# Improving Spam Detection Based on Structural Similarity

Luiz H. Gomes, Fernando D. O. Castro,
Virǵ ılio A. F.A lmeida, Jussara M. Almeida, Rodrigo B. Almeida, Luis M.A. Bettencourt

Steps to Reducing Unwanted Traffic on the Internet Workshop, 2005

Presented By: Dan DeBlasio, Spring 2008
5 March 2008

# Outline

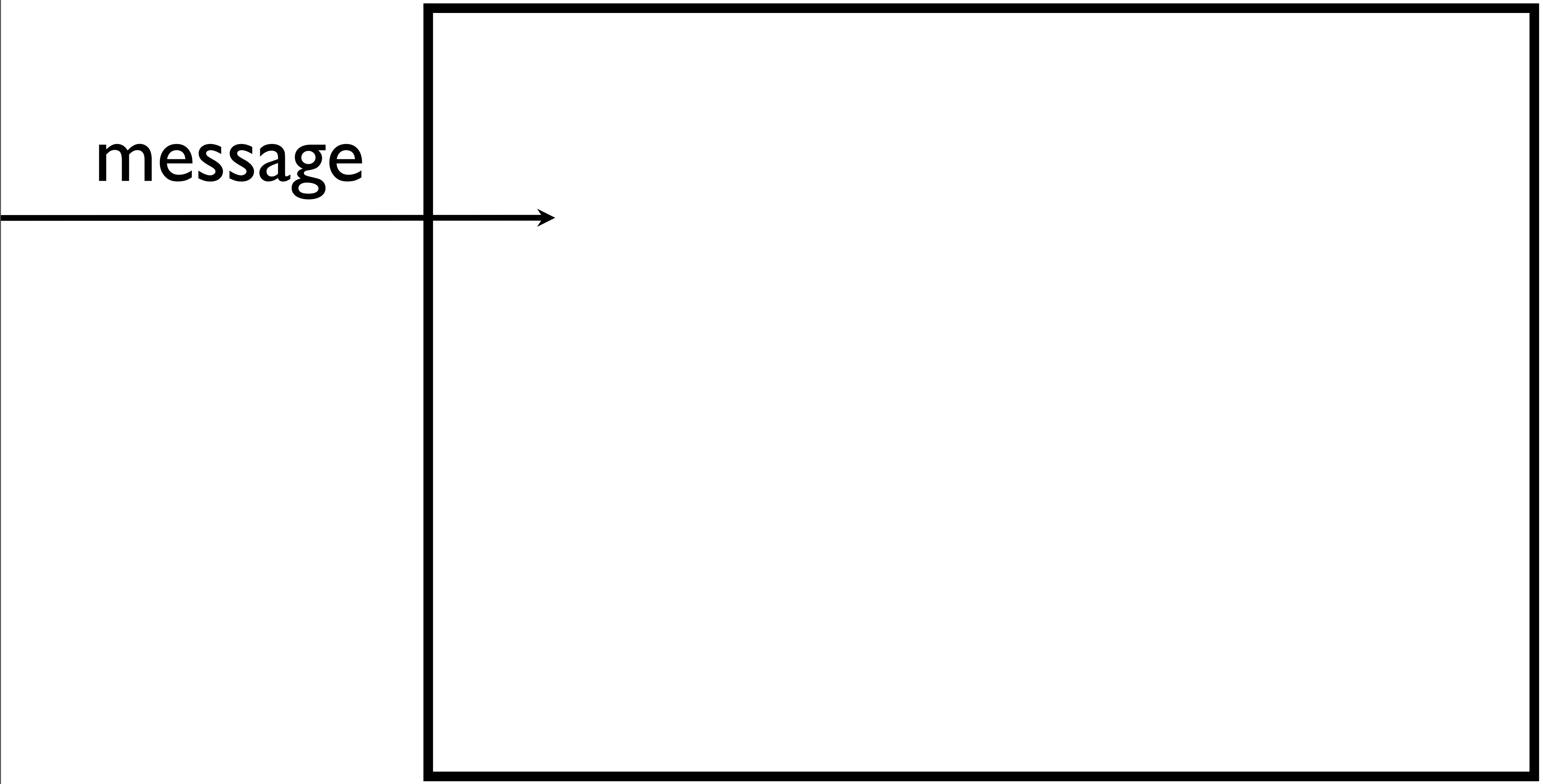
- System Overview
- Algorithm Description
- System Example
- Results
- Conclusion

# Overview

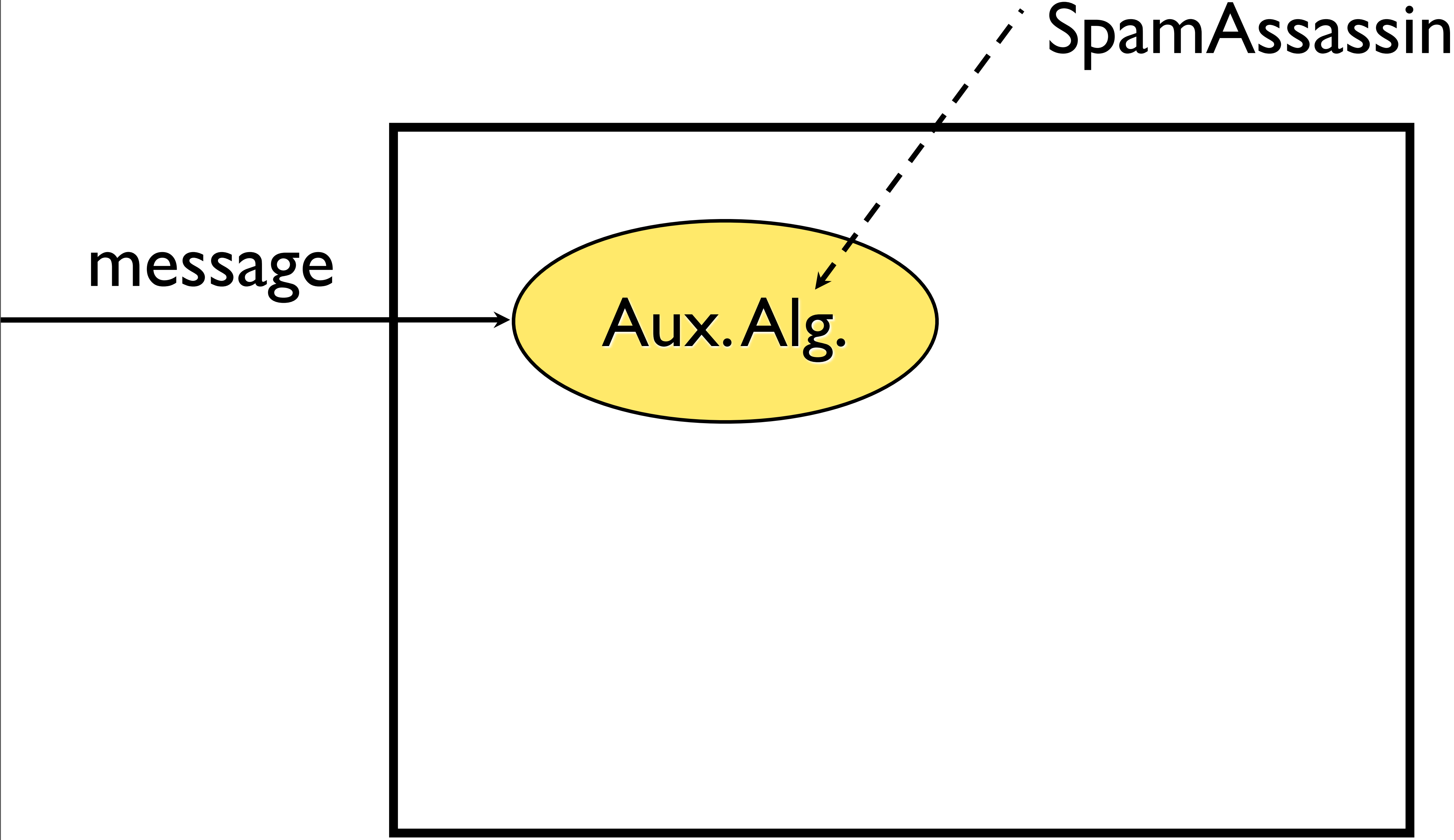
- expand on the results of other spam detection
- helps to reduce false positives
- uses statistical analysis on sender and receiver “contact list”

# Proposed Architecture

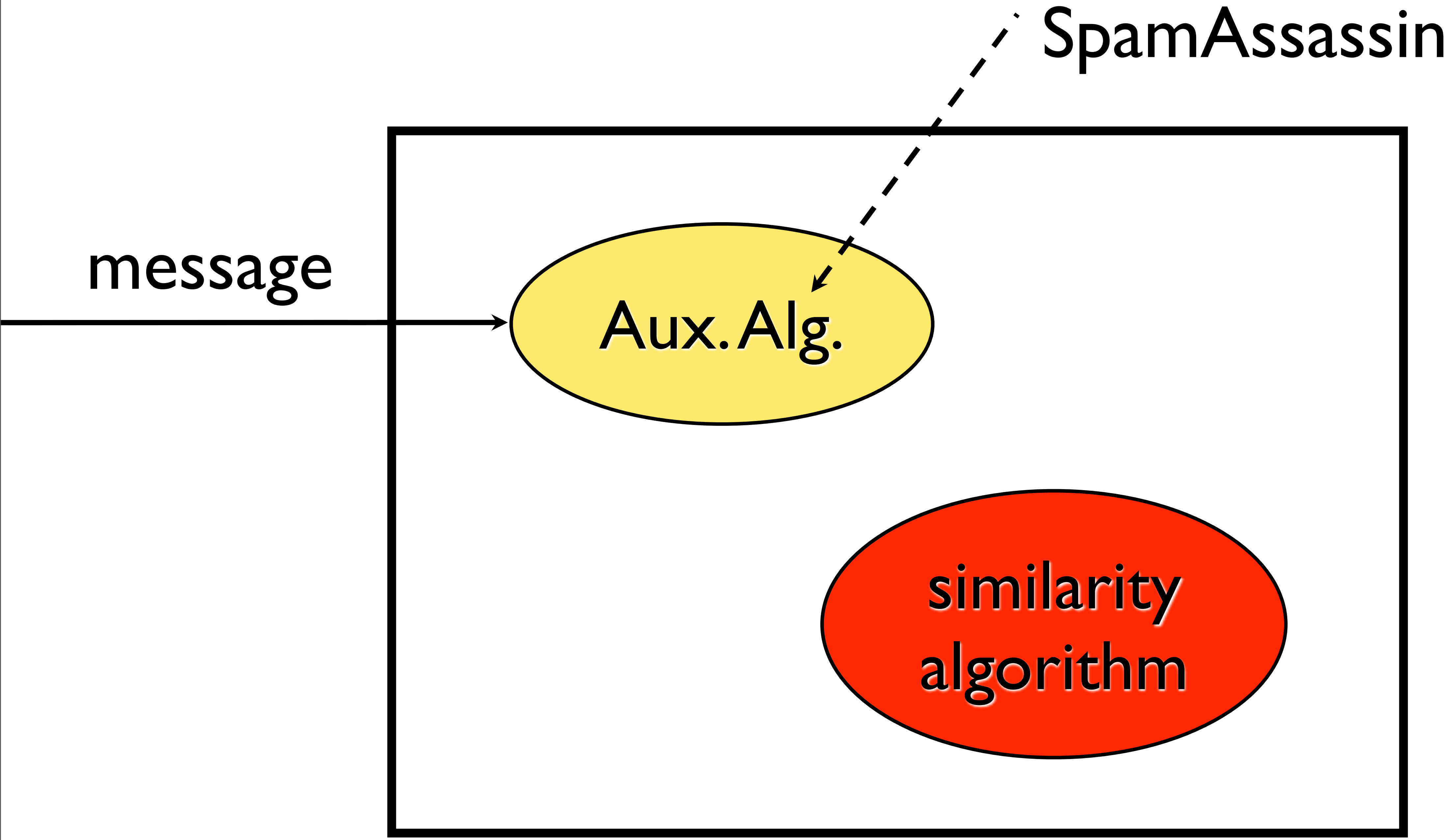
# Proposed Architecture



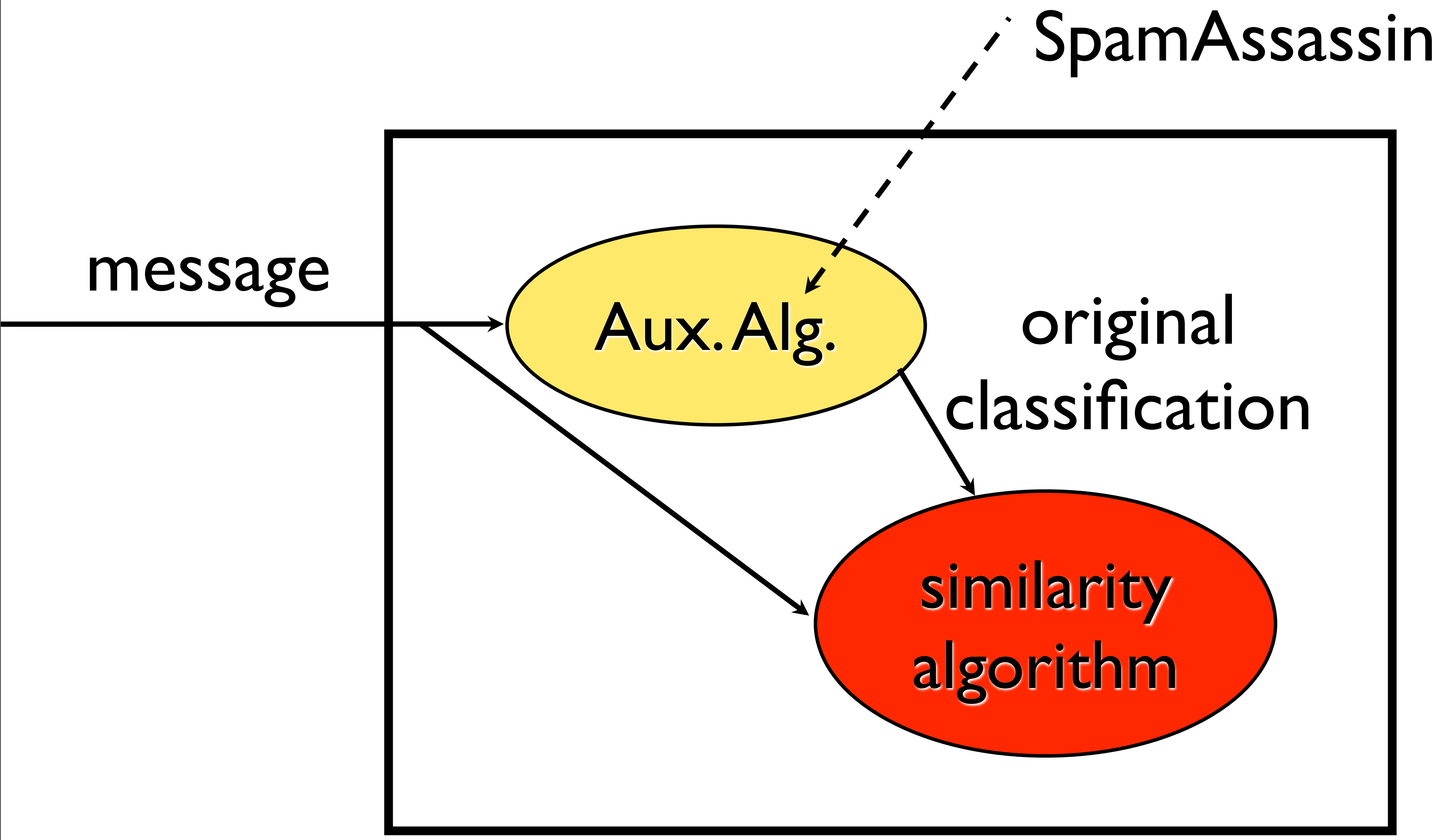
# Proposed Architecture



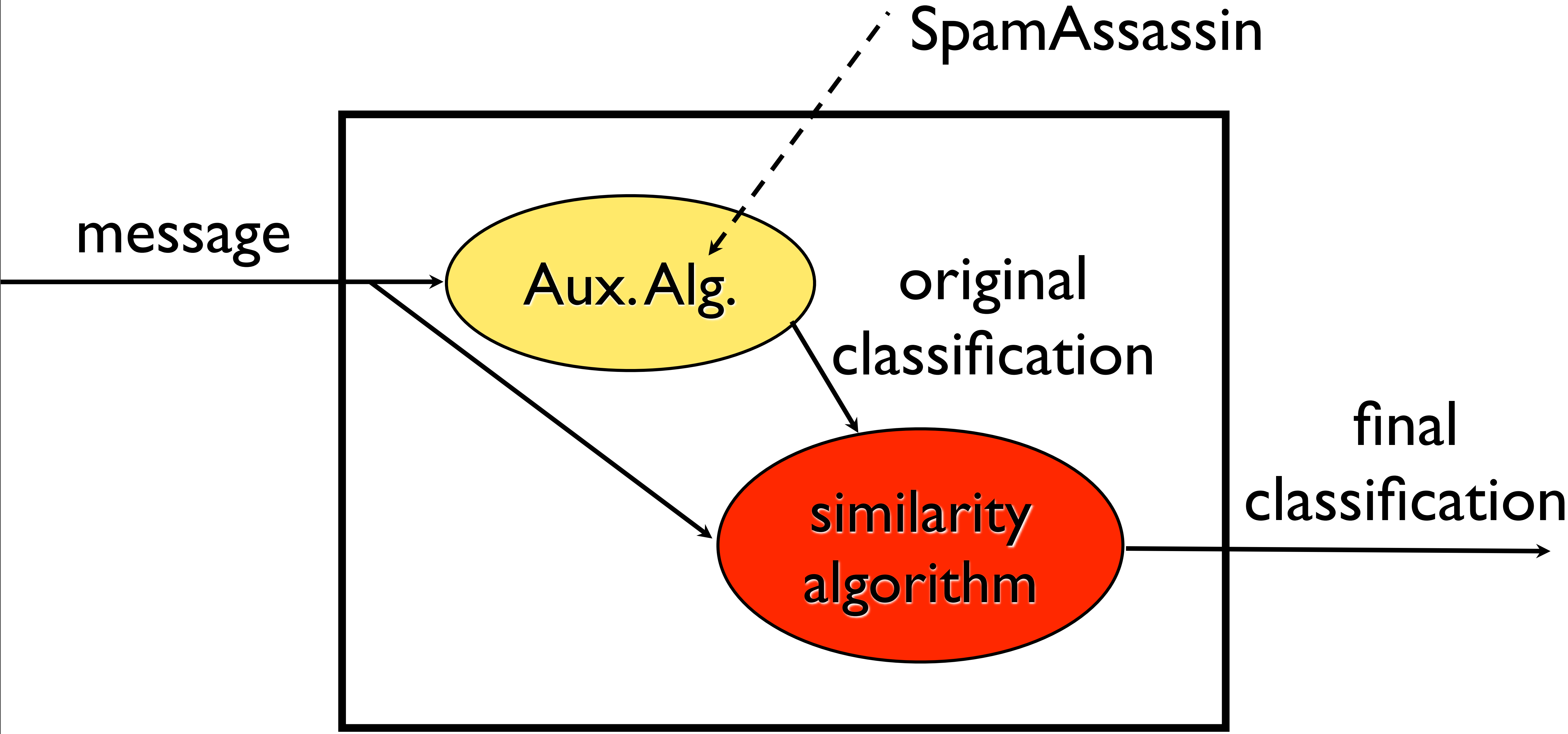
# Proposed Architecture



# Proposed Architecture

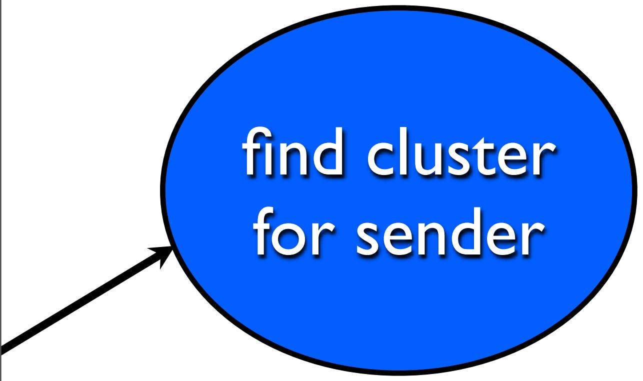


# Proposed Architecture

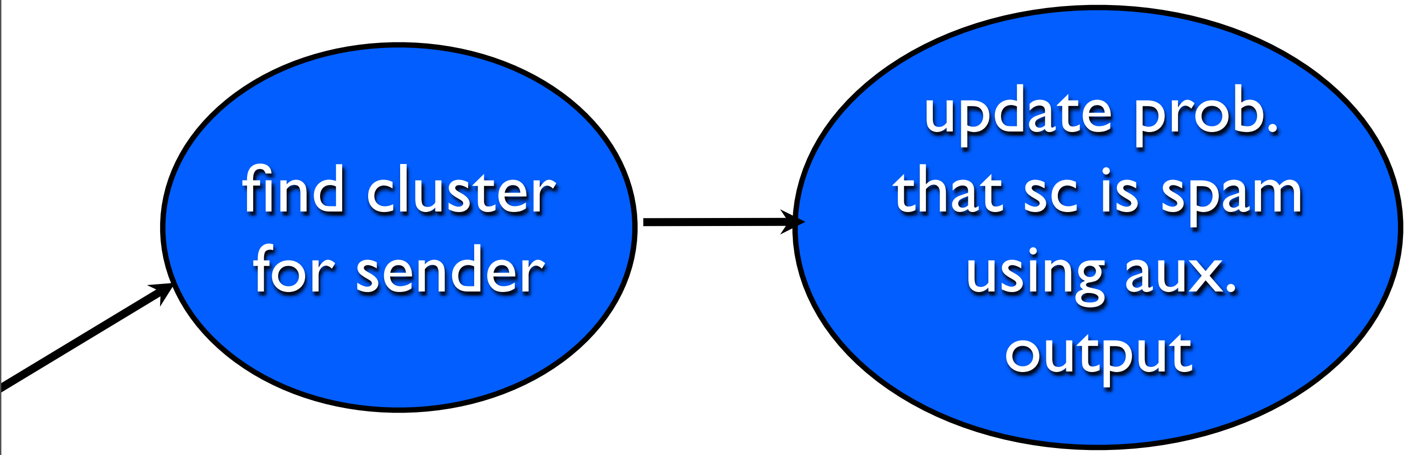


# Basic Algorithm

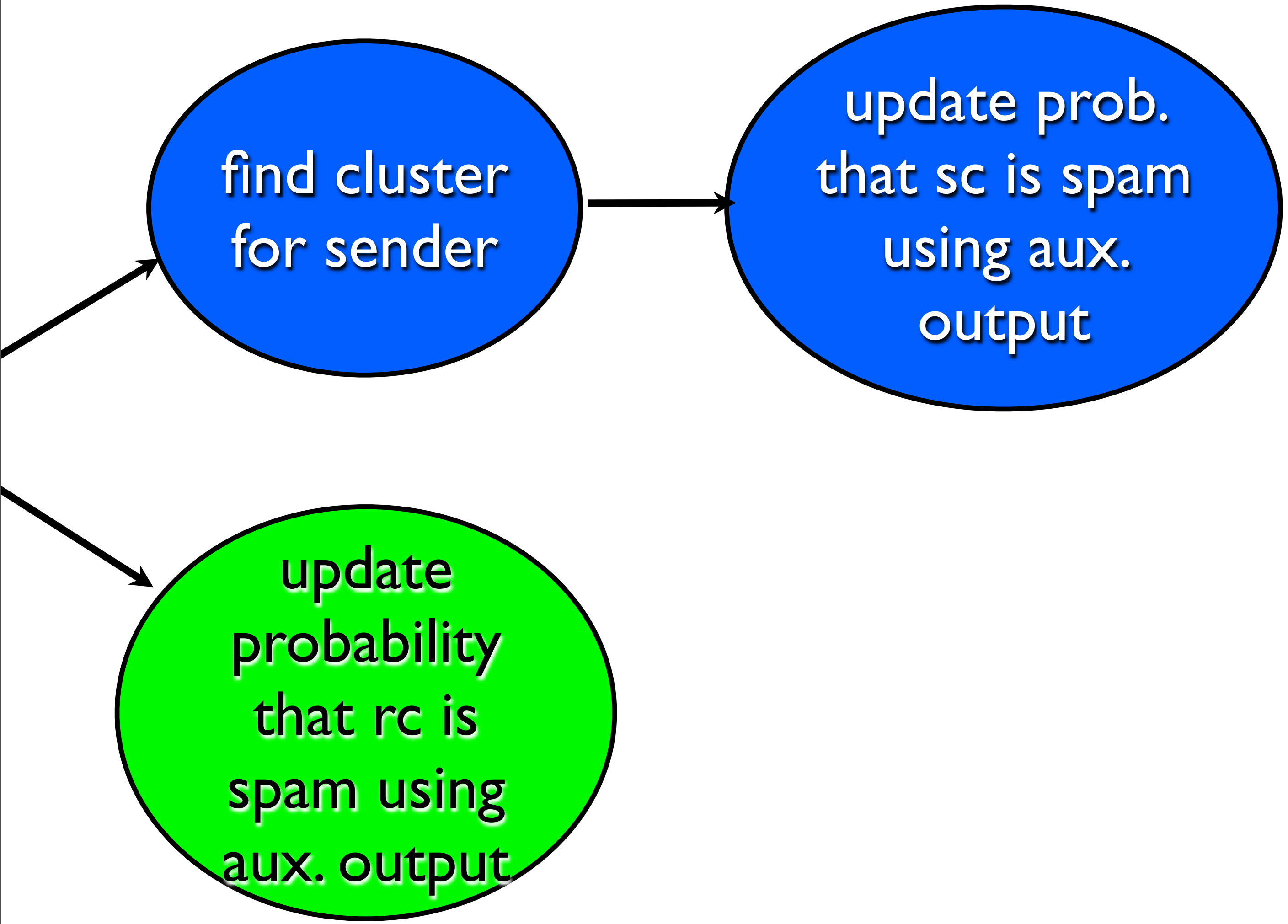
# Basic Algorithm



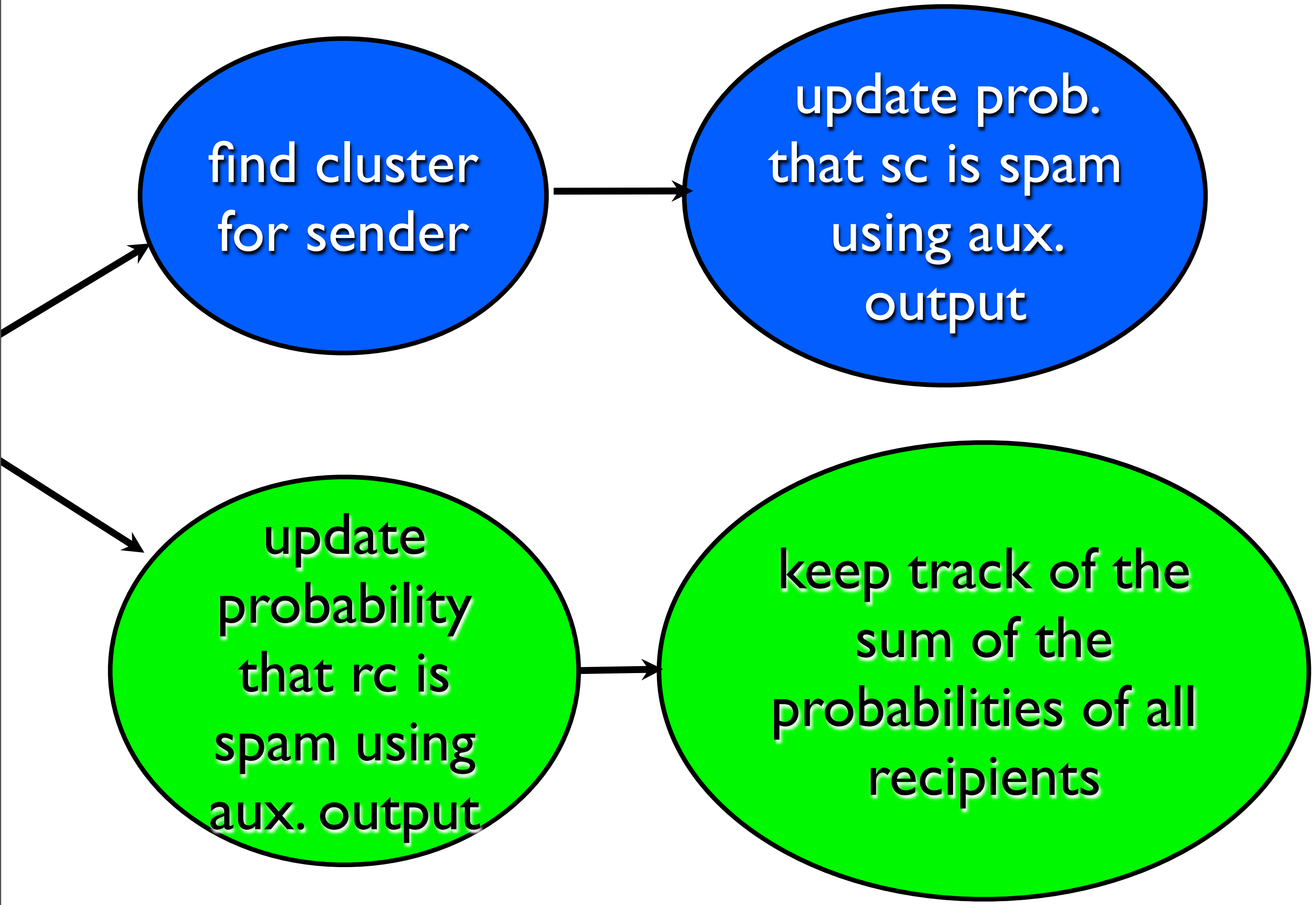
# Basic Algorithm



# Basic Algorithm


find cluster for sender
update prob. that sc is spam using aux. output
update probability that rc is spam using aux. output

# Basic Algorithm



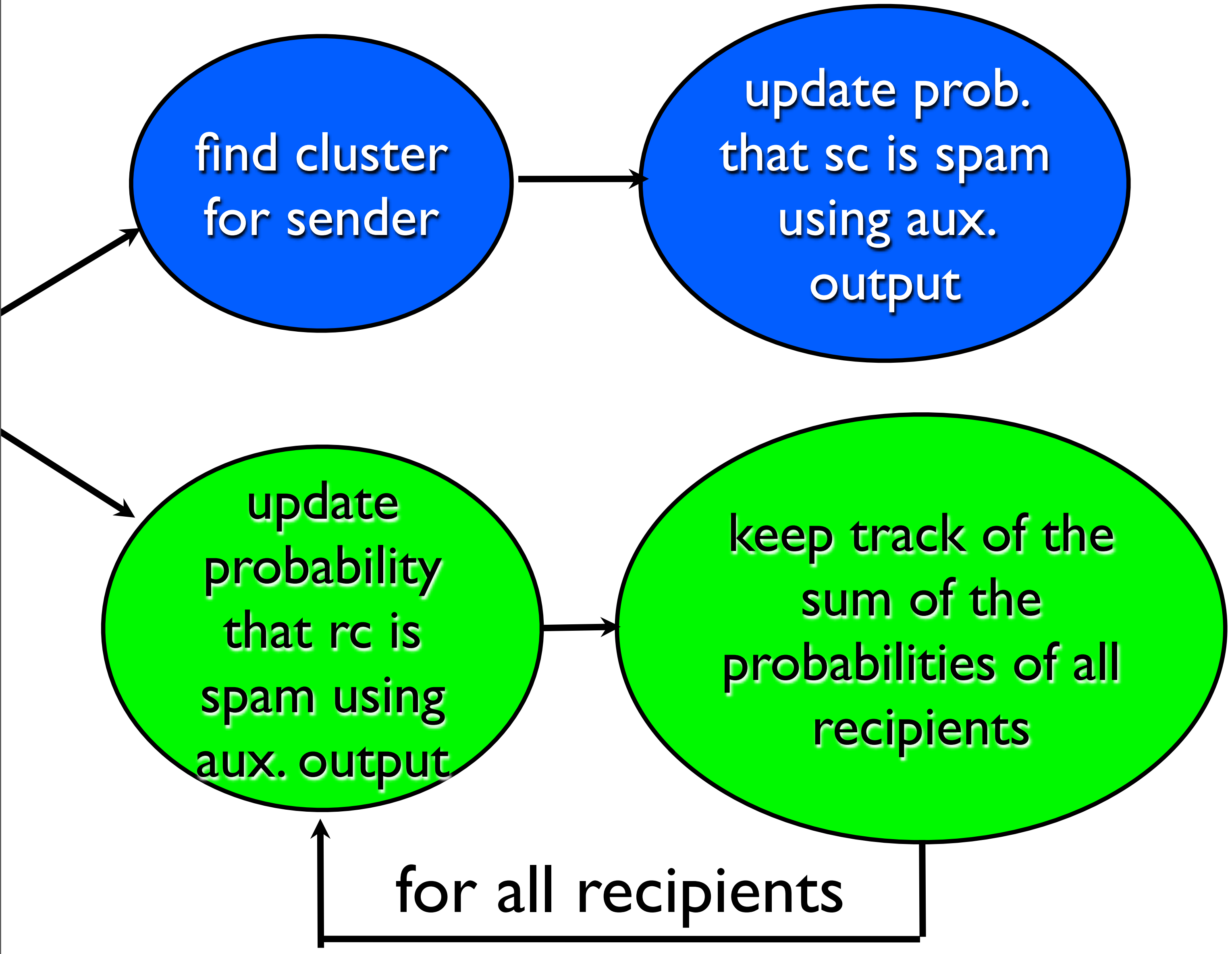
find cluster for sender

update prob. that sc is spam using aux. output

update probability that rc is spam using aux. output

keep track of the sum of the probabilities of all recipients

# Basic Algorithm


find cluster for sender

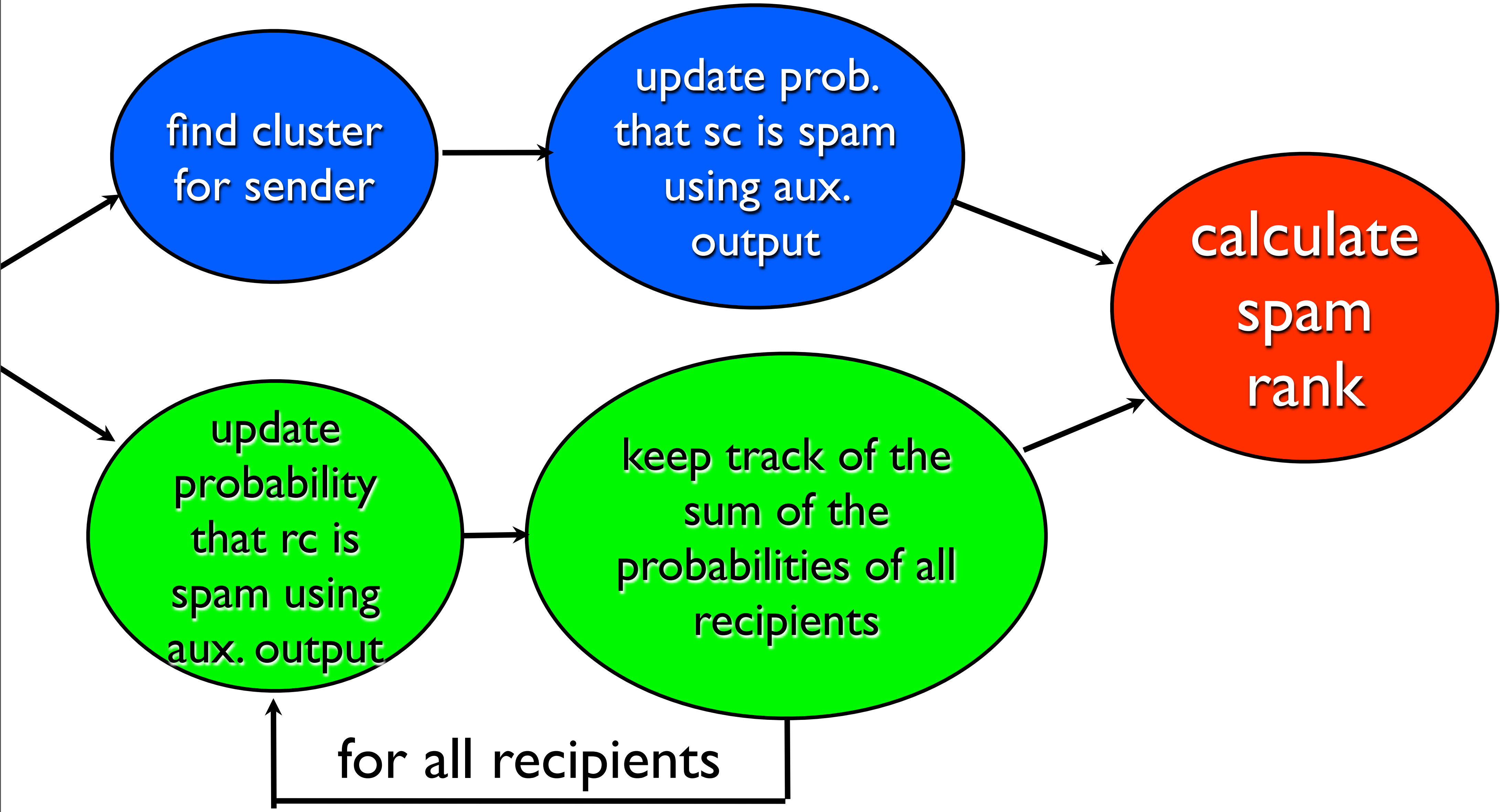
update prob. that sc is spam using aux. output

update probability that rc is spam using aux. output

keep track of the sum of the probabilities of all recipients

for all recipients

# Basic Algorithm



find cluster for sender

update prob. that sc is spam using aux. output

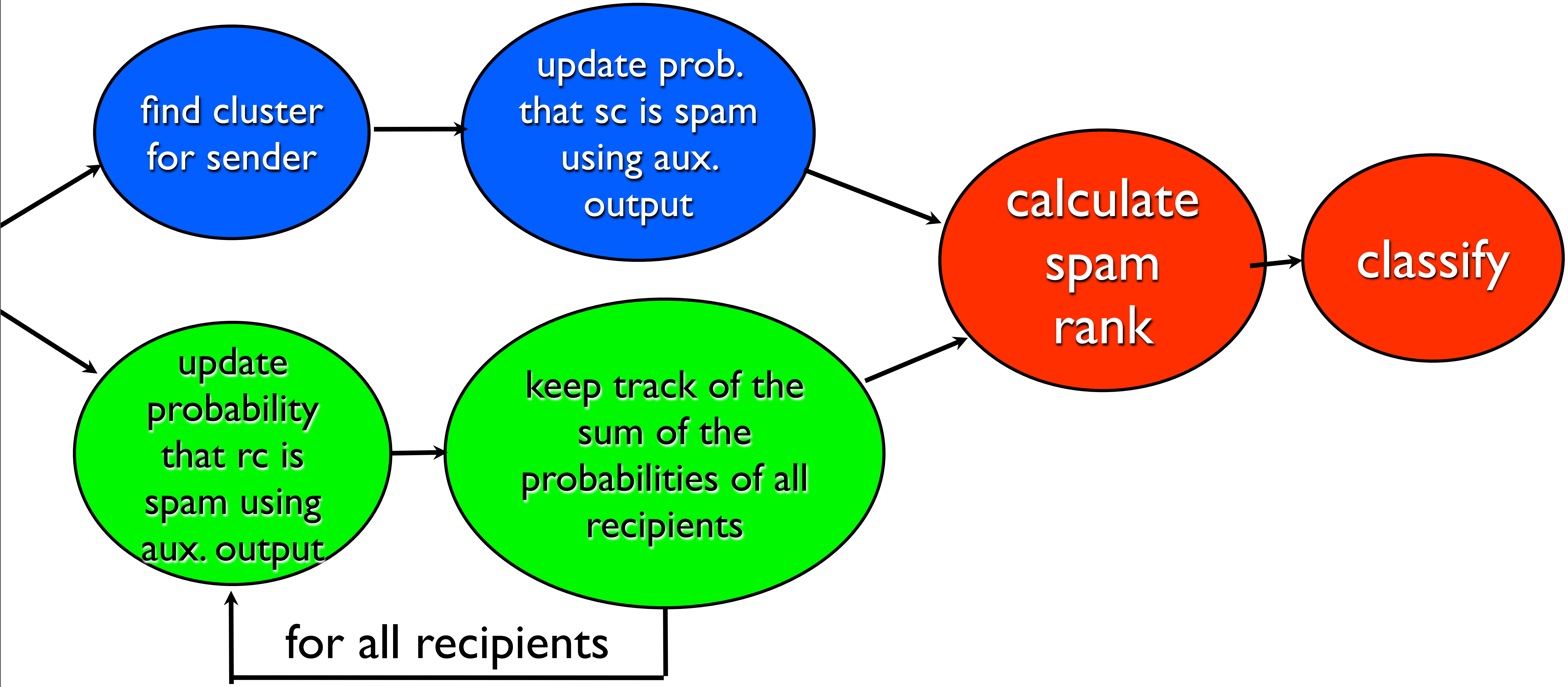
calculate spam rank

update probability that rc is spam using aux. output

keep track of the sum of the probabilities of all recipients

for all recipients

# Basic Algorithm



find cluster for sender

update prob. that sc is spam using aux. output

update probability that rc is spam using aux. output

keep track of the sum of the probabilities of all recipients

for all recipients

calculate spam rank

classify

# Background Equations

$$\vec{s}_i[n] = \begin{cases} 1 & s_i \circledR \ r_n \\ 0 & otherwise \end{cases}$$

vectored version of a sender (or receiver)

# Background Equations

$$sim\left(s_i, s_j\right) = \frac{s_i \ o \ s_j}{\left|s_i\right|\left|s_j\right|} = \cos\left(s_i, s_j\right)$$

numerical representation of similarity
between two senders (receivers)

# Background Equations

$$\overrightarrow{sc_i} = \sum_{s_j \in sc_i} \overrightarrow{s_j}$$

numerical representation of a cluster

# Background Equations

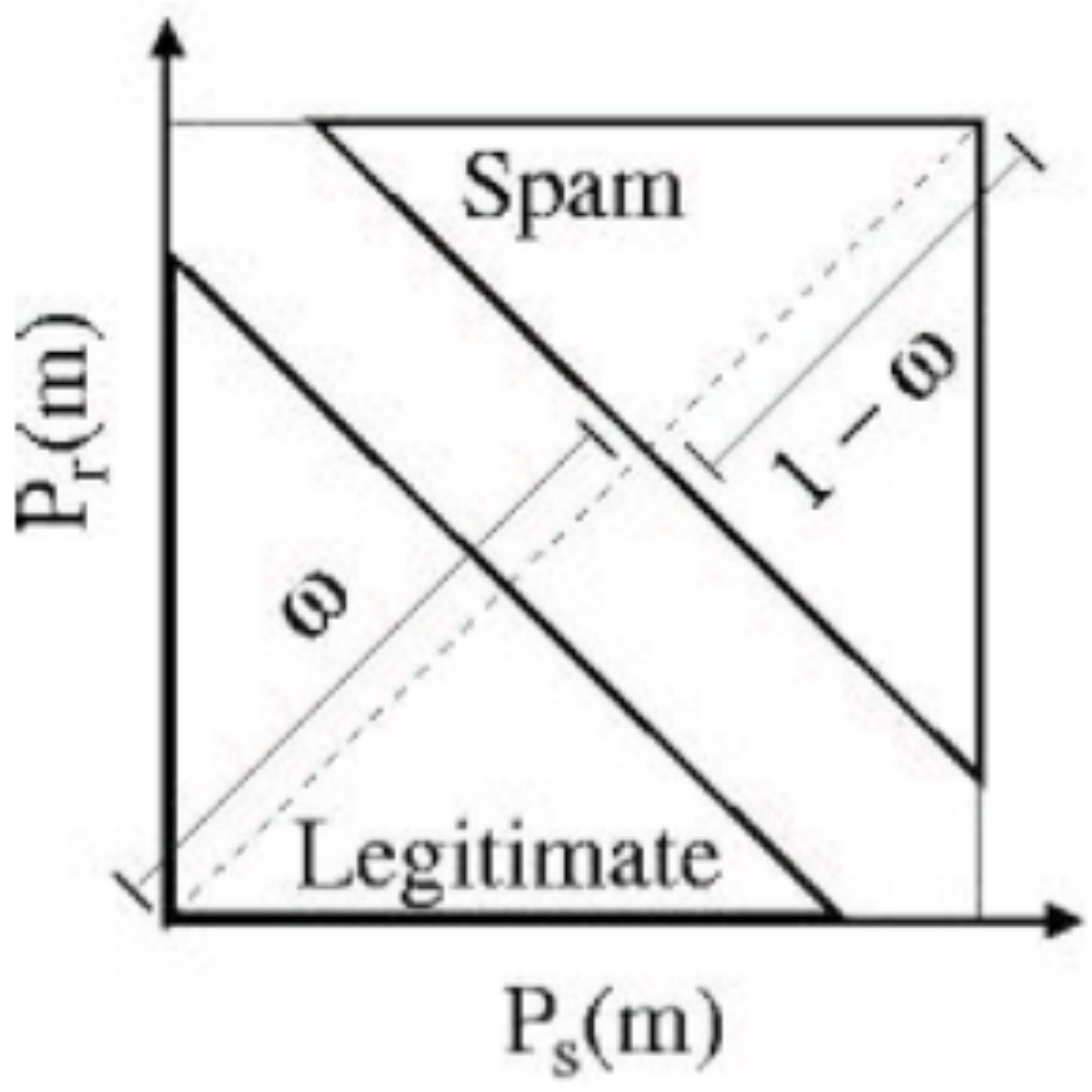
$$sim(sc_i, s)\begin{cases}\cos(\overrightarrow{sc_i} - \vec{s}, \vec{s}) & s \in sc_i \\ \cos(\overrightarrow{sc_i}, \vec{s}) & otherwise\end{cases}$$

numerical representation the similarity between a cluster and a sender (receiver)

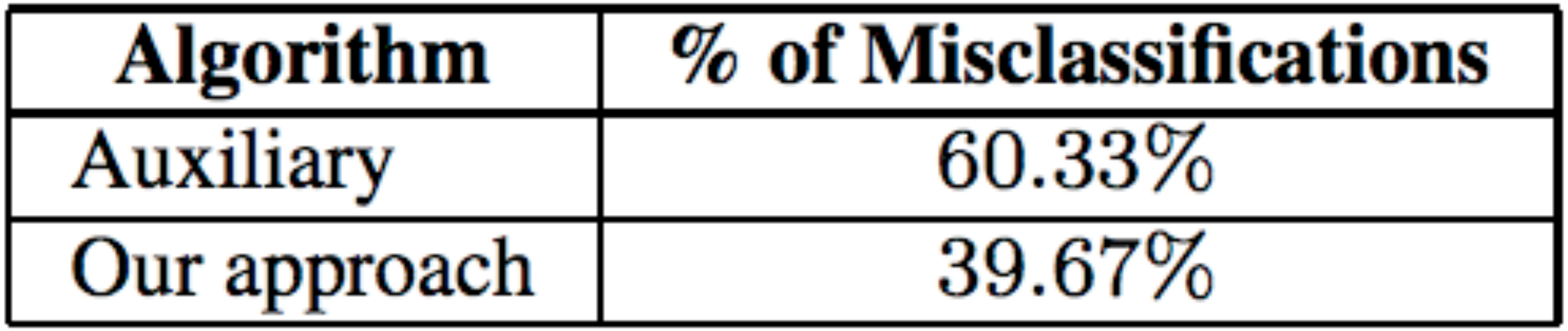
# Thresholds

- sender/receiver is added to cluster if sim is within some bound $\tau$
- marked as SPAM if spam rank is $> \omega$
- marked as not SPAM if spam rank is $< 1-\omega$
- if $\omega >$ rank $> 1-\omega$ then use auxiliary classification

# Graphical Representation



# Results



| Algorithm | % of Misclassifications |
|---|---|
| Auxiliary | 60.33% |
| Our approach | 39.67% |

# Contributions

- provides insigt on how to reduce false positives
- some decent results on simulated environment

# Weaknesses

- is not stand-alone
- no results on real-time systems
- no results on real-world implementation
- generalized senders to domains, not users

# Improvements

- test in real-world
- provide details on implementation