TWEET GROUPER

Daniel Stoll, Aditya Bhandaru
15-780 Graduate AI

PROBLEM

- On Twitter it is common to follow many (100s) accounts.
- Your stream has interleaved content types.
- How would you read about *just* sports? Music? Tech?
- **No current scalable solution to do it!**

OUR SOLUTION

Fine grained category system within your stream!

- **Create a label and provide a few authors who meet the criteria.** We create classifiers for each label and run them on your stream.
- **Is the wrong stuff showing up under your labels?** Just remove them from the label and we will retrain our classifiers!
- **Simple feedback mechanism for an evolving classifier.**
- **How does it work?** Using ML we maintain classifiers for each label. The heuristic is driven by the language used by the label's authors.

RELATED WORKS

- Other works used a classifier based on semantics of a Tweet
  - e.g. Opinions, Random Thoughts, Questions, etc.
- They also focused on analyzing the data by determining the percentage of Tweets in each category.
- Our project uses a classifier based on type of Author.
  - e.g. Sports, Tech, Music, etc.
- We are not focused on analyzing the data, but instead on improving the average user's Twitter experience.

REFERENCE

https://github.com/abhandaru/tweet-grouper, Github Repository.
http://firstmonday.org/ojs/index.php/fm/article/view/2745/2681, Stephen Dann, *Twitter Content Classification.*
http://know-center.tugraz.at/wp-content/uploads/2010/12/Master-Thesis-Christopher-Horn.pdf, Christopher Horn, *Analysis and Classication of Twitter Messages*.

ALGORITHM

- For each canonical author under a label, compute the most commonly occurring words (filter out meaningless tokens).
- To test membership for an author, compute the similarity score. Only accept authors 1σ above the mean score.
- Account for anti-similarity when authors are rejected from a label.

```
def similarity(self, rank1, rank2, count1, count2):
    w1 = count1
    w2 = count2
    return (w1 + w2) / ((1 + rank1)*(1 + rank2))
```

EVALUATION

- Sample data set of 2400 Tweets
- Create label with description, compare observed similarities with expected values.

DRIVER OUTPUT

Label: Schools(CarnegieMellon, Yale)
Top words: yale, mt, cmu, today, us, tonyawards, alumni

Author	Similarity
@SportsCenter	0.26307
@espn	0.45490
@Stanford	10.9394
@Harvard	5.53102
@facebook	5.72631
@nyjets	2.17132
@NBA	0.06691
@UWaterloo	4.23970
Average	3.30712

Members: Yale, CarnegieMellon, Stanford, Harvard, facebook, UWaterloo - *can add facebook to an exclude list.*

FUTURE WORK

- Improving heuristic to more accurately classify authors
- Optimizations (Python implementation is slow)
  - Caching similarity scores
  - Use C++ or another compiled language