EmFORE: Learning Email Folder Classification Rules by Demonstration

Mukul Singh¹, Gust Verbruggen², José Cambronero³, Vu Le³, Sumit Gulwani³

¹Microsoft India
²Microsoft Belgium
³Microsoft Redmond

{singhmukul, gverbruggen, jcambronero, levu, sumitg}@microsoft.com

Abstract

Tools that help with email folder management are limited, as users have to manually write rules to assign emails to folders. We present a demonstration of EmFORE, an iterative learning system that automatically learns and updates such rules from observations. EmFORE is fast enough to suggest and update rules in real time and suppresses mails with low confidence to reduce the number of false positives. EmFORE can use different rule grammars, and thus be adapted to different clients, without changing the user experience. Previous methods do not learn rules, require complete retraining or multiple new examples after making a mistake, and do not distinguish between inbox and other folders. EmFORE learns rules incrementally and can also abstain from predicting, making it an ideal candidate for integration in email clients.

Introduction

Most email services provide tools that help users manage their inbox. Spam prediction reduces inbox clutter. Estimating the significance of emails [Alrashed et al. 2019] helps users focus on important emails in the Focused inbox in Outlook and Priority inbox in Gmail. Search helps users to quickly find specific emails [Mackenzie et al. 2019].

Despite the popularity of emails, adding folder rules to clients is a very challenging task that requires the user to be aware of the underlying rule logic. Furthermore, creating and updating such rules require navigation through multiple menus and inputs, which is very tedious for a day to day task.

In this paper, we present a demonstration of the first system (EmFORE) that learns email folder classification rules by example [Singh et al. 2023a]. EmFORE observes a user moving emails into folders and uses the example mails to learn a rule for each folder. We draw inspiration from the successful application of the programming by example paradigm in commercial products like Excel [Gulwani 2011; Singh et al. 2023b] and Visual Studio [Miltner et al. 2019]. The rules learned by EmFORE consist of generic propositions—which describe properties of emails—that are combined according to the rule grammars found in different popular email clients.

In summary, we present a demonstration of an online algorithm for learning email folder classification rules from a few email examples. Our algorithm is fast and small enough for deployment, can be configured to have fewer false positives and performs better than past systems for email classification.

Approach

Our system takes inspiration from mathematical induction. Let \( R \) be a set of rules consistent with the current emails for a user, where a rule is a formula in propositional logic. If the prediction \( f = R(m) \) for a new email \( m \) is wrong, as indicated by the user moving the email to folder \( f^* \), we update the rule to be consistent with all previous emails and the new email. We introduce three components for doing so: a state \( S \) that tracks candidate propositions for each folder, a space of rules over which \( R \) is learned, and an algorithm for updating \( R \). We now briefly describe each component and refer the interested reader to Singh et al. [2023a] for details.

Example 1. An example of a rule is \( \text{InFrom}('\text{straw}') \lor \text{InTo}('\text{straw}') \) which states that either sender or receiver should contain the word “straw.”

State

The state keeps track of the candidate propositions \( S_f \) for each folder \( f \) and ensures that every proposition \( p \in S_f \) satisfies emails \( (m_i, f_i) \) if and only if \( f_i = f \). Not every proposition must satisfy all mails. Any folder with an empty set of candidate propositions cannot be covered by a rule. Candidate propositions are generated for an email from a set of templates by substituting a placeholder \( e \) with a string constant. Candidate propositions for each folder are ranked to allow greedy selection of promising ones when building rules. This ranking takes into account (1) similarity between the string constants in the proposition and the folder name, (2) average similarity to string constants of the current rule for that folder, and (3) the type of proposition. Similarities are computed with Jaro-Winkler string similarity. Each of these properties yields a score, which are summed to obtain a final score. Whenever an email \( (m_i, f_i) \) comes in and propositions \( P_f \) are generated, we add them to \( S_f \) while maintaining the ranking and remove them from \( S_f \) for \( f_i \neq f_j \).

Rule Space

We limit \( R = \{ (R_f, f) \} \) to each folder \( f \) being represented by a single rule \( R_f \) in disjunctive normal form (DNF) and write \( R = R_f \) for brevity. As every logical formula can be written in DNF, we do not lose expressivity.
We walk through the ten emails in this section explaining how we have implemented EFORE as an extension for Outlook (one of the most popular email clients). This demo consists of a hypothetical scenario of a user receiving ten emails and moving them to two specific folders: papers and work. We walk through the ten emails in this section explaining how EFORE assists the user in categorizing these into folders. In the remainder of this section, we will only show the email sender and subject for brevity.

1. **Inbox**
   - **Subject:** Fwd: Check out this paper!
   - **From:** gust.verbruggen@kuleuven.be
   - **Subject:** Slide deck for upcoming workshop
   - **From:** singhmukul@microsoft.com
   - **Subject:** Upcoming demo presentation
   - **From:** singhmukul@microsoft.com
   - **Subject:** Coffee now free again!
   - **From:** gverbruggen@microsoft.com
   - **Subject:** New guidelines for working from home
   - **From:** gverbruggen@microsoft.com
   - **Subject:** Cool paper
   - **From:** redacted@hotmail.com
   - **Subject:** Upcoming demo presentation

![Image of the demo](image.png)

**Figure 1:** Overview of our demo. (1) A user moves an email to papers and a rule is learned. (2) Next email about papers is correctly moved to that folder. (3) An email about work is suppressed, because the rule for work was recently updated and the new email is not similar to past emails in that folder.

### Updating Rules

When a new email \((m_\ast, f^\ast)\) comes in, each of the rules \(R_f \in \mathcal{R}\) can be updated. If \(m_\ast \neq R_f\), then it requires generalization. Any rule \(R_f\) with \(f \neq f^\ast\) and \(m_\ast \models R_f\) requires specialization. Both steps follow the same pattern of first trying to replace existing propositions and only adding disjuncts (generalize) or conjuncts (specialize) if replacement fails. Candidates for replacement or addition are greedily selected from the state.

### Suppressing Rules

We use a linear combination of five features with sigmoid activation to predict whether a prediction should be suppressed or not. Used features are rule length, number of consecutive correct predictions by the rule, running accuracy for the folder, average running accuracy of specific disjuncts that the mail satisfied, and folder size.

### Demo

We have implemented EFORE as an extension for Outlook (one of the most popular email clients). This demo consists of a hypothetical scenario of a user receiving ten emails and moving them to two specific folders: papers and work. We walk through the ten emails in this section explaining how EFORE assists the user in categorizing these into folders. In the remainder of this section, we will only show the email sender and subject for brevity.

- **From:** redacted@gmail.com
  - **Subject:** Drinks after work
  - The first email is a generic personal message that the user does not want to group in a folder—it remains in the inbox.

- **From:** redacted@gmail.com
  - **Subject:** Fwd: Check out this paper!
  - The second email is about a new research paper and the user moves it into a new papers folder. EFORE instantly learns the rule \(\text{InSubject}(\text{"paper"})\) for the folder, since propositions \(\text{InSubject}\) with arguments \(\text{"papers"}\) that are close to the folder name are highly ranked. Updating a rule remains instantaneous for inboxes with hundreds of emails.

- **From:** redacted@hotmail.com
  - **Subject:** Cool paper
  - The third email is also about a paper and EFORE assigns it to papers. Since this mail is similar to the mail already in that folder, the prediction is not suppressed.

- **From:** gverbruggen@microsoft.com
  - **Subject:** New guidelines for working from home
  - The fifth email is regarding work and the user wants to group work related emails. They add it to a new folder work. EFORE learns the rule \(\text{InSubject}(\text{"work"})\) for the folder.

- **From:** singhmukul@microsoft.com
  - **Subject:** Upcoming demo presentation
  - The sixth email not assigned to a folder, but the user adds it to work. A rule update is triggered (a mild form of concept drift) and the rule is updated to \(\text{InFrom}(\text{"microsoft"})\).

- **From:** gverbruggen@microsoft.com
  - **Subject:** Coffee now free again!
  - The seventh email satisfies the rule for work, but it is suppressed because the remainder of the email is not similar enough to those in the folder. In general, suppression reduces the number of false positives. The user moves it to the correct folder, but a rule update is not triggered.

- **From:** singhmukul@microsoft.com
  - **Subject:** Slide deck for upcoming workshop
  - The eighth mail satisfies the rule for work and is correctly assigned to the work by EFORE.

- **From:** gverbruggen@kuleuven.be
  - **Subject:** Did you check out this new paper?

As expected, EFORE correctly assigns the ninth mail to papers as it satisfies the rule for the folder.

In this demo we show how EFORE learns and updates rules in real time. In our experiments, we find that EFORE outperforms previous systems [Dehghani, Shakery, and Mirian 2016; Carmona-Cejudo et al. 2013] by 20%-25% in correct decision rate on Enron [Klimt and Yang 2004] and Avocado [Oard et al. 2015] email datasets over both online and offline evaluation setups.

### Conclusion

We demo EFORE, a system for learning email folder classification rules by observation. EFORE learns and updates rules in disjunctive normal form as new emails come in and learns to default to the inbox when a good rule cannot be found. Our evaluation shows that EFORE is fast enough to update after every email, learns rules from few observations, is more robust than previous approaches, requires fewer emails to be stored than previous approaches, and can be tuned for the expressivity of different email clients.
References


