A Framework and Tool for Collaborative Extraction of Reliable Information

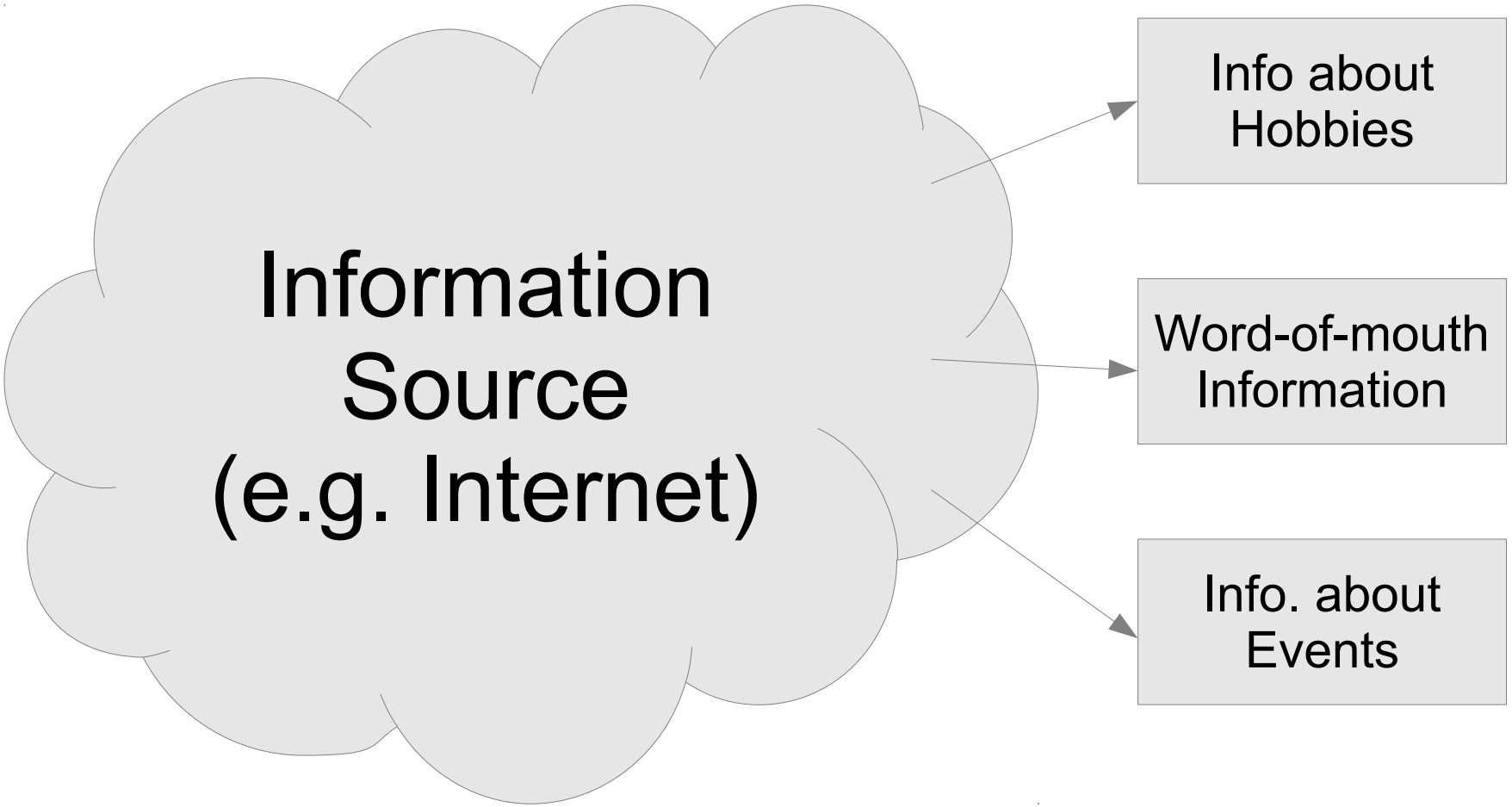
Graham Neubig[1], Shinsuke Mori[2],
Masahiro Mizukami[1]

[1]Nara Institute of Science and Technology
[2]Kyoto University

Background

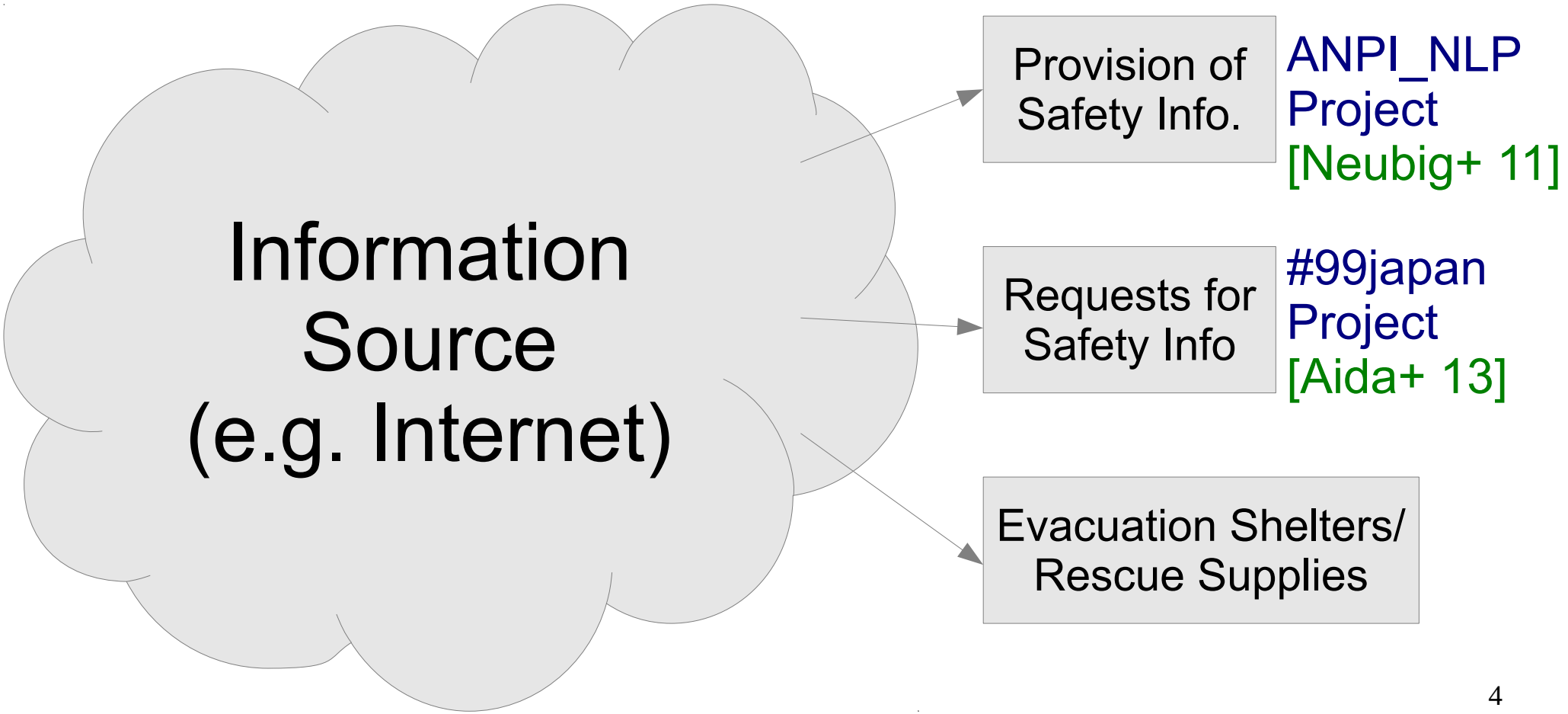
What is Information Extraction?

- Find useful information from large amounts of noise



Information Extraction in Times of Crisis

- Noise is particularly prevalent in times of crisis



Necessities for Crisis-time Information Extraction

- Speed
  - Necessary to provide information ASAP to those in need
- Absolute Reliability
  - Provision of mistaken information could be deadly
  - In general, info will likely require confirmation before consumption
- Difficult to Predict Needs
  - Wildfire → Wind, Earthquake → Diapers, Radiation
- Many volunteers! [Starbird+10, Neubig+11]
- Challenge: How do we let volunteers work efficiently as possible to provide reliable information quickly?

This Work

- We propose a method for efficient extraction of reliable information:
  - Use machine learning (relevance feedback) to decide which examples to show to annotators
  - Web-based collaborative interface to allow multiple annotators to work on a single task
- Evaluation on data from Twitter
- Toolkit freely available open source

webigator:
http://www.phontron.com/webigator

Information Extraction Framework

Information Extraction Task

They really need to open more evacuation areas in Sendai!

They are distributing water at Ishinomaki High School today.

I was able to fill up my car at the gas station at XXX.

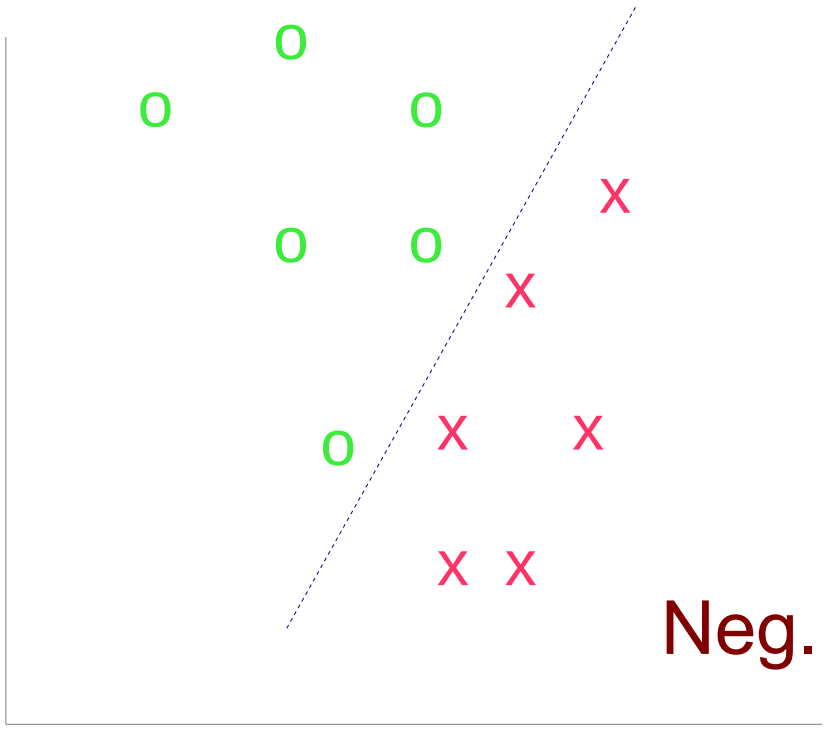
Got to the evacuation center, but I'm almost out of battery!

- Information filtering: Remove documents with no actionable information
- Information extraction: Identify which terms fill slots (e.g. status, location)
- For Twitter, documents are small but numerous, so filtering is a challenge

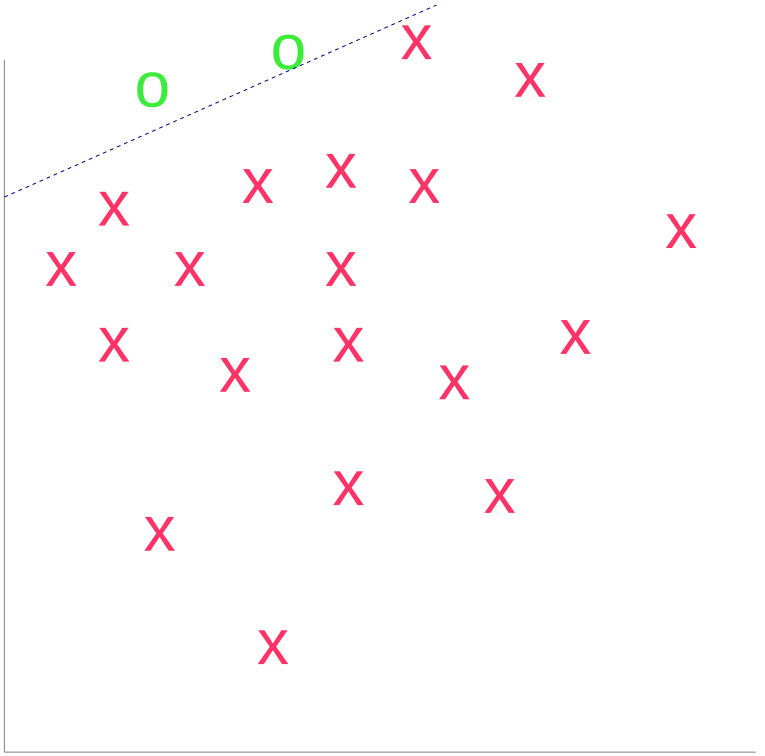
Information Filtering as Classification

- Binary classification of “useful or not?”
  - Define features, use machine learning to learn weights
- Notable for large proportion of negative examples

Normal Classification

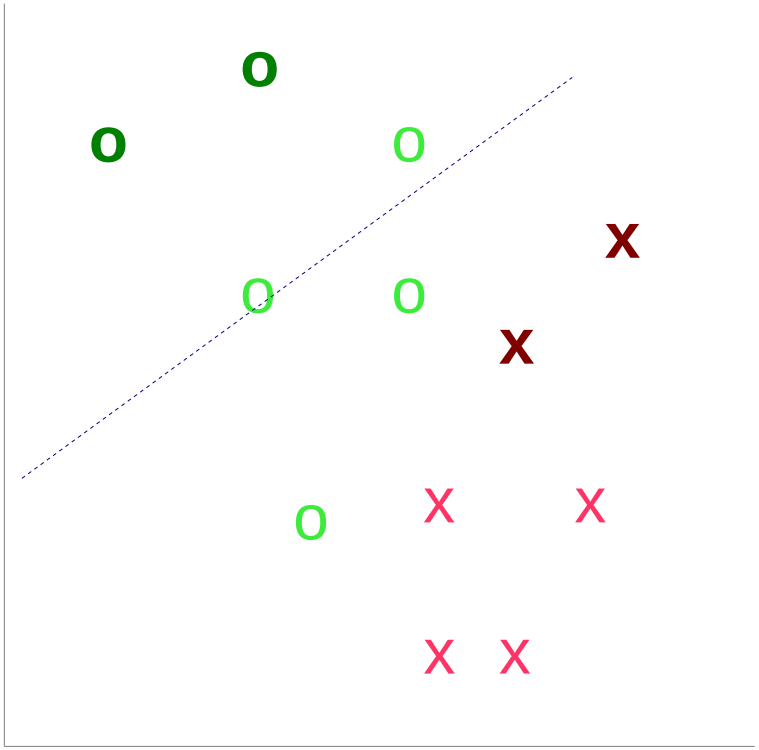


Filtering

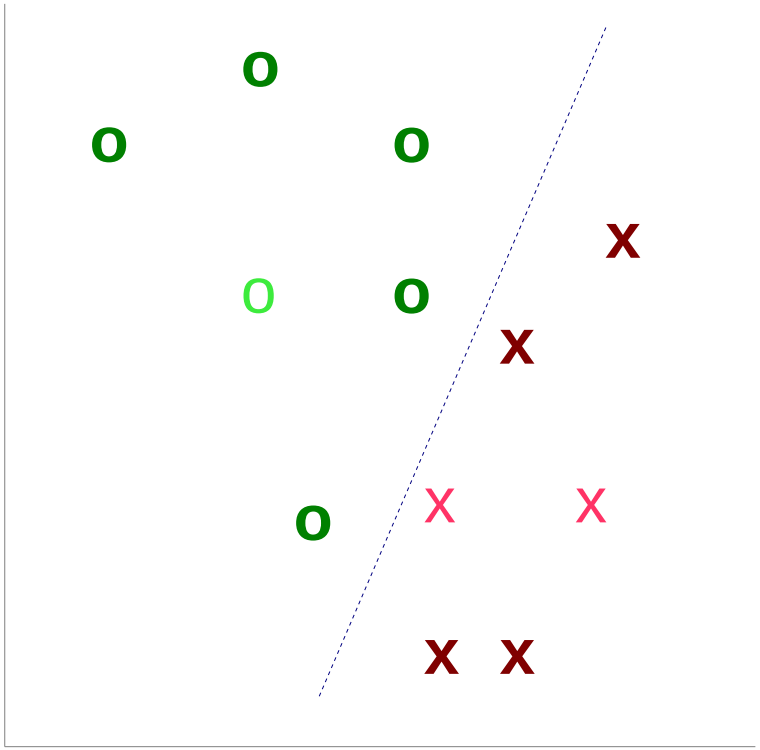


Constructing a Classifier Requires Lots of Data

Little Data



Lots of Data

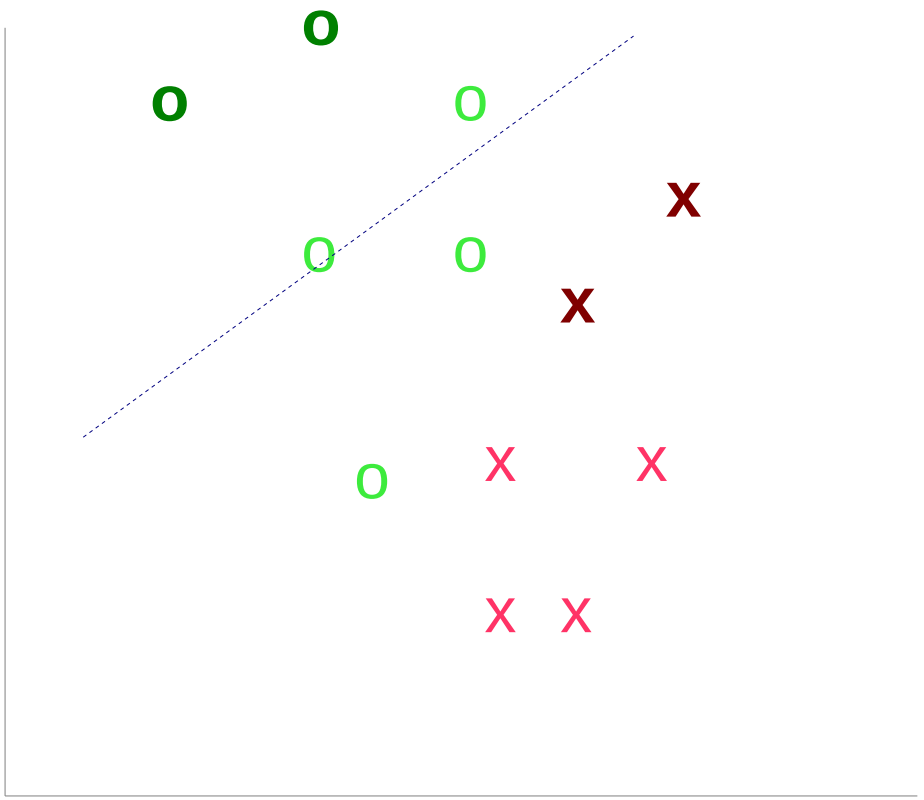


Bold = Lots of Data

Active Learning

- Way to create a good classifier efficiently
- Choose examples to annotate based on predictions

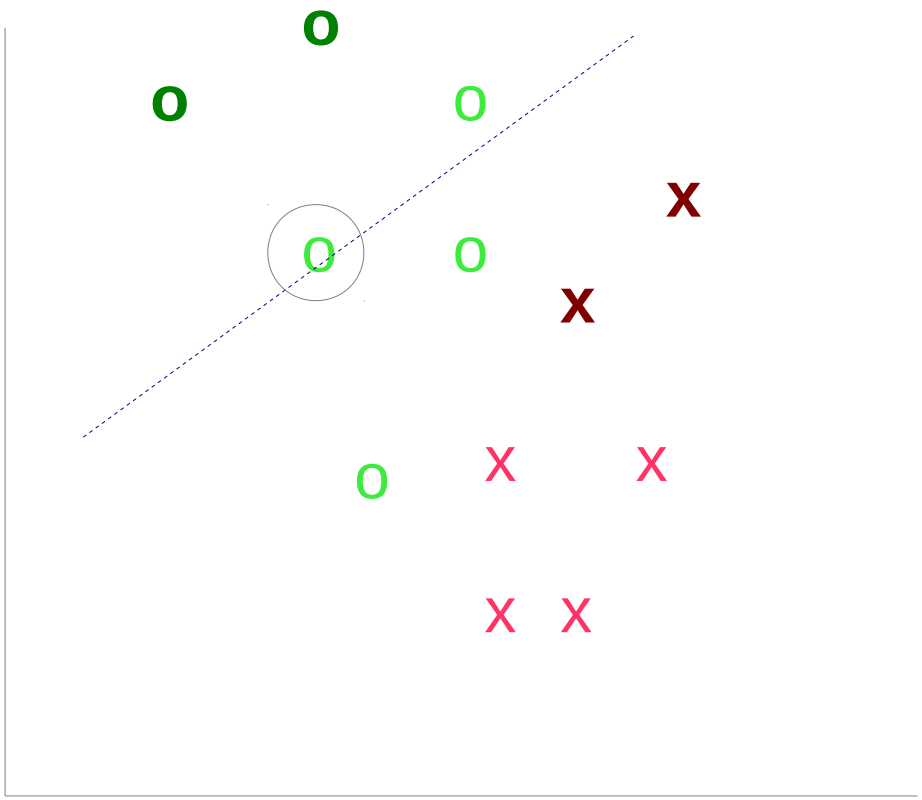
Positive



Negative

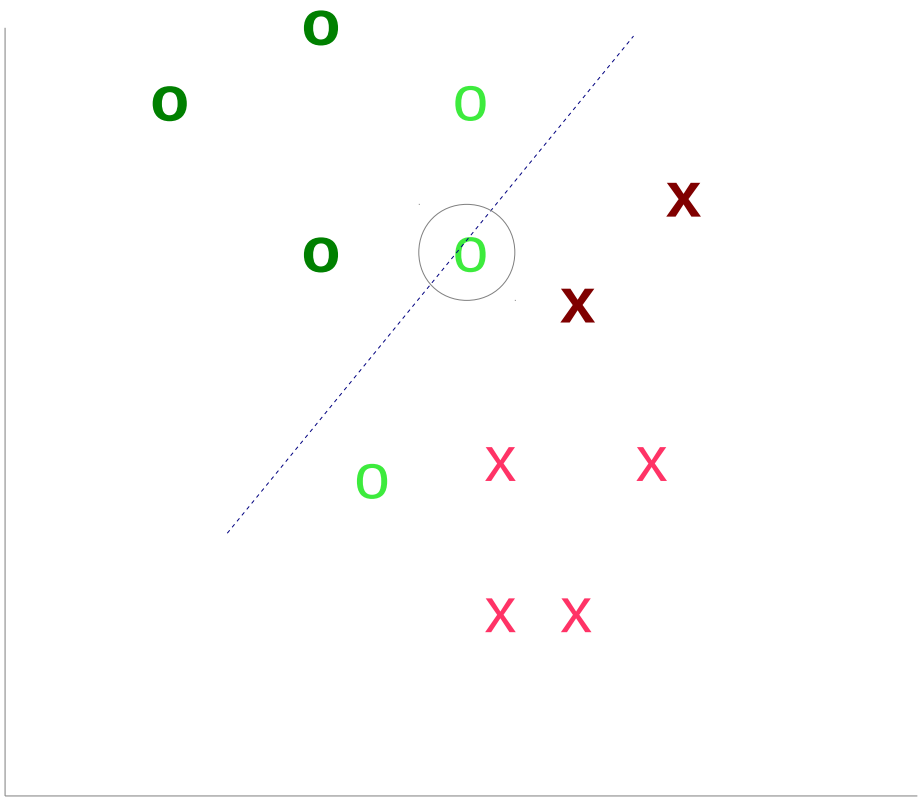
Active Learning

- Way to create a good classifier efficiently
- Choose examples to annotate based on predictions



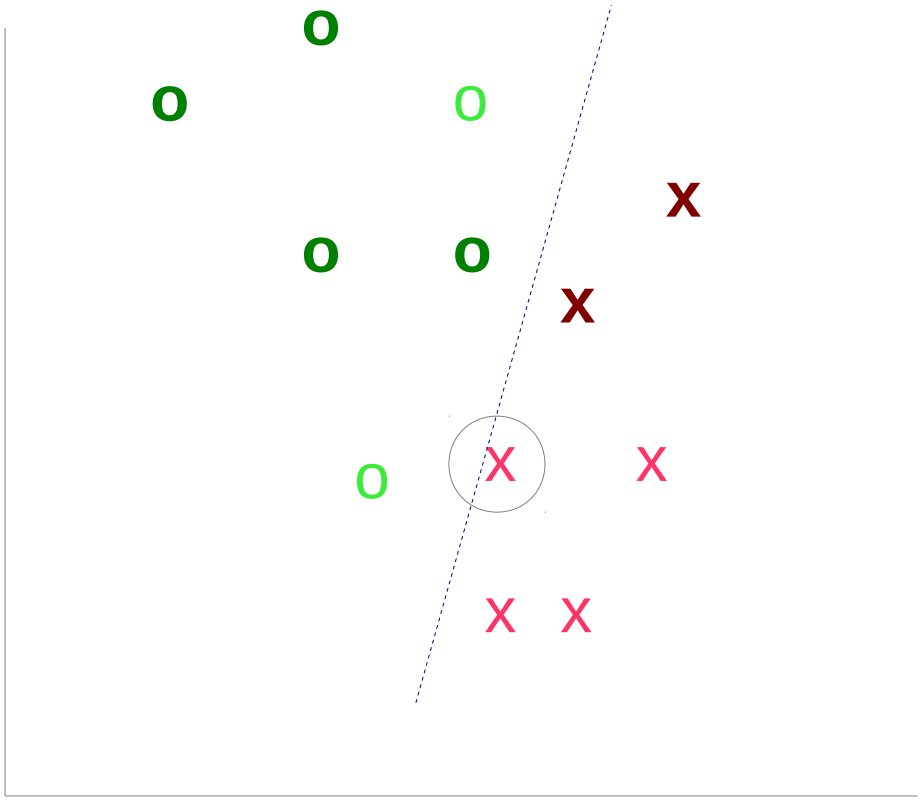
Active Learning

- Way to create a good classifier efficiently
- Choose examples to annotate based on predictions



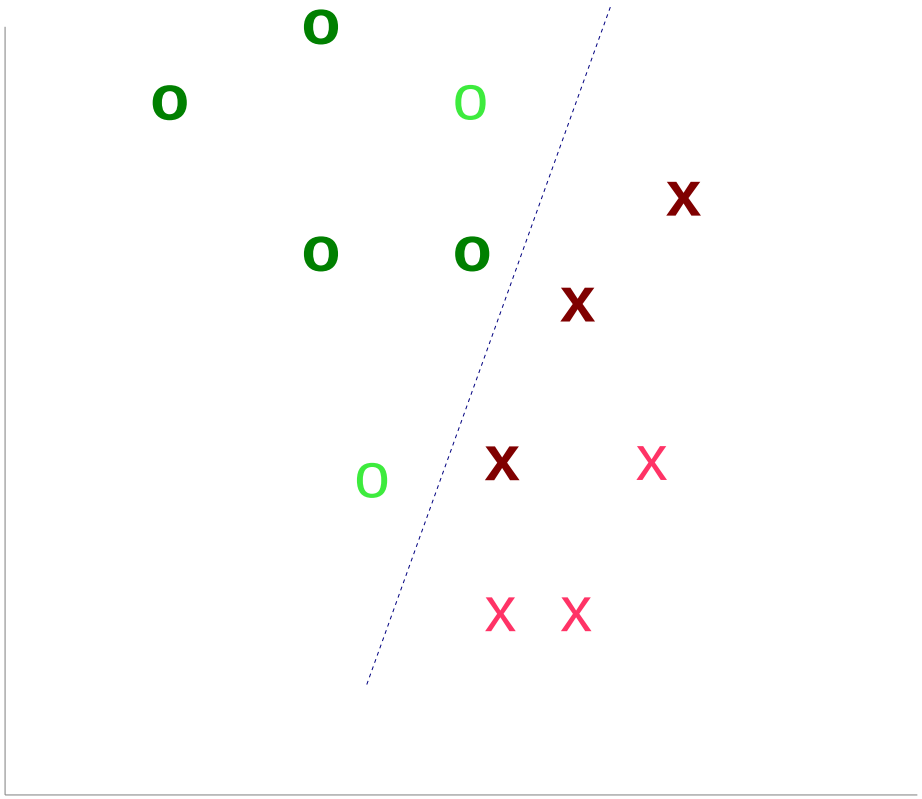
Active Learning

- Way to create a good classifier efficiently
- Choose examples to annotate based on predictions



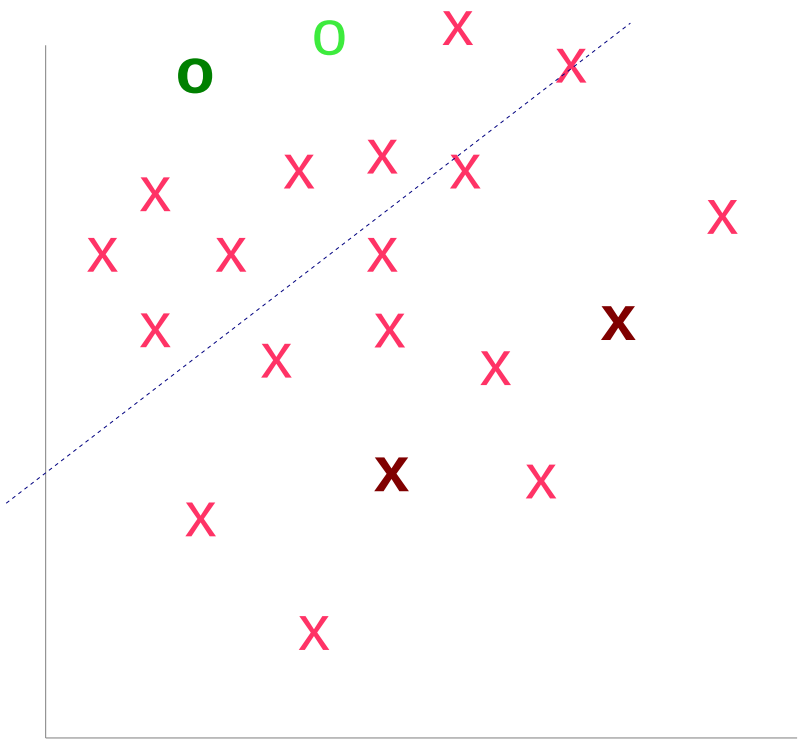
Active Learning

- Way to create a good classifier efficiently
- Choose examples to annotate based on predictions



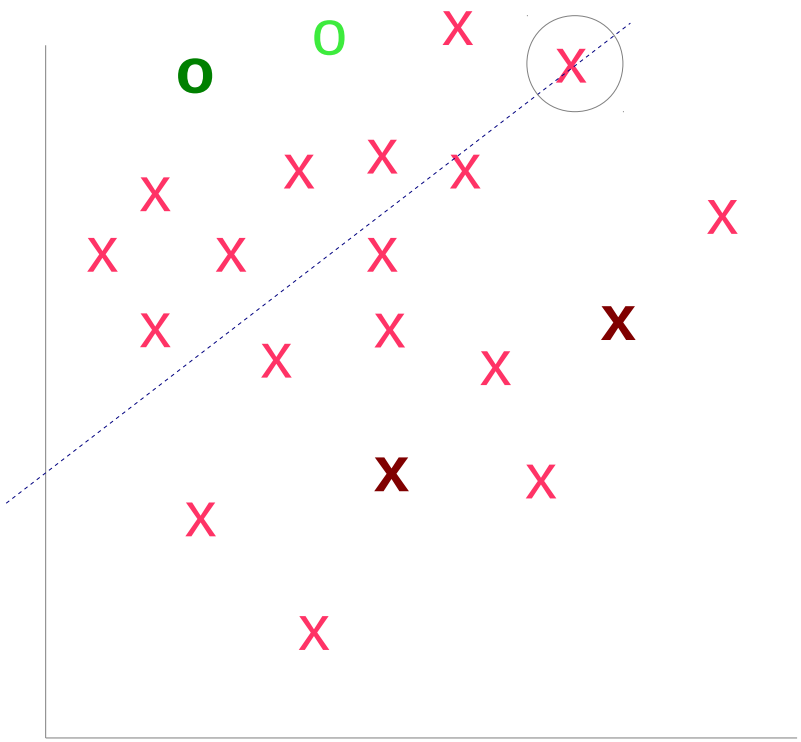
Problems with Unbalanced Data

- In information extraction, almost everything is negative



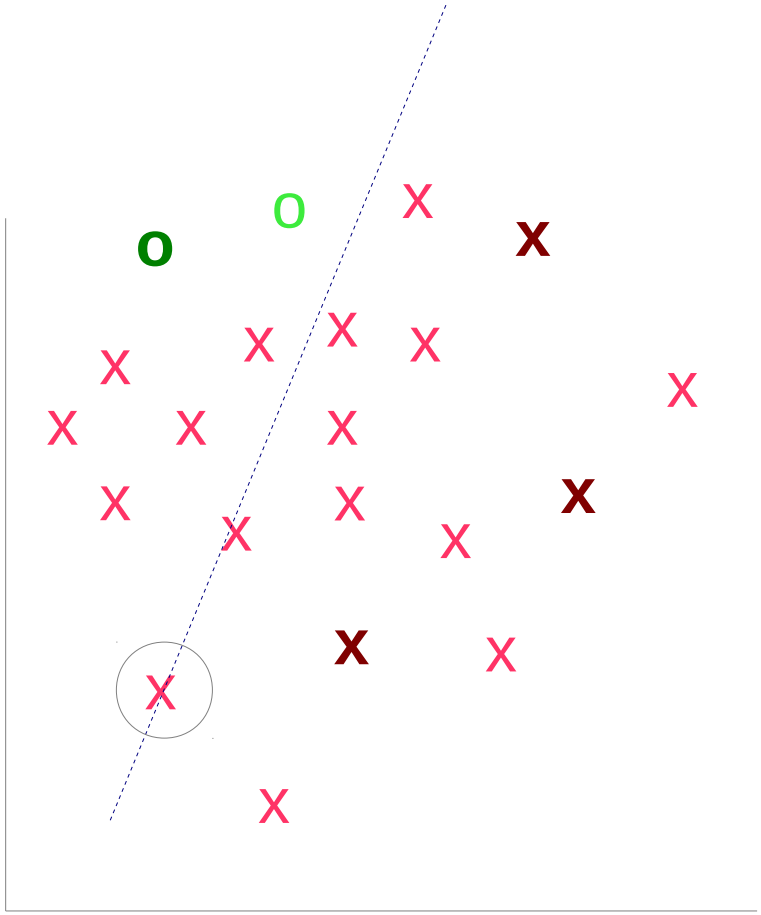
Problems with Unbalanced Data

- In information extraction, almost everything is negative



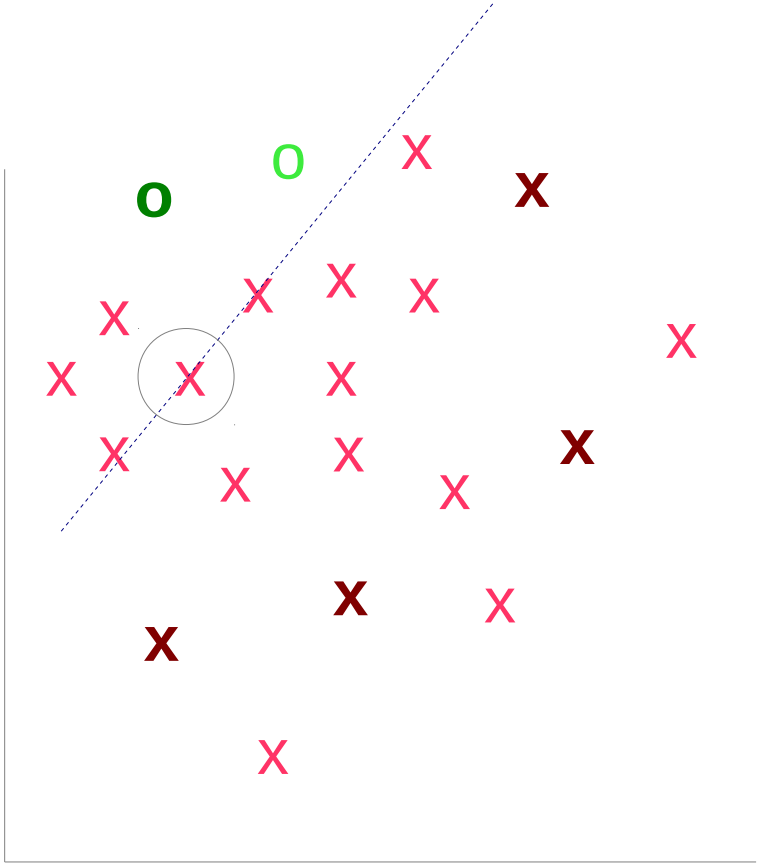
Problems with Unbalanced Data

- In information extraction, almost everything is negative



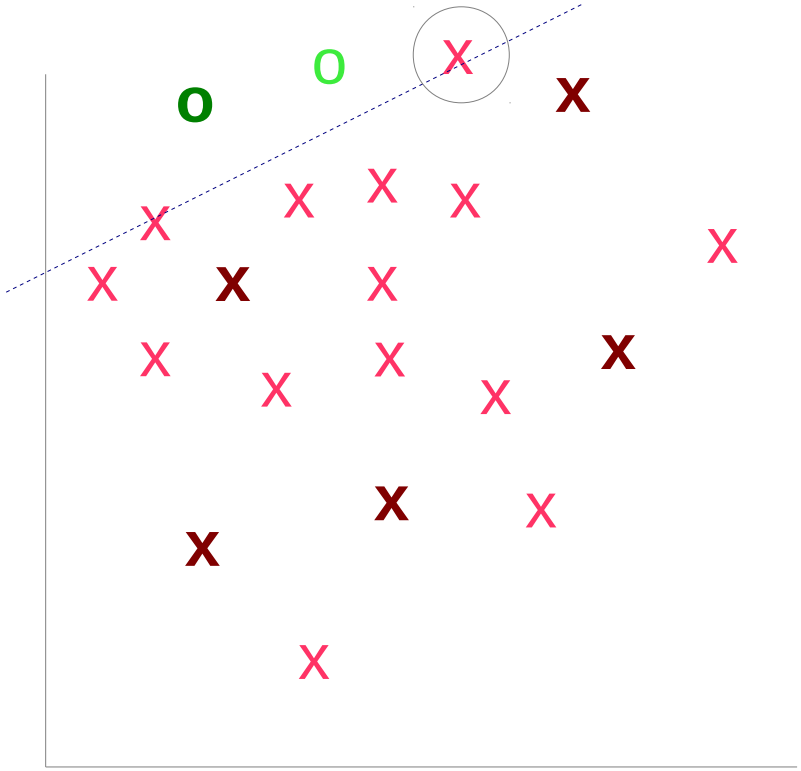
Problems with Unbalanced Data

- In information extraction, almost everything is negative



Problems with Unbalanced Data

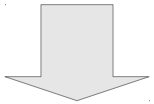
- In information extraction, almost everything is negative



Our Simple Fix

- Small change to example selection criterion

  Standard: Select low confidence examples

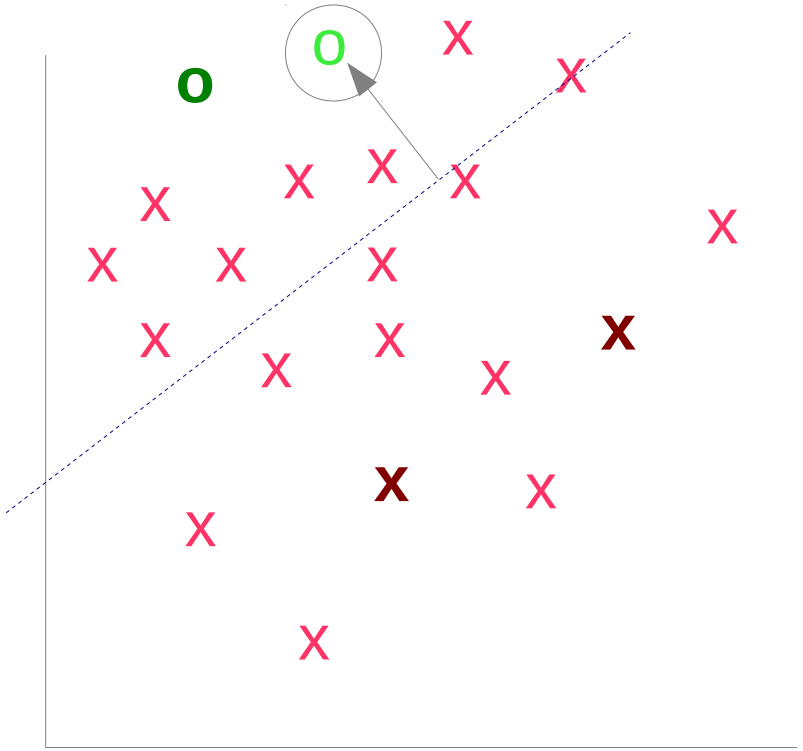
  

  Proposed: Select examples with high probability of being positive

- Effective when final human check is necessary
  - Labeling a positive example =
    finding a highly reliable piece of information

Our Simple Fix

- Finds many positive examples quickly



- Using these positive examples, learn characteristics that help pick out more

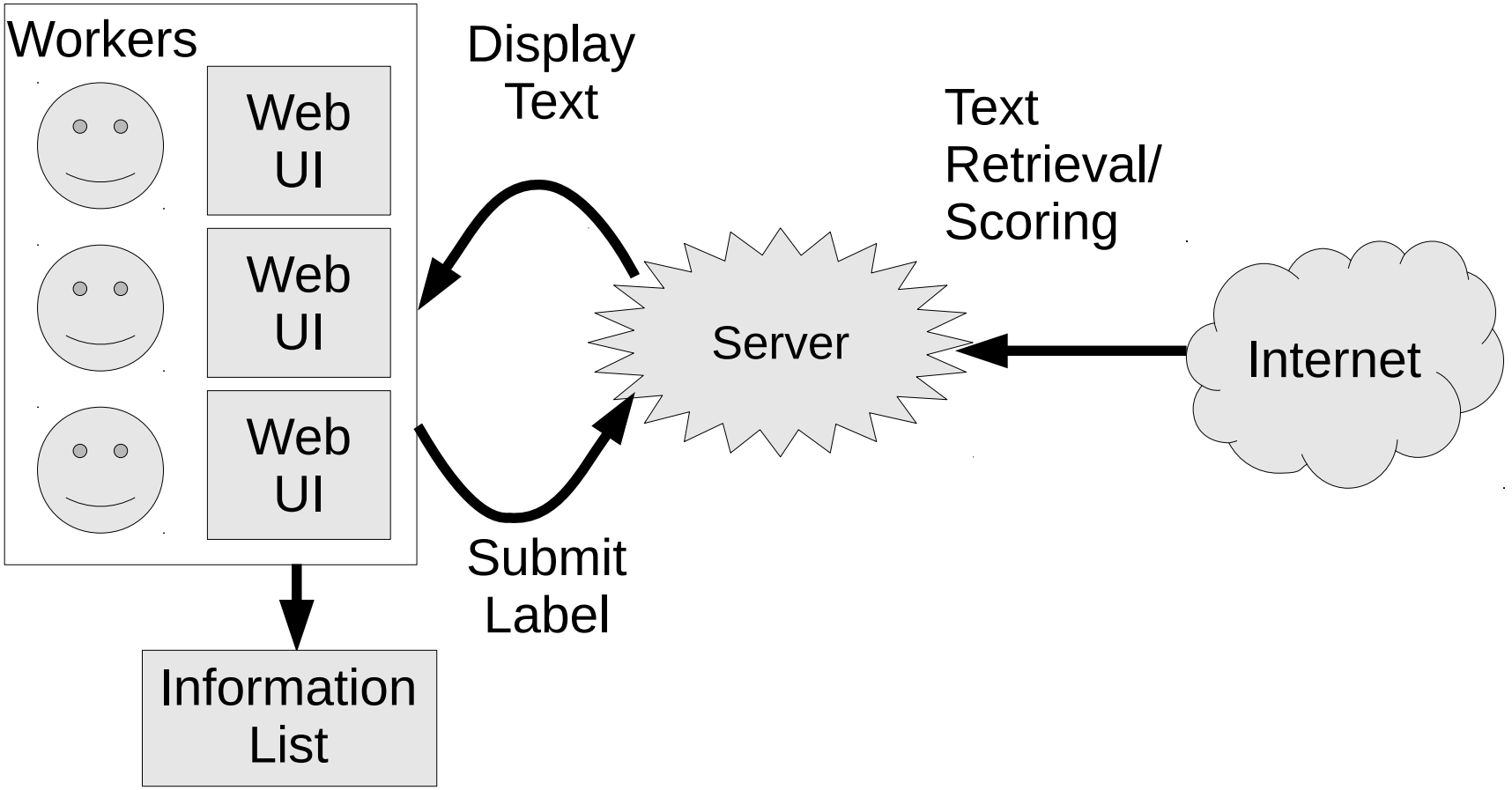
Scaling Up

Too Much Data!

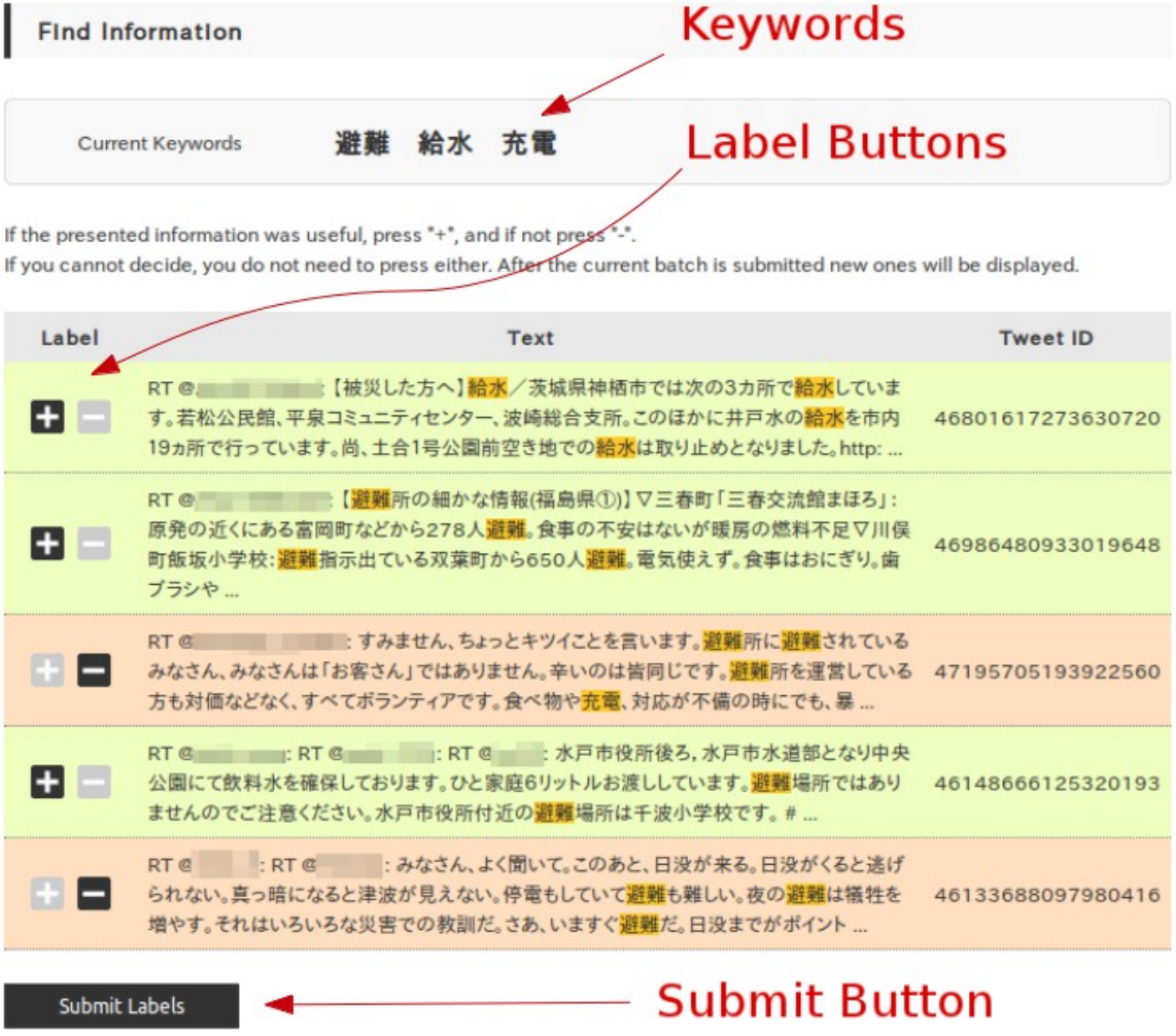
- e.g. Twitter after the Great East Japan Earthquake = peak of 1237 tweets/second
- Problems with:
  - Viewing even the high scoring tweets with one person
  - Rescoring every tweet after each round of learning

Collaborative Web-based Interface

- Allow multiple annotators to cooperate



Web Interface



Efficiency Improvements

1) Simple keyword search filter

Type	Keywords
Evacuation/Supplies	evacuation area, water supplies, food supplies
Safety Info Request	contact, cannot, waiting
Safety Info Provision	contact, safe

2) Rescoring policy

- Maintain a sorted list of highly scored examples
- When retrieving next example:
  - Choose the example highest in the cache, rescore
  - After rescoring, still better than second best, return
  - Otherwise, return to beginning

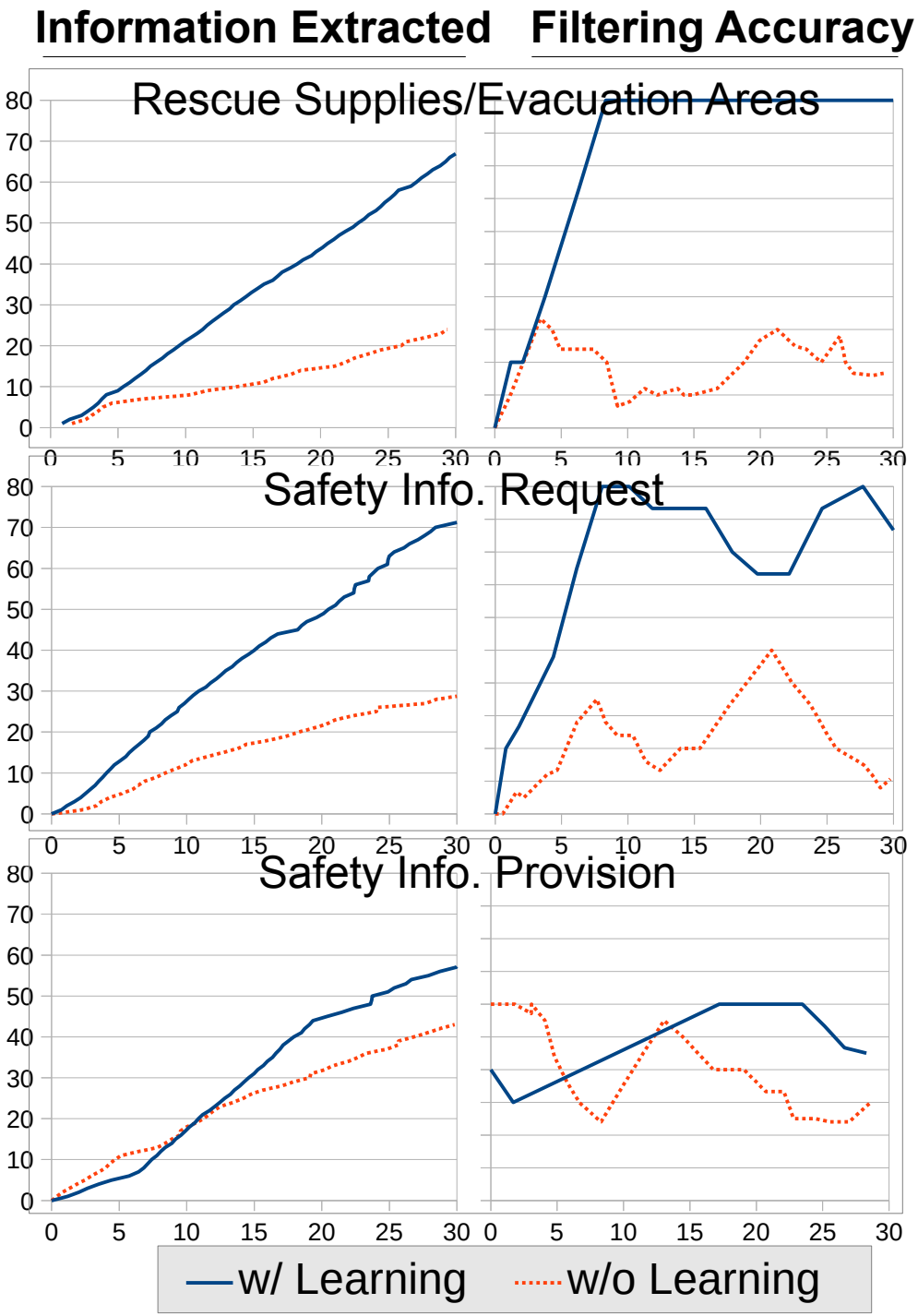
Experiments

Evaluation

- Compared Methods:
  - Keyword search
  - Proposed learning-based method
- Target:
  - 179M tweets week after Great East Japan Earthquake
  - Three types of info: evacuation/rescue supplies, safety info request, safety info provision
- Evaluation measure:
  - Amount of reliable information extracted in 30 mins.
  - Use shared Google Doc as repository for information

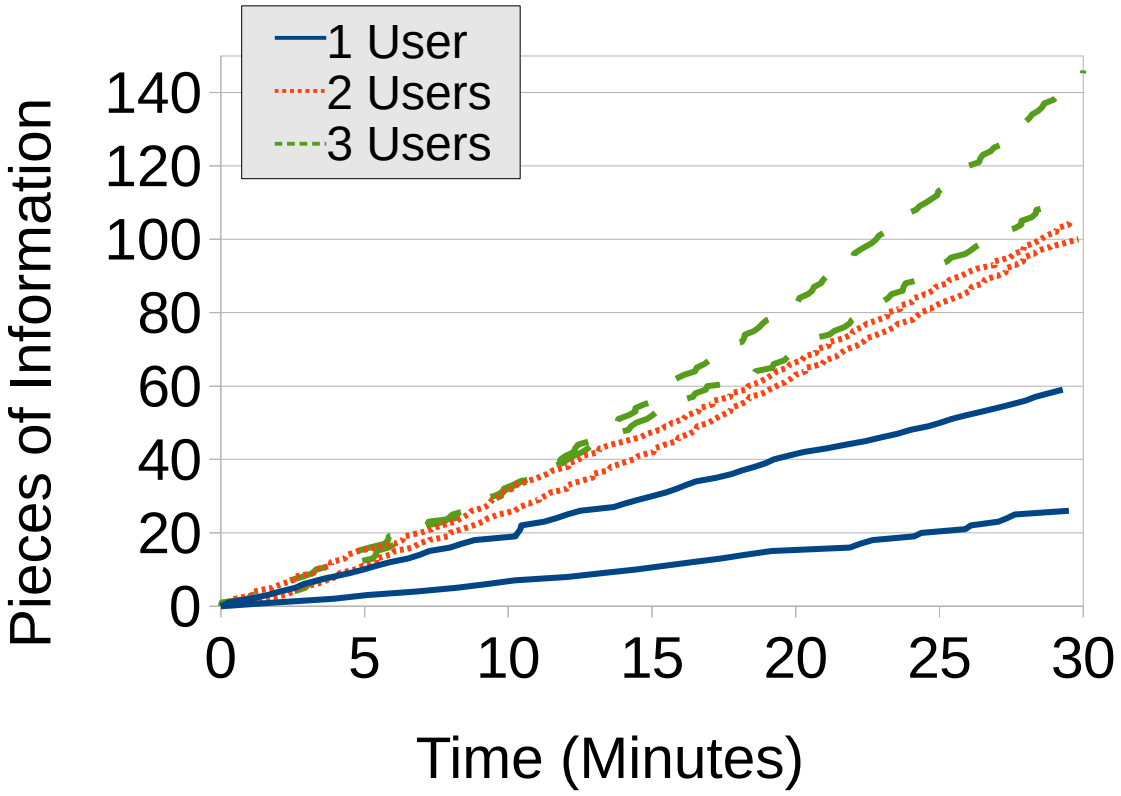
Effect of Learning

- Experiments with one annotator for three tasks
- Observable increase in amount of information extracted and accuracy
- Some tasks easier than others



Effect of Collaboration

- Experiments with 1-3 users using same interface



- As expected, increasing users = increasing efficiency

Conclusion

- A method for information filtering that focuses on positive examples
- More effective than simple keyword search
- Remaining challenges:
  - Identification/clustering of duplicates
  - Application to identification of slots as well

webigator:
http://www.phontron.com/webigator