Email Subject Line Generation - Capstone Project

Data : Cleaning, PreProcessing and Exploratory Data Analysis

Annotated Enron Subject Line Corpus (AESLC)

- The first dataset for email subject line generation (SLG).
- Derived from the Enron Corpus
- It addresses the unique challenges of email subject generation, such as
  - high abstraction and
  - compression ratios.
- This dataset enables research into highly abstractive summarization tasks beyond email subjects, such as **document section title generation**

AESLC Dataset Splits: Training, Validation (Dev) & Test

The Annotated Enron Subject Line Corpus (AESLC) is divided into three datasets: **Training, Validation (Dev), and Test**.

1. **Training Dataset:**
   - **Size**: 10,570 emails.
   - **Purpose**: Used to train the model to learn patterns and relationships between email bodies and their corresponding subject lines.
   - **Features**: Emails are preprocessed to remove boilerplate content, filtered to exclude short or redundant emails, and deduplicated to prevent data leakage.
2. **Validation (Dev) Dataset:**
   - **Size**: 1,960 emails.
   - **Purpose**: Used to evaluate the model's performance during training, helping to tune hyperparameters and prevent overfitting.
   - **Features**: Similar preprocessing as the training set, with each email annotated with three human-written subject lines to account for variability.
3. **Test Dataset:**
   - **Size**: 1,906 emails.
   - **Purpose**: Used to evaluate the final performance of the trained model, providing an unbiased assessment of its ability to generate effective subject lines.
   - **Features**: Also preprocessed and annotated with three human-written subject lines per email, allowing for comprehensive evaluation against human standards.

- These datasets are derived from the Enron Corpus, a collection of emails from Enron employees, and are specifically designed for the task of email subject line generation. The AESLC dataset is notable for its focus on business/personal emails, requiring high abstraction and compression ratios compared to news summarization datasets

AESLC / enron_subject_line / test / arora-h_inbox_3.subject

Rui Zhang first commit

Code Blame 17 lines (13 loc) · 396 Bytes

```
1   Harry - I got kicked out of the system, so I'm sending this from Tom's account.
2   He can fill you in on the potential deal with STEAG.
3   I left my resume on your chair.
4   I'll e-mail a copy when I have my home account running.
5   My contact info is:
6
7   @subject
8   STEAG Power LLC
9
10  @ann0
11  steag potential deal/my resume
12
13  @ann1
14  follow-up on steag deal and resume
15
16  @ann2
17  information regarding potential steag deal
```

AESLC / enron_subject_line / test / arnold-j_inbox_153.subject

Rui Zhang first commit

Code Blame 15 lines (11 loc) · 456 Bytes

```
1   As our last day is Friday, November 30th, we would love to toast the good times and special memories that we have shared with you over the past five years.
2   Please join us at Teala's (W. Dallas) on Thursday, November 29th, beginning at 5pm.
3   Looking forward to being with you,   Lara and Janel     Lara Leibman
4
5   @subject
6   Farewell Drinks
7
8   @ann0
9   our last day
10
11  @ann1
12  join us at teala's on thursday, november 29th at 5pm!
13
14  @ann2
15  getting together on our final day
```

AESLC / enron_subject_line / train / allen-p_sent_251.subject

Rui Zhang first commit

Code Blame 7 lines (6 loc) · 383 Bytes

```
1   Will,   Here is a list of the top items we need to work on to improve the position  and p&l reporting for the west desk.
2   My underlying goal is to create position managers and p&l reports that  represent all the risk held by the desk and estimate p&l with great accuracy.
3   Let's try and schedule a meeting for this Wednesday to go over the items  above.
4   Phillip
5
6   @subject
7   Priority List
```

Dev and Test Files: Contain Email Body, `@subject`, Subject Line, `@ann0`, Annotation 0, `@ann1`, Annotation 1, `@ann2`, Annotation 2.

Training Files: Contain Email Body, `@subject`, Subject Line.

Dataset Creation

The AESLC dataset contains 14,436 emails split into training, validation, and test sets.

- Training: 10,570 emails.
- Validation: 1,960 emails.
- Test: 1,906 emails

Key characteristics:

- Domain: Business/personal emails from the Enron Corpus.
- Average body length: 75 words.
- Average subject length: 4 words (shorter than news headlines)
- Compression ratio: ~18:1 (75 words → 4 words), requiring extreme abstraction

Feature	AESLC
Domain	Business/Personal
Avg. Body Words	75
Avg. Subject Words	4

Email subjects require higher compression ratios (e.g., compressing 75 words to 4) necessitating **extreme abstraction.**

Email subjects rely on implicit sender-recipient context (e.g., "Meeting" assumes prior knowledge of the event)

Subject Annotation

- For the dev/test sets, each email is annotated with **3 human-written subject lines** via Amazon Mechanical Turk to account for *variability*.
- Quality control measures include:
  - Rejecting vague/generic subjects (e.g., "Update" or "Request").
  - Ensuring subjects contain body-specific keywords e.g.,
    - "Current Job Description Needed" instead of "Job Description“,
    - "Revised Budget Proposal" instead of "Proposal".
  - Analyzing inter-annotator agreement using ROUGE-L F1 scores (~34 between human annotations)

Preprocessing Steps

1. Boilerplate Removal:
   Automatically appended content (e.g., disclaimers, advertisements, attachments) is stripped to retain only the author-written body.
2. Filtering Criteria:
   - Emails with fewer than 3 sentences or 25 words are excluded to ensure meaningful content.
   - Replies/forwards (subjects starting with "RE:" or "FW:") are removed to avoid redundancy.
3. Deduplication:
   Identical emails sent to multiple recipients are removed to prevent train-test overlap

Key Challenges

- **Shared Context**: Email subjects often rely on implicit sender-recipient knowledge (e.g., "Request" assumes prior context; "Project Update" assumes familiarity with the project, "Meeting" assumes prior knowledge of the event)), complicating generation.
- **Subject Variability**: Multiple valid subjects exist for the same email, necessitating diverse annotations for evaluation
- This preprocessing pipeline enabled the creation of a benchmark dataset for highly abstractive summarization tasks, advancing research in email subject generation and related domains

Model Architecture

The proposed model uses a two-stage extractor-abstractor framework:

- Multi-Sentence Extractor: A pointer network selects salient sentences from the email body using hierarchical sentence representations and a bidirectional LSTM.
- Multi-Sentence Abstractor: A sequence-to-sequence model with copy mechanisms rewrites the extracted sentences into a subject line. Training involves supervised pretraining followed by reinforcement learning (RL) optimized using a custom **Email Subject Quality Estimator (ESQE)** as the reward function

Evaluation Metrics

- The ESQE metric, a regression-based neural network, predicts subject quality scores by combining CNN-encoded representations of the email body and subject. It achieves a **Pearson correlation of 0.64 with human judgments**.
- Automatic metrics (ROUGE, METEOR, BLEU) and human evaluations show the model **outperforms baselines**, approaching human-level performance

Results and Insights

- This preprocessing pipeline enabled the creation of a benchmark dataset for highly abstractive summarization tasks, advancing research in email subject generation and related domains
- Human Evaluation: Subjects generated by the model received an **average score of 3.65/4.0**, close to human-written subjects (3.82/4.0).
- Automatic Metrics: The model outperforms extractive and abstractive baselines, with improvements of up to **4.3 ROUGE-L points**.
- Challenges: Email subjects often rely on shared context between sender and recipient, complicating generation.
- For example, a subject like "Request" is too vague without body-specific details

Applications and Future Work

- The authors suggest applications in email triaging and document section title generation.
- Future directions include incorporating sender/recipient metadata and exploring few-shot learning for low-resource scenarios

Exploratory Data Analysis

1. Data Cleaning

All Datasets

- **Text Decoding:** Handle character encoding issues.
- **Lowercasing**: Convert all text to lowercase.
- **Punctuation Removal**: Remove punctuation.
- **Special Character Handling**: Remove or replace special characters.
- **Whitespace Handling**: Remove extra whitespace.

Training Dataset

- **Missing Data:** Check and handle missing email bodies or subject lines.
- **Boilerplate Removal:** Use regex or string matching to remove common email footers or signatures.
- **Length Filtering:** Remove very short emails (e.g., < 25 words).

Dev and Test Datasets

- Missing Data: Same as training.
- Boilerplate Removal: Same as training.
- Length Filtering: Same as training.
- **Ensure Annotation Consistency:** check for the quality and relevancy of the annotation

2. Data Preprocessing

All Datasets

- **Tokenization**: Split the text into individual words (tokens).
- **Stop Word Removal**: Remove common words (e.g., "the," "a," "is").
- **Stemming/Lemmatization:** Reduce words to their base form.
- **Vectorization:** Convert the text data into numerical vectors.
  - *TF-IDF* (Term Frequency-Inverse Document Frequency): Weight words by importance.
  - *Word Embeddings* (Word2Vec, GloVe, BERT embeddings): Capture semantic meaning.

3. Data Analysis

Training Dataset

- **Email Length Distribution:** Analyze the distribution of email body lengths to understand the dataset's characteristics.
- **Vocabulary Analysis:** Identify the most frequent words in the email bodies and subject lines to gain insights into common topics.

Dev and Test Datasets

- **Annotation Analysis:** Check the distribution and variability of annotations in these datasets. Check the distribution of key words, semantic similarity, and subject relevancy with mail body

4. Importance of Training & Test Dataset

To generate subject lines for a given email, you should train your model using the training dataset and evaluate it using the test dataset.

Therefore, your file reading process should consider this split:

- **Training Phase:** Read and process all files from the **training directory** to train your subject line generation model.
- **Testing Phase:** When you want to generate a subject line for a new email (or evaluate your model), you'll read and process the relevant files from the **test directory** (or a separate set of unseen emails).

5. Importance of Dev Dataset

The dev dataset is used for **validation**. During the training process, you'll use the dev dataset to:

- **Tune Hyperparameters**: Adjust settings like learning rate, batch size, etc., to optimize model performance.
- **Monitor Overfitting**: Check if the model is starting to memorize the training data instead of learning general patterns. Performance on the dev set will degrade if overfitting occurs.

In essence, the dev dataset acts as a "practice test" for your model during training, allowing you to make adjustments and improvements before the final evaluation on the test data.

6. Use of each dataset in email subject line generation project

1.Training Dataset:

1. Purpose: To train your machine learning model.
2. Action: The model learns the relationship between email bodies and their corresponding subject lines.

2.Dev (Validation) Dataset:

1. Purpose: To evaluate the model's performance during training and fine-tune hyperparameters.
2. Action: Use it to prevent overfitting and adjust model settings.

3.Test Dataset:

1. Purpose: To assess the final performance of your trained model.
2. Action: Provides an unbiased evaluation of the model's ability to generate subject lines for unseen emails.

Overall Goals of EDA

- Understand the characteristics of the email data.
- Identify patterns and relationships between email bodies and subject lines.
- Uncover potential issues or biases in the data.
- Guide feature engineering and model selection.

EDA - Training Dataset

The goal is to learn the relationship between the mail and its subject.

1. Basic Statistics:
    1. Number of files
2. Email Body Analysis:
    1. Average Length.
    2. Distribution of Lengths.
    3. Most Frequent Words.
    4. Top N-grams.
3. Subject Line Analysis:
    1. Average Length
    2. Distribution of Lengths.
    3. Most Frequent Words.
    4. Top N-grams.
4. Email Body and Subject Line Relationships:
    1. Correlation between email body length and subject line length.
    2. Common words and phrases in both email bodies and subject lines.
    3. Examples of email bodies and their corresponding subject lines.
    4. Check to see Subject contains mail key words

Dev and Test Datasets

The goal here is to check if the data suits the requirements and the quality of annotations

1. Basic Statistics:
    1. Number of files.
2. Email Body Analysis:
    1. Average Length.
    2. Distribution of Lengths.
3. Subject Line Analysis:
    1. Average Length.
    2. Distribution of Lengths.
4. Annotation Analysis:
    1. Annotation Lengths Distribution
    2. Vocabulary Overlap Between Annotations
    3. Annotation Diversity: Are annotations similar or do they capture different aspects? Calculate pair-wise similarity scores (e.g., cosine similarity of TF-IDF vectors) between annotations for the same email.
5. Email Body vs. Subject Line/Annotations:
    * **Relevance of Subject Lines and Annotations:** Check how much overlap in keywords there is between the email body, the generated subject line, and the provided annotations
6. Human Performance Baseline:
    * Use the multiple human annotations in the dev/test sets to establish a "human performance" baseline. Compare the automatic metrics (ROUGE, etc.) between the annotations to get an estimate of human-level performance.

Implementation Notes

- **Libraries:** Use libraries like nltk, scikit-learn, matplotlib, and seaborn for text processing, analysis, and visualization.
- **Iterate:** Read data in chunks if the dataset is large to avoid memory issues.
- **Visualize:** Use histograms, box plots, word clouds, and other visualizations to explore data characteristics.