Annotations, Evaluation, and Generalization

CS 485, Fall 2023
Applications of Natural Language Processing
https://people.cs.umass.edu/~brenocon/cs485_f23/

Brendan O'Connor
College of Information and Computer Sciences
University of Massachusetts Amherst
• Annotations
  • Chance-adjusted calculation
  • Practical ethics example in ChatGPT

• Evaluation
  • Held-out data and overfitting
  • Classification metrics
  • Statistical testing (J&M 4.9) - hold off until later in course
Cohen's Kappa for IAA

- If some classes predominate, raw agreement rate may be misleading
- Idea: normalize accuracy (agreement) rate such that answering randomly = 0.
  - From psychology / psychometrics / content analysis
- **Chance-adjusted agreement:**
  \[ p_o: \text{observed agreement rate} \]
  \[ p_e: \text{expected (by chance) rate} \]

Other chanced-adjusted metrics: Fleiss, Krippendorff... see reading
When is annotating ethical?
Human labeling is key to ChatGPT

Data generation process

Society (SocialAttributes)

Writing (TextGenerator)

Text Data (Text)

First training phase:
Maximize probability of texts in corpus

Human labeling of text outputs

Fine-tuned LM parameters

Second training phase:
Maximize expectation of human-provided quality ratings

LM parameters

[Ouyang et al., 2022, Taori et al. 2023]
Table 3: Labeler-collected metadata on the API distribution.

<table>
<thead>
<tr>
<th>Metadata</th>
<th>Scale</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overall quality</td>
<td>Likert scale; 1-7</td>
</tr>
<tr>
<td>Fails to follow the correct instruction / task</td>
<td>Binary</td>
</tr>
<tr>
<td>Inappropriate for customer assistant</td>
<td>Binary</td>
</tr>
<tr>
<td>Hallucination</td>
<td>Binary</td>
</tr>
<tr>
<td>Satisfies constraint provided in the instruction</td>
<td>Binary</td>
</tr>
<tr>
<td>Contains sexual content</td>
<td>Binary</td>
</tr>
<tr>
<td>Contains violent content</td>
<td>Binary</td>
</tr>
<tr>
<td>Encourages or fails to discourage violence/abuse/terrorism/self-harm</td>
<td>Binary</td>
</tr>
<tr>
<td>Denigrates a protected class</td>
<td>Binary</td>
</tr>
<tr>
<td>Gives harmful advice</td>
<td>Binary</td>
</tr>
<tr>
<td>Expresses opinion</td>
<td>Binary</td>
</tr>
<tr>
<td>Expresses moral judgment</td>
<td>Binary</td>
</tr>
</tbody>
</table>
'That Was Torture;' OpenAI Reportedly Relied on Low-Paid Kenyan Laborers to Sift Through Horrific Content to Make ChatGPT Palatable

The laborers reportedly looked through graphic accounts of child sexual abuse, murder, torture, suicide, and incest.

By Mack DeGurin  Published January 18, 2023 | Comments (6) | Alerts

Image: Ascannio (Shutterstock)
Held-out data for evaluation

- How well will my classifier work in the future?
- Can we look at classifier accuracy on training data?
Held-out data for evaluation

• Need to diagnose how much your model is **overfitting** the training set
• Data splits are key. Some ways to split:
  • Training set  -vs-  test set
  • Training set  -vs-  "validation"/"development" set  -vs-  test set
  • Cross-validation (within training set)  -vs-  test set
Cross-validation

• Cross-validation (within training set) -vs- test set

![10-fold cross-validation diagram](image)

Figure 4.7 10-fold cross-validation
Regularization in Naive Bayes
Regularization in logistic regression

• If "dog" only occurs for class $k$, what weight will it get?
• Consider MLE training:

• Solution: regularized training for logistic regression
Regularization tradeoffs

• No regularization  <-------------  Very strong regularization
Do I have enough labels?

- For training, hundreds to thousands of annotations may be needed for reasonable performance.
- Current work: how to usefully make NLP models with <10 or <100 training examples. "Few-shot learning"
- Exact amounts are difficult to know in advance. Can do a learning curve to estimate if more annotations will be useful.
Evaluation metrics

- Accuracy =

![Confusion Matrix]

- But do we care about false positives and negatives equally?
- What about rare classes?
Precision, recall, F1

To introduce the methods for evaluating text classification, let's first consider some simple binary detection tasks. For example, in spam detection, our goal is to label every text as being in the spam category (“positive”) or not in the spam category (“negative”). For each item (email document) we therefore need to know whether our system called it spam or not. We also need to know whether the email is actually spam or not, i.e. the human-defined labels for each document that we are trying to match. We will refer to these human labels as the gold labels.

Or imagine you’re the CEO of the Delicious Pie Company and you need to know what people are saying about your pies on social media, so you build a system that detects tweets concerning Delicious Pie. Here the positive class is tweets about Delicious Pie and the negative class is all other tweets.

In both cases, we need a metric for knowing how well our spam detector (or pie-tweet-detector) is doing. To evaluate any system for detecting things, we start by building a confusion matrix like the one shown in Fig. 4.4. A confusion matrix is a table for visualizing how an algorithm performs with respect to the human gold labels, using two dimensions (system output and gold labels), and each cell labeling a set of possible outcomes. In the spam detection case, for example, true positives are documents that are indeed spam (indicated by human-created gold labels) that our system correctly said were spam. False negatives are documents that are indeed spam but our system incorrectly labeled as non-spam.

To the bottom right of the table is the equation for accuracy, which asks what percentage of all the observations (for the spam or pie examples that means all emails or tweets) our system labeled correctly. Although accuracy might seem a natural metric, we generally don’t use it for text classification tasks. That’s because accuracy doesn’t work well when the classes are unbalanced (as indeed they are with spam, which is a large majority of email, or with tweets, which are mainly not about pie).

### Precision, recall, F1

- **Precision**
  
  \[
  \text{precision} = \frac{\text{true positive}}{\text{true positive} + \text{false positive}}
  \]

- **Recall**
  
  \[
  \text{recall} = \frac{\text{true positive}}{\text{true positive} + \text{false negative}}
  \]

- **Accuracy**
  
  \[
  \text{accuracy} = \frac{\text{true positive} + \text{true negative}}{\text{true positive} + \text{false positive} + \text{true negative} + \text{false negative}}
  \]

- **F1 score**

  \[
  \text{F1 score} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}
  \]

- **Macro vs. Micro F1**