EMFORE: Learning Email Folder Classification Rules by Demonstration

Mukul Singh[1], **Gust Verbruggen**[2], **José Cambronero**[3], **Vu Le**[3], **Sumit Gulwani**[3]

[1]Microsoft India
[2]Microsoft Belgium
[3]Microsoft 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 learns 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 $\mathcal{R}$ 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 $f = \mathcal{R}(m_\star)$ for a new email $m_\star$ is wrong, as indicated by the user moving the email to folder $f^\star$, we update the rule to be consistent with all previous emails and the new email. We introduce three components for doing so: a state $\mathcal{S}$ that tracks candidate propositions for each folder, a space of rules over which $\mathcal{R}$ is learned, and an algorithm for updating $\mathcal{R}$. We now brielfy describe each component and refer the interested reader to Singh et al. [2023a] for details.

Example 1. *An example of a rule is InFrom("straw") $\vee$ InTo("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_i are generated, we add them to S_{f_i} while maintaining the ranking and remove them from S_{f_j} where $f_i \neq f_j$.

Rule Space

We limit $\mathcal{R} = [(R_f, f)]$ to each folder f being represented by a single rule R_f in disjunctive normal form (DNF) and write $\mathcal{R} = [R_f]$ for brevity. As every logical formula can be written in DNF, we do not lose expressivity.

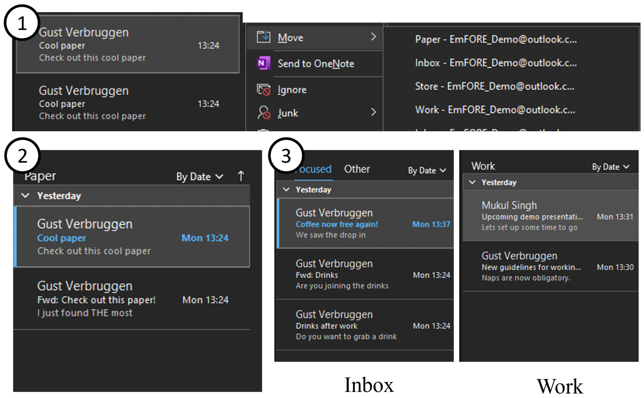

Figure 1: Overview of our demo. (1) A user moves an email to paper and a rule is learned (2) Next email about papers is correctly moved to that folder. (3) An email about work is suppressed, because the rule for work was recently updated and the new email is not similar to past emails in that folder.

Updating Rules

When a new email $(m_\star, f^\star)$ comes in, each of the rules $R_f \in \mathcal{R}$ can be updated. If $m_\star \nvDash R_{f^\star}$ then it requires generalization. Any rule R_f with $f \neq f^\star$ and $m_\star \vDash R_f$ 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 EMFORE 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 EMFORE 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. EMFORE instantly learns the rule InSubject(“paper”) for the folder, since propositions (InSubject) with arguments (“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 EMFORE assigns it to papers. Since this mail is similar to the mail already in that folder, the prediction is not suppressed.

From: redacted@outlook.com
Subject: Fwd: drinks

The fourth email is again a personal message and the user does not want move it to any folder—it remains in the inbox.

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. EMFORE learns the rule InSubject(“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 InFrom(“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 EMFORE.

From: gust.verbruggen@kuleuven.be
Subject: Did you check out this new paper?

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

In this demo we show how EMFORE learns and updates rules in real time. In our experiments, we find that EMFORE 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 EMFORE, a system for learning email folder classification rules by observation. EMFORE 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 EMFORE 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

Alrashed, T.; Lee, C.-J.; Bailey, P.; Lin, C.; Shokouhi, M.; and Dumais, S. 2019. Evaluating user actions as a proxy for email significance. In *The World Wide Web Conference*, 26–36.

Carmona-Cejudo, J. M.; Castillo, G.; Baena-García, M.; and Morales-Bueno, R. 2013. A comparative study on feature selection and adaptive strategies for email foldering using the ABC-DynF framework. *Knowledge-Based Systems*, 46: 81–94.

Dehghani, M.; Shakery, A.; and Mirian, M. S. 2016. Alecsa: attentive learning for email categorization using structural aspects. *Knowledge-Based Systems*, 98: 44–54.

Gulwani, S. 2011. Automating string processing in spreadsheets using input-output examples. *ACM Sigplan Notices*, 46(1): 317–330.

Klimt, B.; and Yang, Y. 2004. The enron corpus: A new dataset for email classification research. In *European conference on machine learning*, 217–226. Springer.

Mackenzie, J.; Gupta, K.; Qiao, F.; Awadallah, A. H.; and Shokouhi, M. 2019. Exploring user behavior in email refinding tasks. In *The World Wide Web Conference*, 1245–1255.

Miltner, A.; Gulwani, S.; Le, V.; Leung, A.; Radhakrishna, A.; Soares, G.; Tiwari, A.; and Udupa, A. 2019. On the fly synthesis of edit suggestions. *Proceedings of the ACM on Programming Languages*, 3(OOPSLA): 1–29.

Oard, D.; Webber, W.; Kirsch, D. A.; and Golitsynskiy, S. 2015. Avocado Research Email Collection. https://catalog.ldc.upenn.edu/LDC2015T03. Accessed: 2023-12-15.

Singh, M.; Cambronero, J.; Gulwani, S.; Le, V.; and Verbruggen, G. 2023a. EmFore: Online Learning of Email Folder Classification Rules. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, 2280–2290.

Singh, M.; Sánchez, J. C.; Gulwani, S.; Le, V.; Negreanu, C.; Raza, M.; and Verbruggen, G. 2023b. Cornet: Learning Table Formatting Rules By Example. *Proc. VLDB Endow.*, 16(10): 2632–2644.