with the proposed answer filled in.

Whenever an email is sent, Responsio identifies whether the outbound mail is a reply to an inbound mail; when the sent email includes one of the standard answers, the inbound mail is filed into the list of example mails for that answer. These examples are displayed in the Responsio manager window. It is also possible to manually drag and drop emails into the example folders. Whenever an example list changes, a model update request is posted. The Responsio training unit processes these requests and starts the learning algorithm.

![Figure 1: Responsio add-in to Microsoft Outlook.](image-url)
4 Case study

The data used in this study were provided by the TELES European Internet Academy (TEIA), an education provider that offers classes held via the internet. TEIA uses the Responsio system in its customer case department. In order to evaluate the performance of the predictors, we manually labeled all inbound emails within a certain period with the matching answer.

Roughly 72% of all emails received can be answered by one of nine standard answers. 42% of all emails (224 instances) are “product inquiries”; 10% (56 instances) correspond to category “server down”; 4% correspond to “send access data” (22 instances) and “degrees offered” (21 instances). 3% (15 instances) correspond to “free trial period”, 2% to “government stipends” and “homework late” (13 instances each), 1% of emails fall into “TELES product inquiries” and “scholarships” (7 instances each). Only 28% of emails received require individual answers.

We use the receiver operating characteristic (ROC) analysis to assess the performance of the trained decision functions [4]. The ROC curve of a decision function plots the number of true positives against the number of false positives. The area under the ROC curve (the AUC performance) is equal to the probability that, when we draw one positive and one negative example at random, the decision function will assign a higher value to the positive example than to the negative. We performed between 7 and 20-fold stratified cross validation and averaged the AUC values measured on the held out data.

First, we studied the performance of a decision function provided by the Naive Bayes algorithm as well as the Support Vector Machine, SVMlight [1]. Figure 2 shows that the SVM impressively outperforms Naive Bayes in all cases except for one (TELES product inquiries). Remarkably, the SVM is able to identify even very specialized questions with as little as seven positive examples with between 80 and 95% AUC performance.

![Figure 2: ROC curves of Naive Bayes and SVMlight.](image-url)
While the transductive SVM [3] does improve the AUC performance for classes with few labeled instances, it increases, on the other hand, the training time from several seconds to several minutes – which is prohibitive for a desktop application. Similarly, the co-training algorithm increases recognition performance if (and only if) only few labeled examples are available (see fig. 3) and increases computation time roughly by a factor of 200. Utilizing unlabeled examples in semi-supervised classification both effectively and efficiently remains one of the main challenges in text classification.

5 Discussion

We have discussed the problem of identifying instances of frequently asked questions in emails, using only stored inbound and outbound emails as training data. Our empirical data show that identifying a relatively small set of standard questions automatically is feasible; we obtained AUC performance of between 80 and 95% using as little as seven labeled positive examples.

Several email assistance systems have been presented. For instance, [4,5] use text classifiers in order to predict the correct folder for an email. In contrast to these studies, we studied the feasibility of identifying instances of particular questions rather than general subject categories. Related is the problem of filtering spam email. Keyword based approaches, Naive Bayes (e.g. [6]) and rule-based approaches have been compared.

Acknowledgment

This work has partially been supported by the German Science Foundation DFG under grant SCHE 540/10-1.

References


