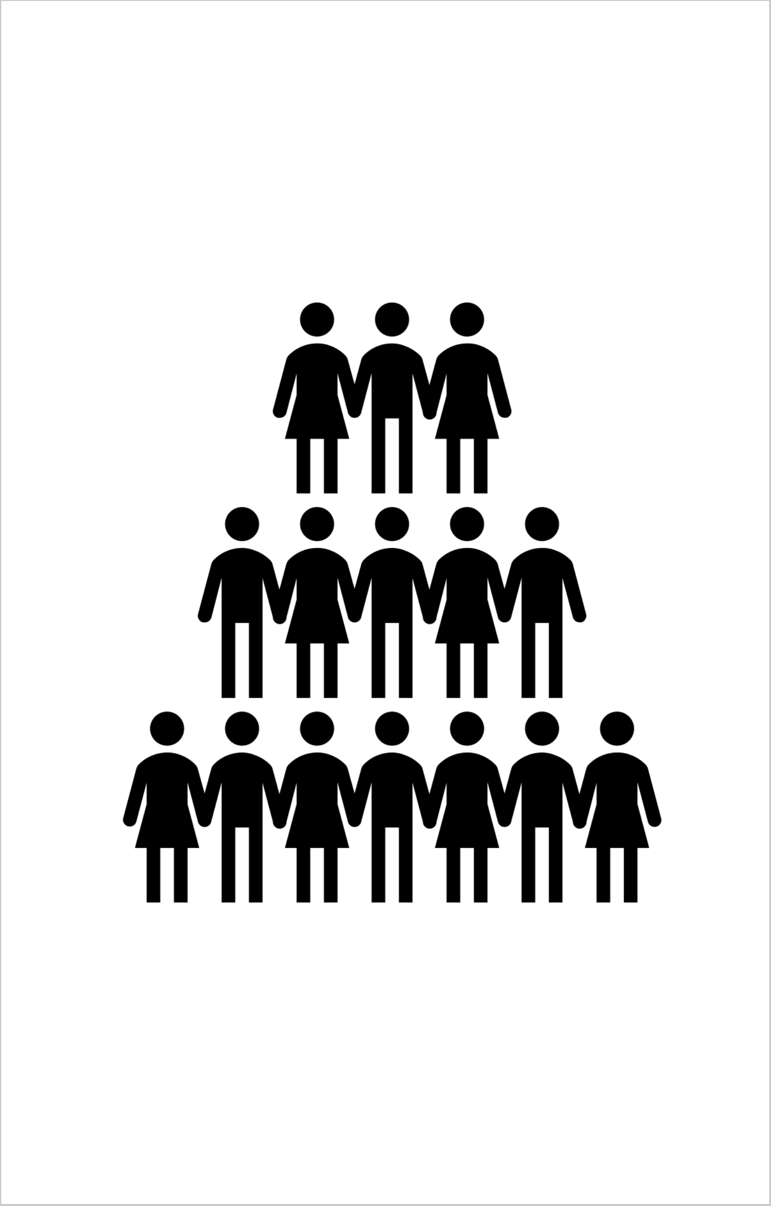
Creating Sentiment and Affect Lexicons

Natalie Parde

UIC CS 421



How to build a lexicon?

- Expert labels
  - Very reliable 🙂
  - Costly and time-consuming 😕
- Crowdsourced labels
  - Less reliable 😕
  - Inexpensive and quick 🙂

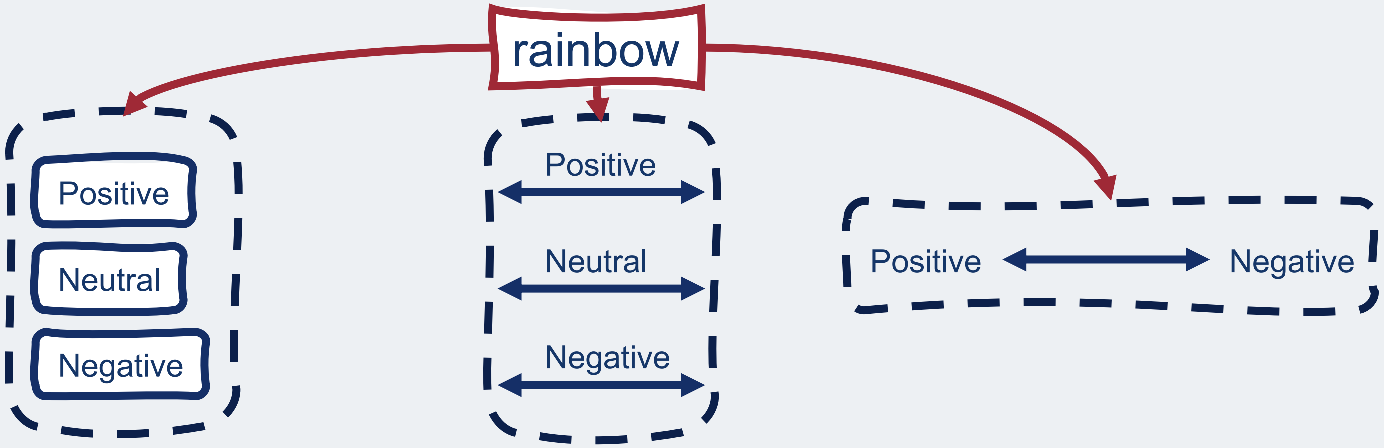
Crowdsourcing Resources

- **Amazon Mechanical Turk:**
  - https://www.mturk.com
- **Appen:**
  - https://appen.com
- **Your own online survey**

Annotation Schemata

- What kinds of labels will be permissible for your annotators?
- How will they know which labels to select?

Positive: A word that evokes a happy or content emotion.
Examples: *love*, *great*, *happy*
Neutral: A word that does not particularly evoke any emotion.
Examples: *pencil*, *refrigerator*, *khaki*
Negative: A word that evokes a sad or angry emotion.
Examples: *violence*, *evil*, *upset*



Adjudication

- Third-party adjudicator
- Majority label
- Average label

Semi-Supervised Induction of Affect Lexicons

- **Semi-supervised label induction:** The process of labeling new, unlabeled instances based on their similarity to instances in a small, labeled seed set
- Two main families:
  - **Axis-based** induction
  - **Graph-based** induction

Axis-Based Lexicon Induction

- Given a seed set, how similar is the instance to positive instances and how different is it from negative instances?
  - Compute an embedding for each seed word
  - Find the centroid of the embeddings for positive words, and the centroid of the embeddings for negative words
    - $\mathbf{V}^{+} = \frac{1}{n}\sum_{n}^{n} E(w_i^{+})$
  - Compute the axis by subtracting one centroid from another
    - $\mathbf{V}_{axis} = \mathbf{V}^{+} - \mathbf{V}^{-}$
  - Compute the similarity between a given word embedding and the axis
    - $\text{score}(w) = \cos(E(w), \mathbf{V}_{axis}) = \frac{E(w) \cdot \mathbf{V}_{axis}}{\|E(w)\| \|\mathbf{V}_{axis}\|}$
  - Higher similarities indicate closer alignment with the positive class

Graph-Based Lexicon Induction

- Given a graph that connects words with their nearest neighbors, how likely is it that a random walk from a positive word ends on the given word?
  - Define a graph that connects each word to its *k* nearest neighbors, with edges weighted by word similarity
  - Identify words in the graph belonging to a labeled seed set
  - Starting at a word from the seed set, perform an edge-weighted random walk
  - Assign an unlabeled word's score based on the probability of landing on it during a random walk from a positive seed and a random walk from a negative seed
    - $\text{score}^+(w_i) = \frac{\text{score}^+(w_i)}{\text{score}^+(w_i)+\text{score}^-(w_i)}$
  - Repeat multiple times using bootstrapping, and assign confidence to word scores based on their standard deviation across multiple runs