AI Newsroom Tool: Automated FOIL Response Organizer

Cullen Fagan

The Problem

FOIL Requests are one of our really exciting tools as journalists – it's how we get a ton of vital information from the government **BUT** they're the worst! They take forever and are hard to keep track of.

Use Case: I am a reporter who is wants to cast a wide net in terms of FOILs – I'm talking filing multiple a day, just in case something sticks! But I don't want to clog my email with responses AND I anticipate most of them won't pan out.

The Solution

A dummy email account and an automated FOIL response categorizer!

How does the tool work?

The code is pretty straightforward: It uses imaplib to log into the account, read the bodies of new emails, and then creates a model based on training data that sorts the emails into types of responses. It then produces a data frame of these responses.

```python
        if content_type == "text/plain":
            return part.get_payload(decode=True).decode()
        elif content_type == "text/html":
            return part.get_payload(decode=True).decode()
    else:
        return msg.get_payload(decode=True).decode()
```

```python
In [130…   results = []

           for email_id in email_ids:
               status, msg_data = mail.fetch(email_id, "(RFC822)")

               for response_part in msg_data:
                   if isinstance(response_part, tuple):
                       msg = email.message_from_bytes(response_part[1])
                       body = extract_email_body(msg)

                       label = classify_email(body)

                       results.append((body, label))
```

```python
In [131…   df = pd.DataFrame(results, columns=["email_content", "predicted_label"])
           df
```

Out[131…

	email_content	predicted_label
0	Please see the attached documents regarding yo…	records

Training Data

Content from real FOIA response letters, manually sorted into 50 "extensions", "denials", "records", and "other".

I sourced the training from muckrock.com, where they have a HUGE database of FOIA requests and responses. I used 200 responses for the training set.

Demo

Next Steps

For Records

- Download pdf attachments through imaplib
- Forward them to my email.

For Extensions

- Use Regex to pull the date indicated in the extension notification.
- Put dates into df
- Reminder on date?

For Denials

- Put all denials in a spreadsheet with their content and forward to me in an email to review for appeals.

Other?

For emails that can't be sorted into the above categories, we can either dismiss them by getting really good at IDing the first three OR do what we probably should do, which is break them down further – categories such as "requests for payment" and "receipt."

What libraries would I need for the next steps?

- re
- smtplib
- datetime, probably

Limitations + Improvements

Missing Categories

Problem

There are way more kinds of responses than the ones I've indicated – the 'other' category is way too broad which is confusing the data and making it hard to identify the other categories.

Solution

More categories in the training data.

Better Indicators

Problem

Right now, this tells me broadly about what's happening in my inbox, and can point me to check if there are new records, etc. But it's not super specific if I want to check individual FOIAs in the spreadsheet, and it's not historical – it only checks new emails.

Solution

Regex to pull FOIA ID, sender, other identifiers.

Accuracy

Problem

The model isn't very accurate right now.

Solution

It needs WAY more training data.

Thank you!