CSC510
Automated Online Email Categorization

Group C

Shijie Li, Zhe Yu, Amritanshu Agrawal, Di Chen

Introduction (Jerry)

Outline:

- Problems & Challenges
- Solutions & Results
- Evaluation & Demo

Problems of Current Email Categorization

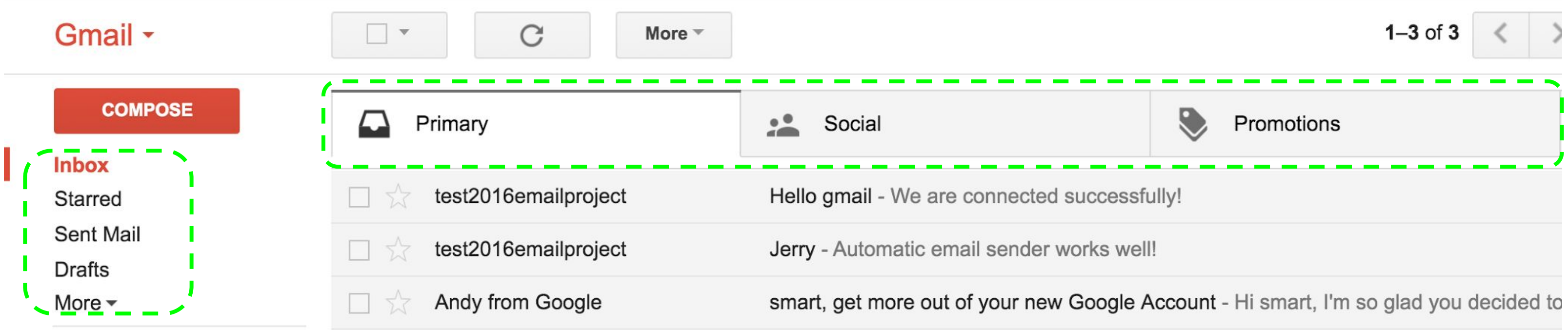
Problem #1

- Limited automatic email categorization:
  - Most Email applications only categorize email into default folders.
  - Not applicable to customized categories created by user.

Problems of Current Email Categorization

Problem #1

- Limited automatic email categorization:
  - Most Email applications only categorize email into default folders.
  - Not applicable to customized categories created by user.



Problems of Current Email Categorization

Problem #2

- Explicit user feedbacks needed:
  - Most Email applications require user to manually assign new labels beyond default categories.
  - Inconvenient for frequent use
  - Inefficient for large amount

Project Goal

User-oriented:

- Personalized
- Private
- Automatic
- Flexible
- Efficient
- Accurate
- Active

THE CHALLENGE

- **Cold Start:**
  - Lack of Training Examples
- **Implicit Feedback:**
  - User is busy/lazy to provide explicit confirmation
- **Unbalanced Importance:**
  - Miss important emails due to incorrect categorization
- **Different Types of Folders:**
  - Not Every Folder is Content-aware
- **Non-text Content:**
  - Attachments, pictures, urls can be useful

THE CHALLENGE

- **Cold Start:**
  - Lack of Training Examples
- **Implicit Feedback:**
  - User is busy/lazy to provide explicit confirmation
- **Unbalanced Importance:**
  - Miss important emails due to incorrect categorization
- **Different Types of Folders:**
  - Not Every Folder is Content-aware
- **Non-text Content:**
  - Attachments, pictures, urls can be useful

THE CHALLENGE

- **Cold Start:**
  - Lack of Training Examples
- **Implicit Feedback:**
  - User is busy/lazy to provide explicit confirmation
- **Unbalanced Importance:**
  - Miss important emails due to incorrect categorization

Lack of Training Examples

- **What we expect:**
  - Hundreds of training examples for each folder in order to achieve a reliable classifier.

- **What we get:**
  - About 5 to 10 emails per folder for training.
  - Tens of rows and thousands of columns in the feature matrix.

Lack of Training Examples

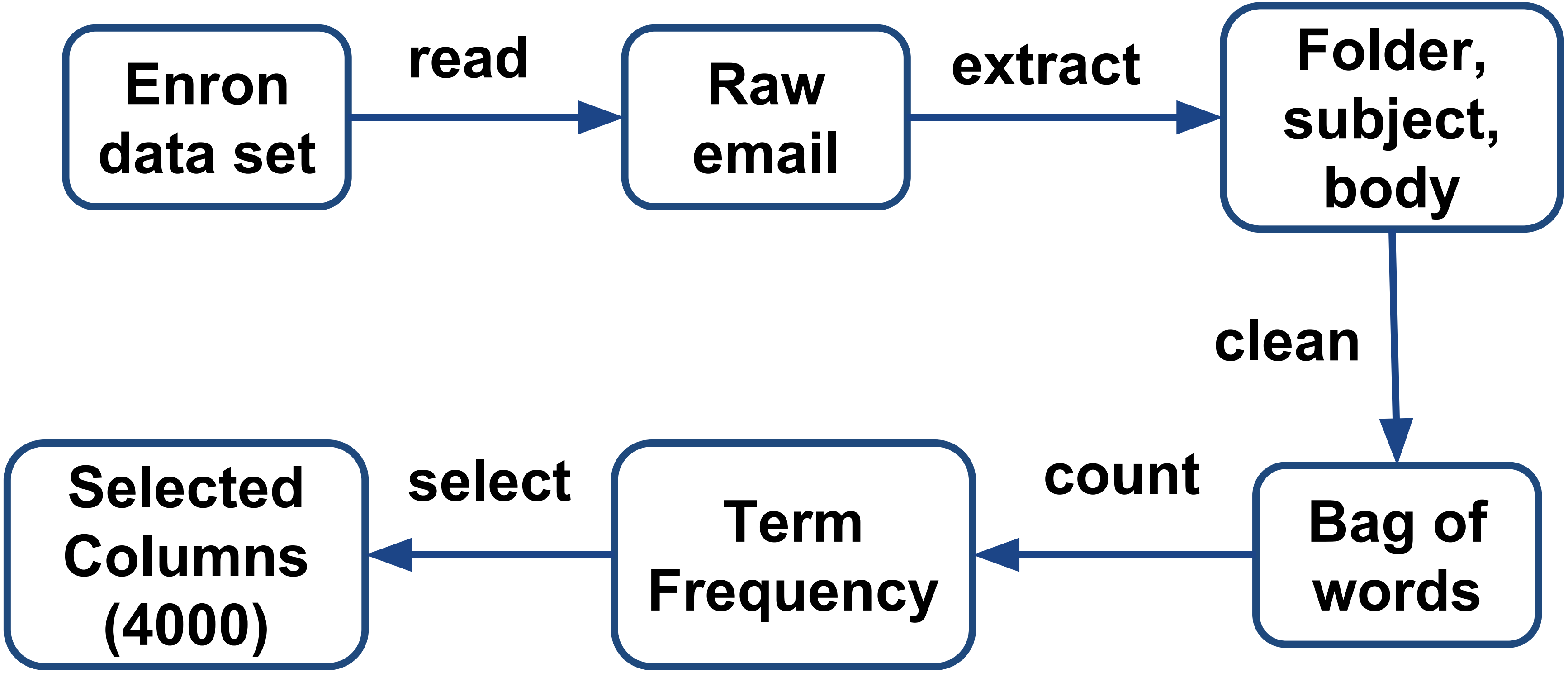
- **Solutions:**
  - **Try different classifiers**. Is there a particular classifier performing better when having few training examples?
  - **Incremental Learning.** Keep retraining the classifier with new emails.
  - **Dimensionality Reduction.** Use LDA to reduce the number of features before training.

Experiment Design

- **Performance metrics:**
  - **Macro F score:** 2*P*R/(P+R)
    - P=Mean(Precision of each folder)
    - R=Mean(Recall of each folder)

- **Data sets:**
  - **7 data sets** from 7 users in Enron data set.
  - **5 to 10 folders** in each data set.
  - **50 emails** at least in each folder.

Preprocessing

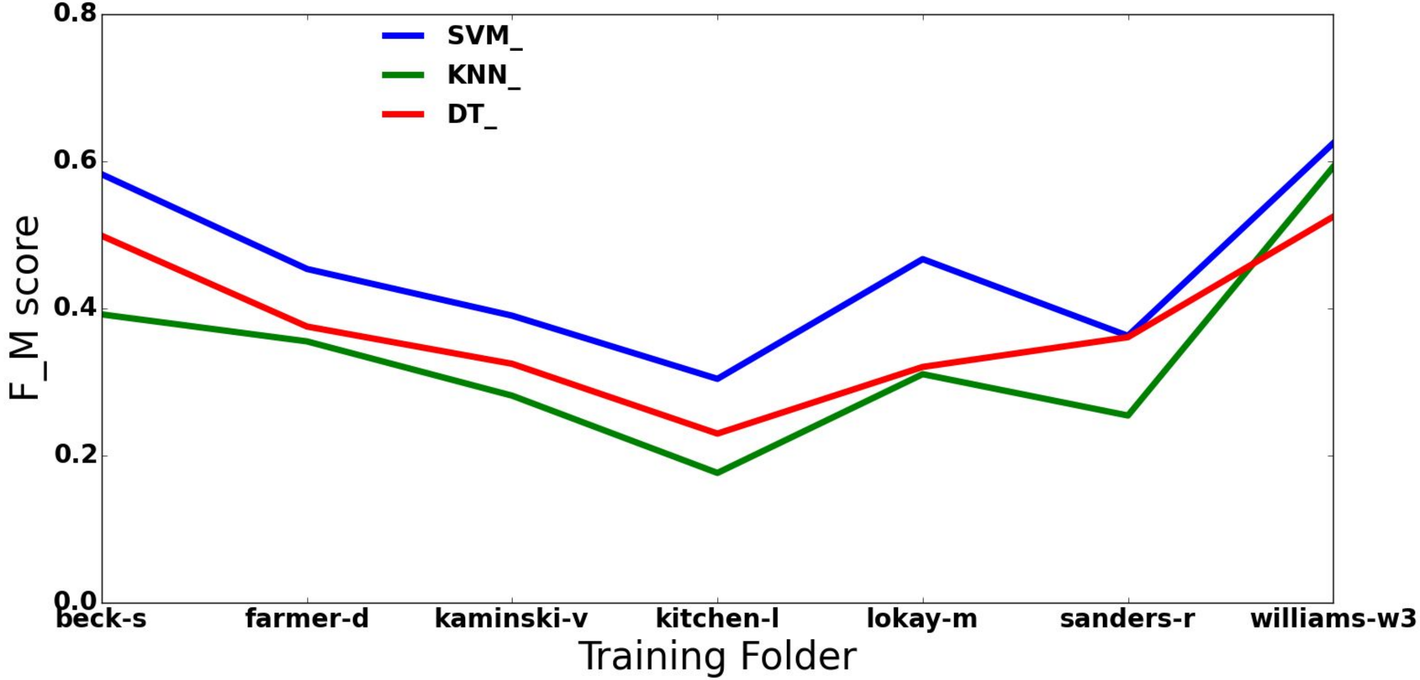


Solution 1: Different classifiers

- **Experiment:**
  - 5 to 10 randomly selected emails in each folder for training.
  - Three classifiers: Decision Tree, K nearest neighbors, support vector machine.
  - Test on all the rest.
  - Repeat 25 times on 7 data sets

Solution 1: Different classifiers

- **Results:**

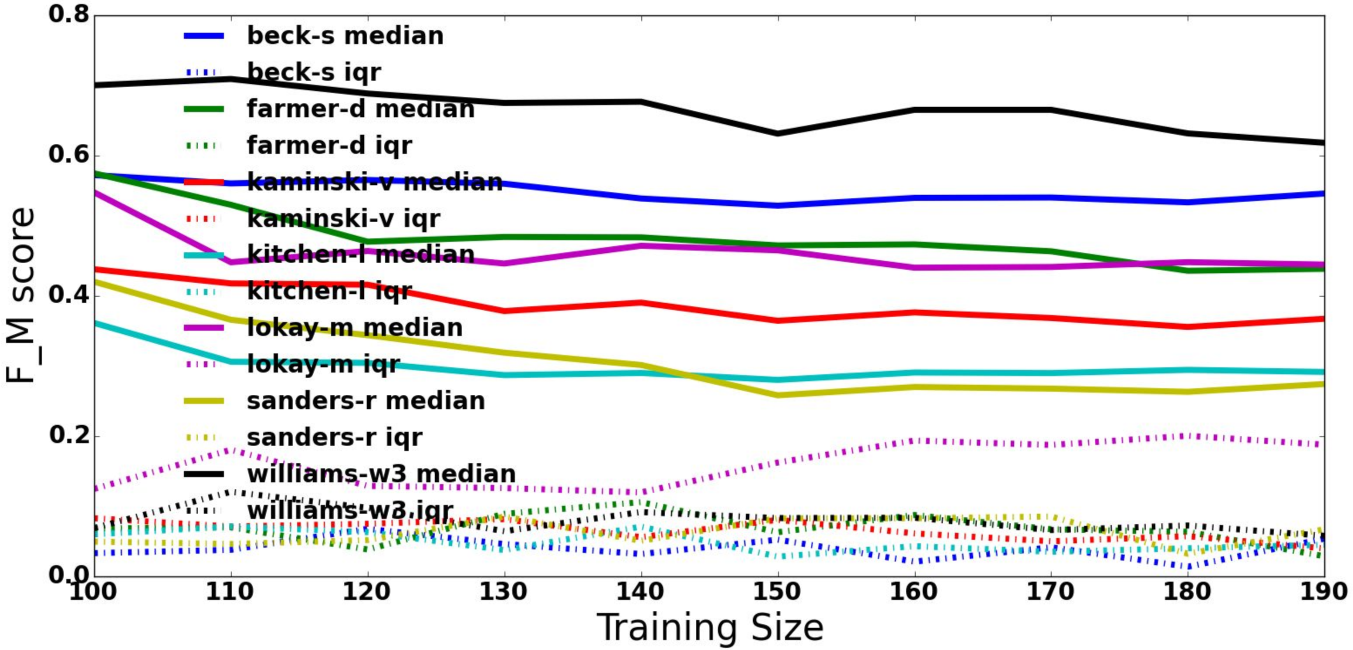


Solution 2: Incremental Learning

- **Experiment:**
  - 5 to 10 randomly selected emails in each folder for initial training.
  - SVM classifier.
  - Three ways to select new emails for retraining. (Brutal, Credit, Wrong)
  - Test on a hold out data set with half the population.
  - Repeat 25 times on 7 data sets

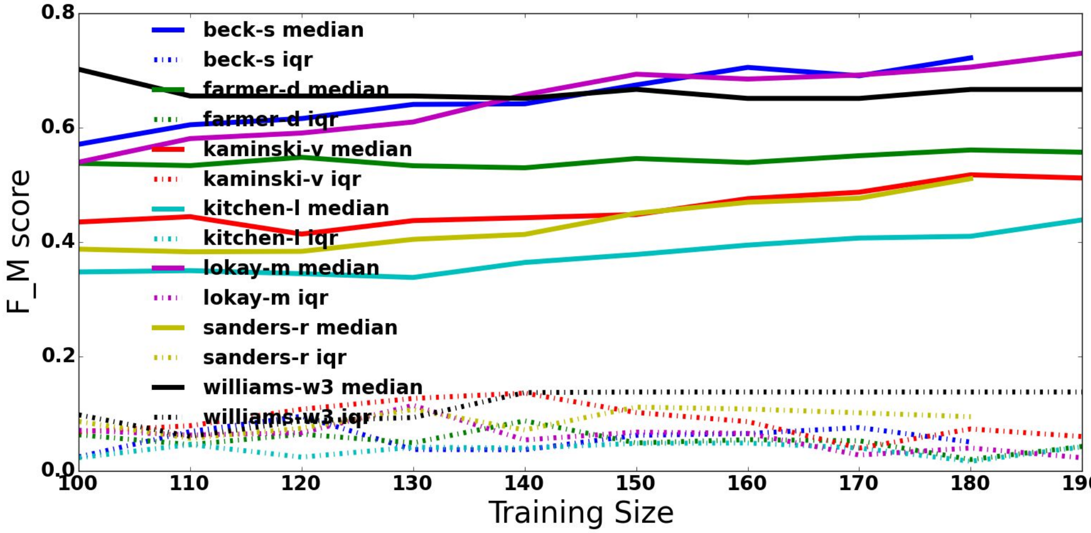
Solution 2: Incremental Learning

- **Brutal:** add everything we have into the training set.



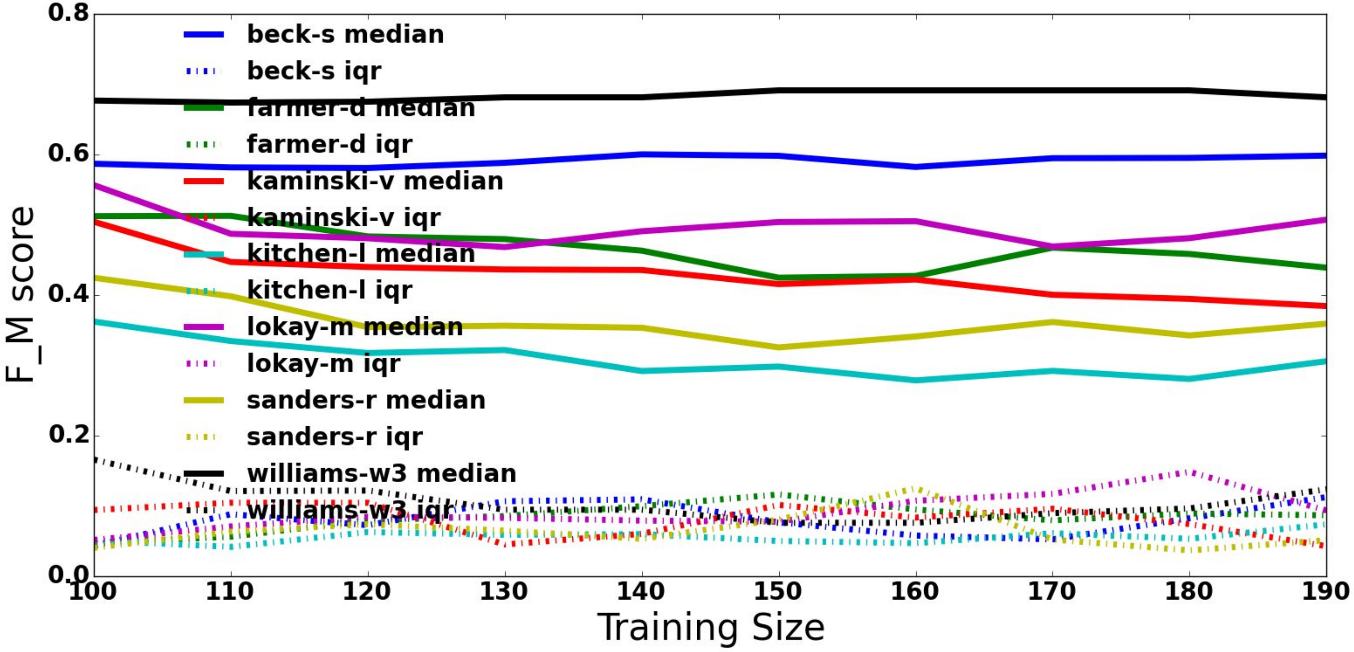
Solution 2: Incremental Learning

- **Credit:**
    - Each email has a credit of 1-Probability (true_folder)
    - Emails with top N credit goes into the training set. N keeps growing.



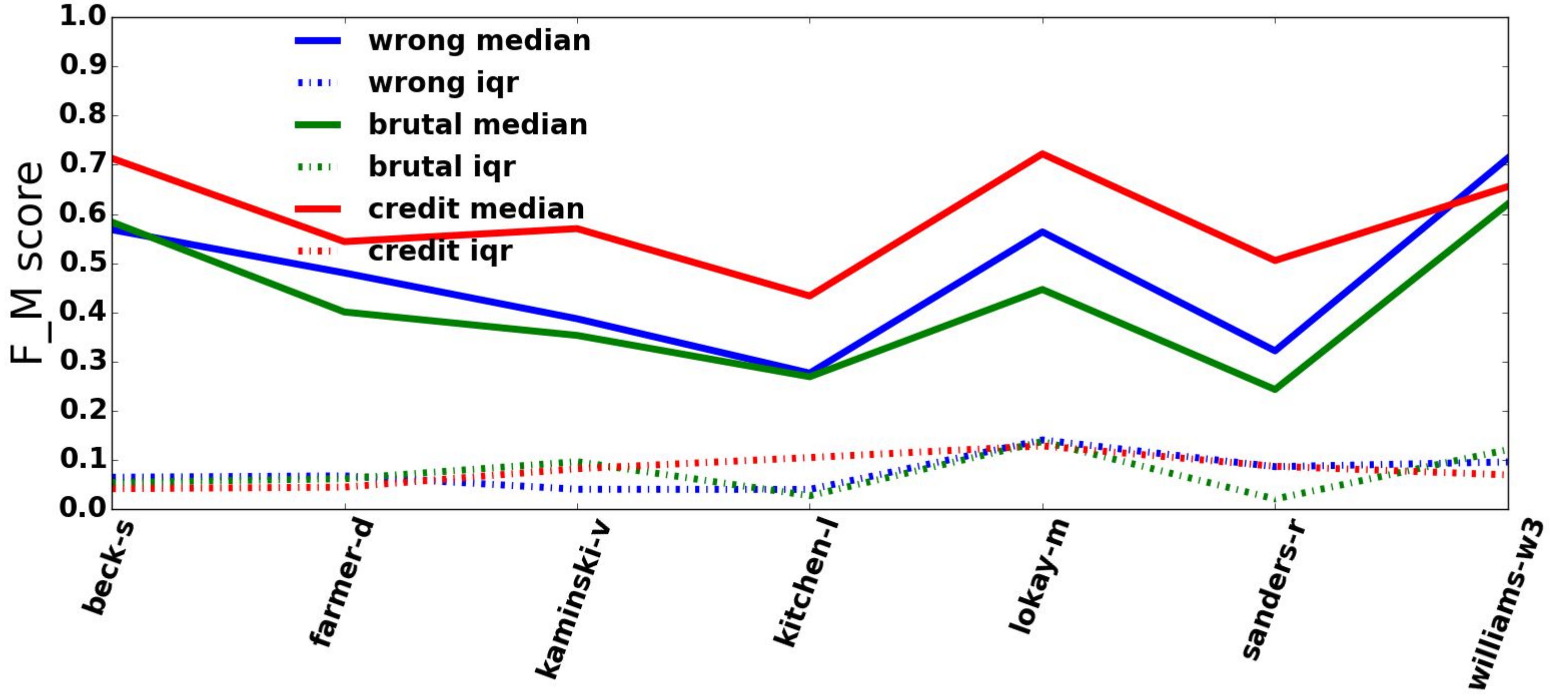
Solution 2: Incremental Learning

- **Wrong:** add every wrongly predicted email into the training set.



Solution 2: Incremental Learning

- **Compare the three**

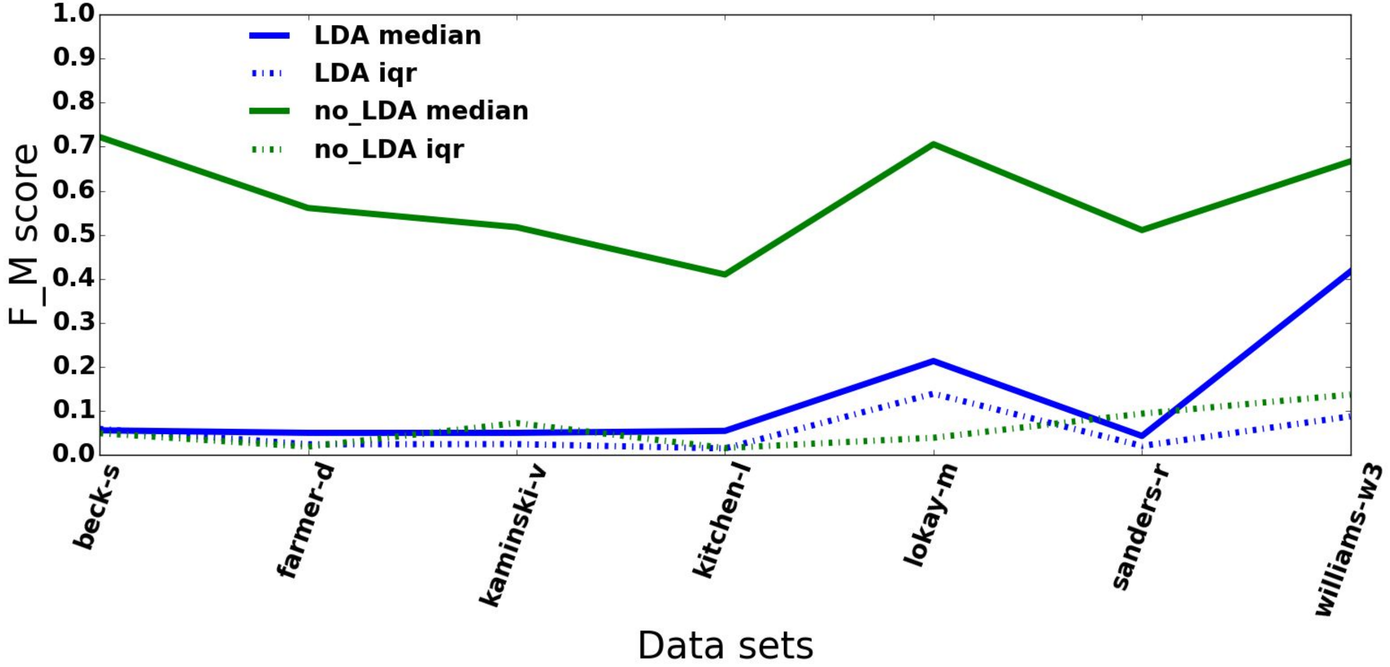


Solution 3: Dimensionality Reduction

- **Experiment:**
  - 5 to 10 randomly selected emails in each folder for initial training.
  - Use LDA to map the 4000 words to 100 topics.
  - Train by SVM.
  - Select new emails for retraining by Credit.
  - Test on a hold out data set with half the population.
  - Repeat 25 times on 7 data sets

Solution 3: Dimensionality Reduction

- **Result**



Conclusion

- **Best solution for lack of training examples:**
  - Train by SVM.
  - Select new emails for retraining by Credit.
  - Do not use LDA for dimensionality reduction.

THE CHALLENGE

- **Cold Start:**
  - Lack of Training Examples
- **Implicit Feedback:**
  - User is busy/lazy to provide explicit confirmation
- **Unbalanced Importance:**
  - Miss important emails due to incorrect categorization

TAKE THE CHALLENGE

No Explicit Feedback

- **What we expect:**
  - Users will do explicit feedback of modifying the label of emails if they are labelled wrong.

- **What we get:**
  - Limited feedback or no feedback at all.

No Explicit Feedback

- **Solutions:**
  - **Simple Implicit Feedback (SIF)**. When the user changes any label, immediately treats all remaining labels as correct.
  - **Implicit Feedback without SIF (IFwoSIF).** Maintain a count of the total number of IF events to reach a minimum threshold.
  - **Implicit Feedback with SIF (IFwSIF).** Inclusion of both the above.

Details

- **The implicit events are defined as follows:**
  - User add or remove a tag on the message;
  - User add or remove a flag from the message;
  - User move the message to a folder;
  - User copy, reply, forward, or print a message;
  - User save an attachment from the message.

Solution

- Literature review suggests to proceed ahead with **Implicit Feedback with SIF (IFwSIF).**
- **IFwSIF** If the user changes a label, then implicit feedback examples are immediately created. Otherwise, continue to count up implicit events to reach a specified threshold.
- **In our design, we just implemented User reading a message from time and again**. The other events could have been implemented.

THE CHALLENGE

- **Cold Start:**
  - Lack of Training Examples
- **Implicit Feedback:**
  - User is busy/lazy to provide explicit confirmation
- **Unbalanced Importance:**
  - Miss important emails due to incorrect categorization

TAKE THE CHALLENGE

Unbalanced Importance

- **What we expect:**
  - Properly classified mails to a single folder.
- **What we get:**
  - Miss important emails due to incorrect categorization.

Solution

- **Multi-folder Categorization:**
  - Changed the problem to multi-label, multi-classification problem.
  - Different weight can be addressed on the labels.
  - The folders with high rank, the threshold can be reduced to allow emails to go into that folder more easily.

GUI Development

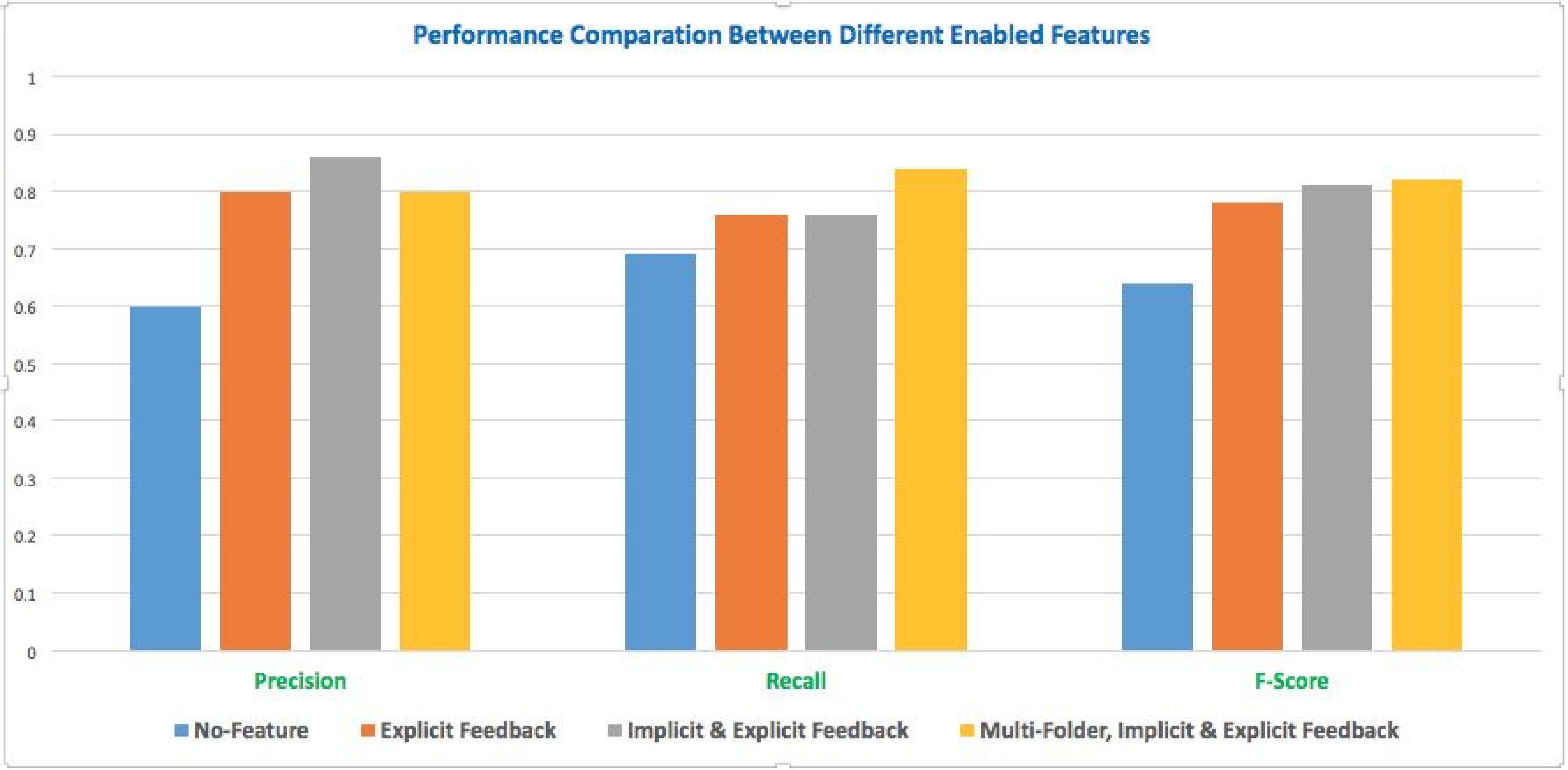
- Developed using Python Tkinter package.
- Gmail as our sample for making the Mailbox GUI.
- **Features**:
  - Read mails
  - Move mails to different folders.
  - Creation of user defined folders.
  - On demand selection of different features which includes
    - Implicit User Feedback
    - Explicit User Feedback
    - Multi-folders

Demo

Telemetry

- Four users operate on GUI for testing, each with four different setups of features.
- User task is to make sure every new coming email end up in the correct folder.
- Data is stored for whether a new coming email is correctly predicted and a Macro Precision, Recall, F score is calculated for each experiment.

Results


Performance Comparation Between Different Enabled Features
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
Precision
Recall
F-Score
No-Feature
Explicit Feedback
Implicit & Explicit Feedback
Multi-Folder, Implicit & Explicit Feedback

Future Work

- Test more.
- Continue on challenges.
- Build a real product and test more.

Reference

[1] R. Bekkerman. Automatic categorization of email into folders: Benchmark experiments on enron and sri corpora. 2004.

[2] V. Bellotti, N. Ducheneaut, M. Howard, and I. Smith. Taking email to task: the design and evaluation of a task management centered email tool. In Proceedings of the SIGCHI conference on Human factors in computing systems, pages 345-352. ACM, 2003.

[3] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. the Journal of machine Learning research, 3:993-1022, 2003.

[4] V. R. Carvalho and W. W. Cohen. Learning to extract signature and reply lines from email. In Proceedings of the Conference on Email and Anti-Spam, volume 2004, 2004.

[5] G. V. Cormack. Email spam ltering: A systematic review. Foundations and Trends in Information Retrieval, 1(4):335-455, 2007.

[6] M. Dehghani, A. Shakery, and M. S. Mirian. Alecsa: Attentive learning for email categorization using structural aspects. Knowledge-Based Systems, 201

[7] B. Klimt and Y. Yang. The enron corpus: A new dataset for email classication research. Pages 217-226, 2004.

[8] H. Murakoshi, A. Shimazu, and K. Ochimizu. Construction of deliberation structure in e-mail communication. Computational Intelligence, 16(4):570-577, 2000.

[9] M. Sokolova and G. Lapalme. A systematic analysis of performance measures for classication tasks. Information Processing & Management, 45(4):427-437, 2009.

[10] M. S. Sorower, M. Slater, and T. G. Dietterich. Improving automated email tagging with implicit feedback. Pages 201-211, 2015.

[11] R. Sproat, J. Hu, and H. Chen. Emu: An e-mail preprocessor for text-to-speech. In Multimedia Signal Processing, 1998 IEEE Second Workshop on, pages239-244. IEEE, 1998.

Thank You!

Q&A