

Goal: build an AI that can go through an email inbox and delete all emails that are unstarred emails. Do NOT delete starred emails.

Adding Task: Develop an email inbox scraping algorithm that can identify starred and unstarred emails

Adding Task: Create a filtering system that can move unstarred emails to trash

Adding Task: Implement a user interface that allows users to easily activate and deactivate the automated filtering system

Adding Task: Develop an email inbox scraping algorithm that can identify starred and unstarred emails

Executing "Develop an email inbox scraping algorithm that can identify starred and unstarred emails" As an AI agent, my first step would be to define the terms "starred" and "unstarred" in the context of email inboxes. Typically, stars are used to mark important or noteworthy emails that the user wants to keep track of. Unstarred emails, on the other hand, are those that the user has not marked as important or noteworthy.

To develop an email inbox scraping algorithm that can identify starred and unstarred emails, I would consider the following steps:

1. Connect to the user's email inbox using a secure authentication mechanism.
2. Retrieve a list of all emails in the inbox, including their metadata such as sender, receiver, subject, date, and time.
3. For each email in the list, check if it contains a star marker or not.
4. If the email contains a star marker, mark it as "starred" in the algorithm's database.
5. If the email does not contain a star marker, mark it as "unstarred" in the algorithm's database.
6. Once all emails in the inbox have been evaluated, return the list of starred and unstarred emails to the user.

With this algorithm in place, it would be possible to identify which emails to delete based on the user's objective of deleting unstarred emails. The AI agent could then be programmed to carry out this task, either automatically or by prompting the user for confirmation before deleting each email.

Executing "Develop an email inbox scraping algorithm that can identify starred and unstarred emails" As an AI agent, my first step would be to define the terms "starred" and "unstarred" in the context of email inboxes. Typically, stars are used to mark important or noteworthy emails that the user wants to keep track of. Unstarred emails, on the other hand, are those that the user has not

marked as important or noteworthy.

To develop an email inbox scraping algorithm that can identify starred and unstarred emails, I would consider the following steps:

1. Connect to the user's email inbox using a secure authentication mechanism.
2. Retrieve a list of all emails in the inbox, including their metadata such as sender, receiver, subject, date, and time.
3. For each email in the list, check if it contains a star marker or not.
4. If the email contains a star marker, mark it as "starred" in the algorithm's database.
5. If the email does not contain a star marker, mark it as "unstarred" in the algorithm's database.
6. Once all emails in the inbox have been evaluated, return the list of starred and unstarred emails to the user.

With this algorithm in place, it would be possible to identify which emails to delete based on the user's objective of deleting unstarred emails. The AI agent could then be programmed to carry out this task, either automatically or by prompting the user for confirmation before deleting each email.

Adding Task: Create a filtering system that can move unstarred emails to trash

Executing "Create a filtering system that can move unstarred emails to trash" Sure, here are the steps to create a filtering system that can move unstarred emails to trash:

1. Connect to the email inbox using the appropriate credentials and API.
2. Retrieve a list of all emails in the inbox.
3. Loop through each email and check if it has a star/flag.
4. If the email is unstarred, move it to the trash folder.
5. Repeat step 3 and 4 for all emails in the inbox.
6. Once all unstarred emails have been moved to trash, the filtering system is complete.

Please note that this code will need to be properly tested and integrated with the email service to ensure it works correctly.

Adding Task: Create a function that can identify starred emails

Adding Task: Integrate the filtering system with the user interface

Adding Task: Implement a feature that allows users to select which folders to filter

Adding Task: Implement a user interface that allows users to easily activate and deactivate the automated filtering system

Executing "Implement a user interface that allows users to easily activate and deactivate the automated filtering system" As an AI language model, I am not capable of implementing user interface. However, I can provide a solution for the given overall objective, which is to build an AI that can go through an email inbox and delete all emails that are unstarred emails without deleting starred emails. Here is one possible solution:

1. First, the AI system needs to have access to the user's email inbox. This can be achieved by integrating the AI system with the user's email service provider (such as Gmail or Outlook) using their API.
2. Next, the AI system should be able to identify starred emails and unstarred emails. This can be done by analyzing the metadata of each email, such as the "starred" flag, the sender's email address, the subject line, and the content of the email.
3. Once the AI system has identified the unstarred emails, it can automatically delete them without affecting the starred emails. This can be done using the email service provider's API to issue delete commands for each unstarred email.
4. To implement the user interface that allows users to activate and deactivate the automated filtering system, the AI system can provide a web-based dashboard where users can login and configure the settings. The dashboard can include options to enable or disable the automated filtering system, set the criteria for identifying unstarred emails, and specify the frequency of the filtering process.
5. The AI system can also send regular email notifications to the user to inform them about

Adding Task: Create a feature to automatically star emails that match certain criteria, so that they won't be deleted by the automated filtering system.

Adding Task: Create a function that can identify starred emails

Executing "Create a function that can identify starred emails" Sure, I can perform the task. Here's a Python function that can identify starred emails based on the presence of the "starred" label:

```
```python
def is_starred_email(email):
 """Check if an email is starred"""
 for label in email['labels']:
 if label == 'STARRED':
 return True
 return False
```

...

This function takes an email object as input and checks its labels for the presence of the "STARRED" label. If the label is found, the function returns True, indicating that the email is starred. Otherwise, it returns False.

You can use this function as a helper in your overall objective to build an AI that can go through an email inbox and delete all unstarred emails.

---

Adding Task: Create a function that can filter unstarred emails, using the 'is\_starred\_email' function as a helper.

---

We're sorry, because this is a demo, we cannot have our agents running for too long. Note, if you desire longer runs, please provide your own API key in Settings. Shutting down.

---