

Adapting a Statistical E-mail Filter

David Kohlbrenner
TJHSST
IT.com

Training

- All filters begin blank
 - Token database is empty, as is messages
- Trained with a corpus of spam / nonspam
 - Specified by user, this is supervised learning

•Methods for training as email is seen

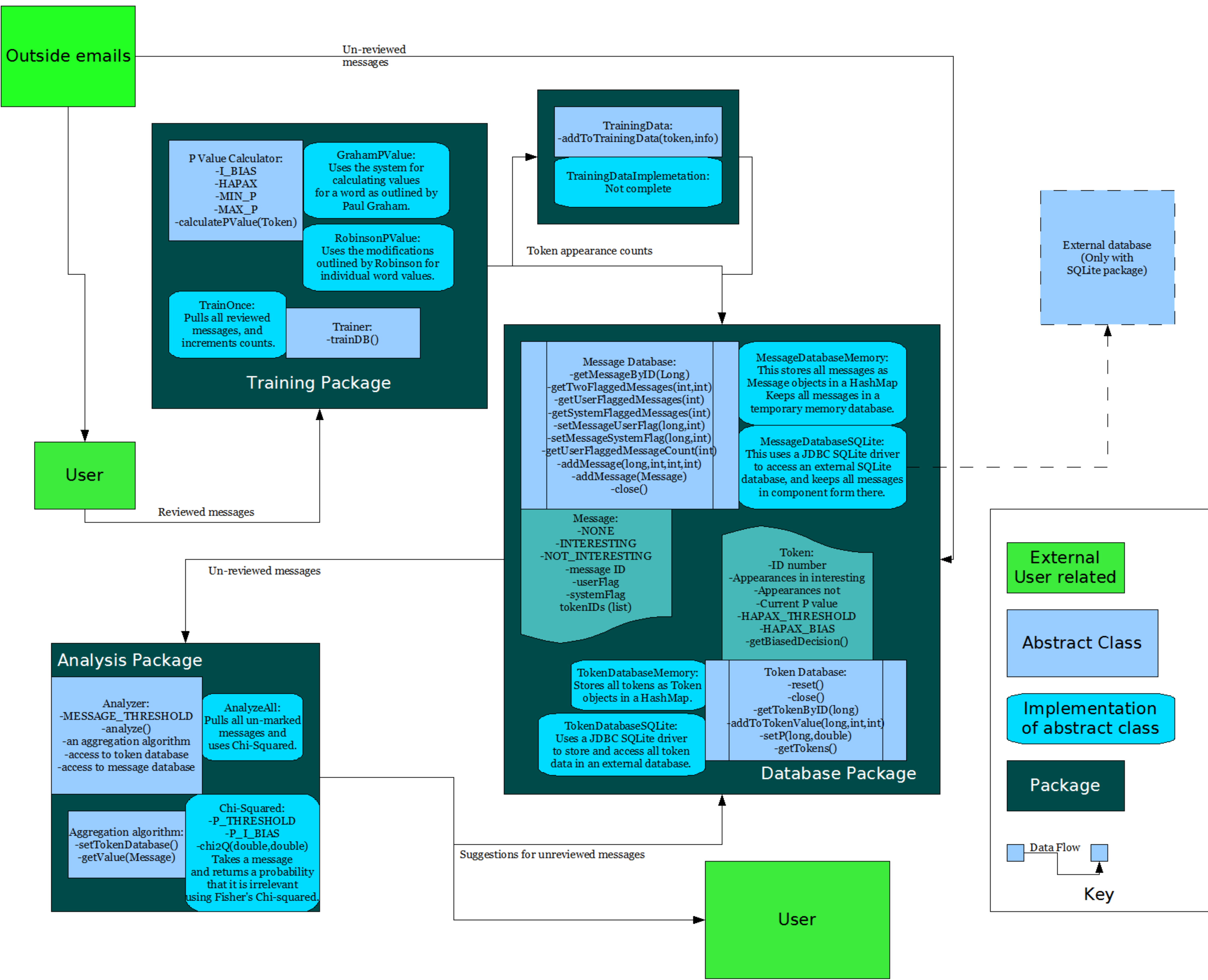
- TEFT
 - Train every email the user marks
- TOE
 - Train whenever the prediction does not match user classification.
- TUNE
 - TOE, but retrain the message until it is correctly categorized.
- This filter currently has a TEFT module

Analysis

- Email's tokens are compared to training data
- Some aggregated percentage is created for email
 - This can be done with one of two algorithms
- Paul Graham's
 - $P = \frac{((AS)/(TS))}{(((AS)/(TS)) + ((AI)/(TI)))}$
 - where AS and AI are the total appearances in spam and innocent email, and TS and TI are the total number of spam and innocent emails in the corpus.
- Gary Robinson's
 - $F = \frac{(S \cdot X + N \cdot P())}{(S + N)}$
 - where S is a tuning variable, N is the total appearances of the token, P is Graham's value for the token, and X is the hapax value
- This filter uses Robinson's

Databases

- Two databases
 - Token Database
 - Holds all information about all tokens
 - Each token is a word, phrase, HTML tag, or more
 - Database holds appearance counts.
 - Message Database
 - Holds all messages, and their classifications
 - Both user and system (or guess) classifications
- Each message is an email.



Results

- Works to an extent
 - Test data had very limited feature set
 - Test data was based on personal writing style
 - Little time to test/tune
- 56%-57% accuracy at best
 - Measured by interesting predicted/interesting actual
 - Also mistakes/interesting marked
- More testing will be done
 - With current data, no conclusions can be drawn about this filter.

Why this design?

- Highly modular parts
 - Databases, analyzing methods, and training methods are easily swappable.

- Corpus will be static set, with just the categorized messages changing and increasing.

Tokenization, or Feature Set Creation

- Though this is part of pre-processing, it is critical to the functionality of this filter
- Tokenization is the process of turning an email into a list of parts, or tokens.
- A token can be a word, a phrase, or even HTML and header features.
- Example:
 - “The orange ball” Becomes
 - “The”
 - “orange”
 - “ball”
 - “The orange”
 - “orange ball”
 - “The orange ball”
 - “*FONT Times New Roman”
 - “*FONTSIZE 30 pt”
 - etc...