# Online Social Media Sensing 2: Geo-location and Redundant Content

CSE 40437/60437-Spring 2015

Prof. Dong Wang

# Papers discuss today

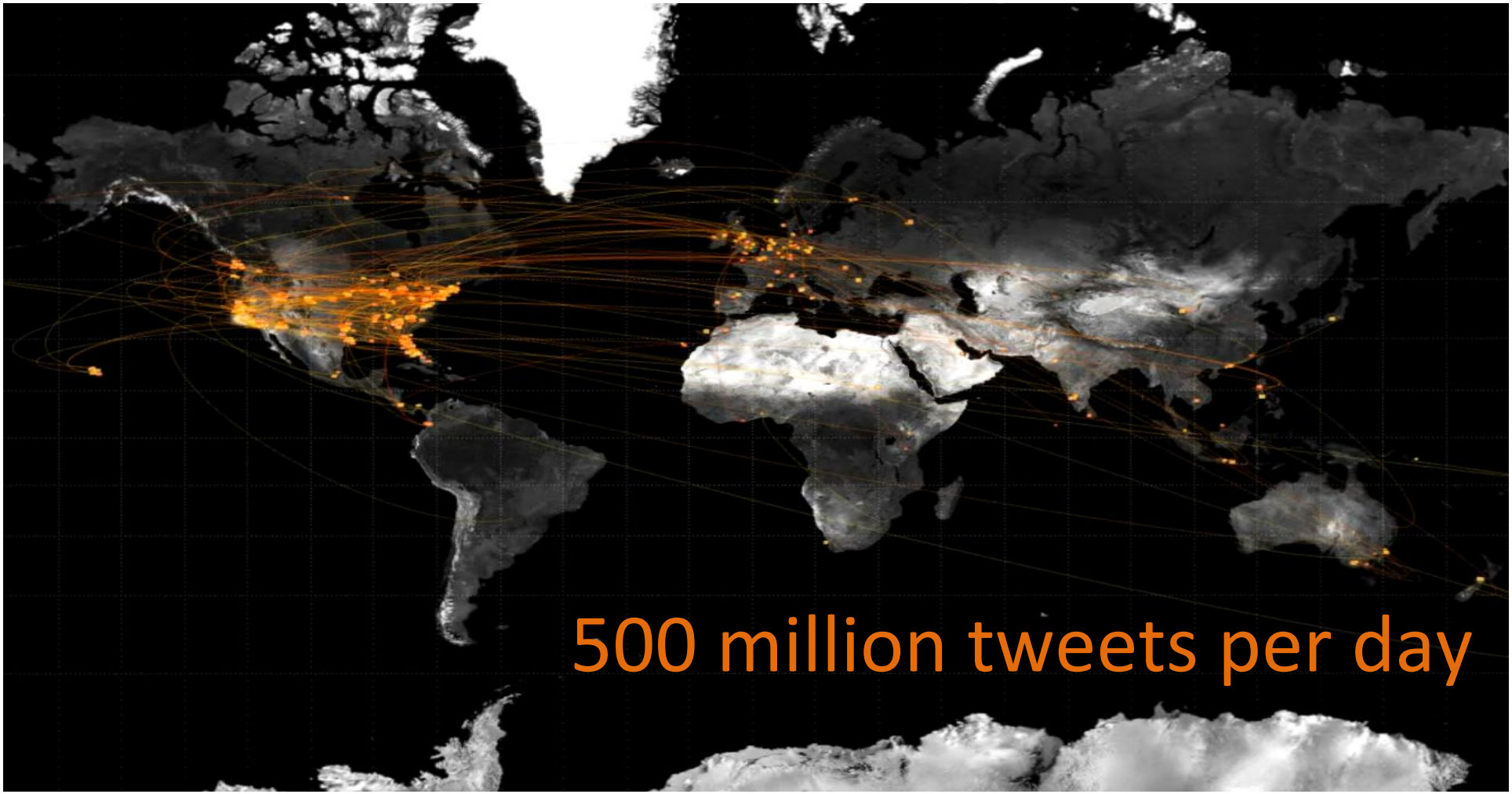
Paper 3: "You are where you tweet: a content-based approach to geo-locating twitter users." Cheng, Zhiyuan, et. al. Proceedings of the 19th ACM international conference on Information and knowledge management. ACM, 2010.

Paper 4: "From tweets to polls: Linking text sentiment to public opinion time series." O'Connor, Brendan, et al. ICWSM 11 (2010): 122-129.

# Geo-locations of Tweets

Paper 3: "You are where you tweet: a content-based approach to geo-locating twitter users." Cheng, Zhiyuan, et. al. Proceedings of the 19th ACM international conference on Information and knowledge management. ACM, 2010.

# The Promise: Twitter as “human” sensing

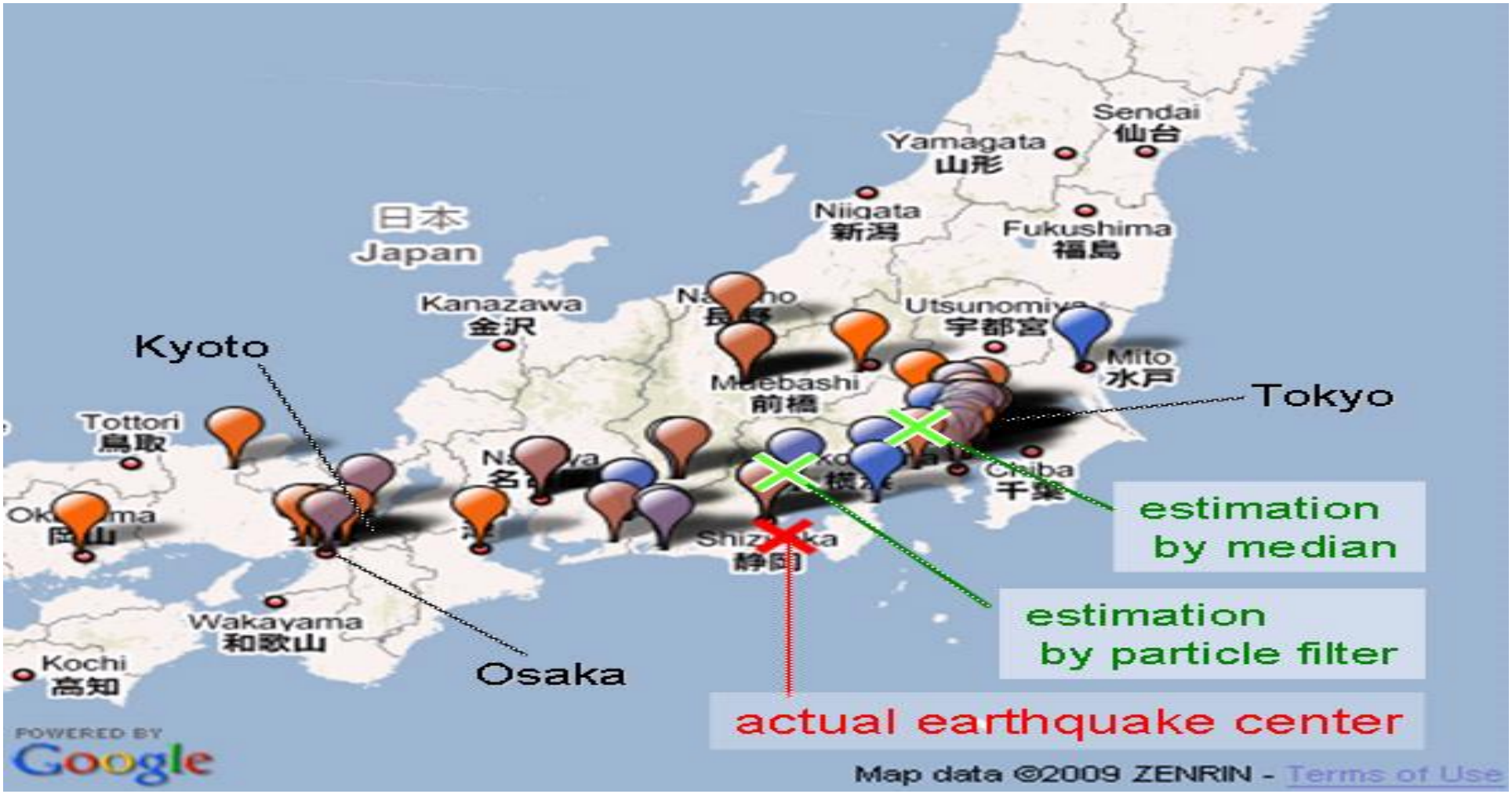


# Question to think

- How would you use the geo-location information of the twitter users or tweets if they are available?
- What are the pros and cons of using such geo-information?

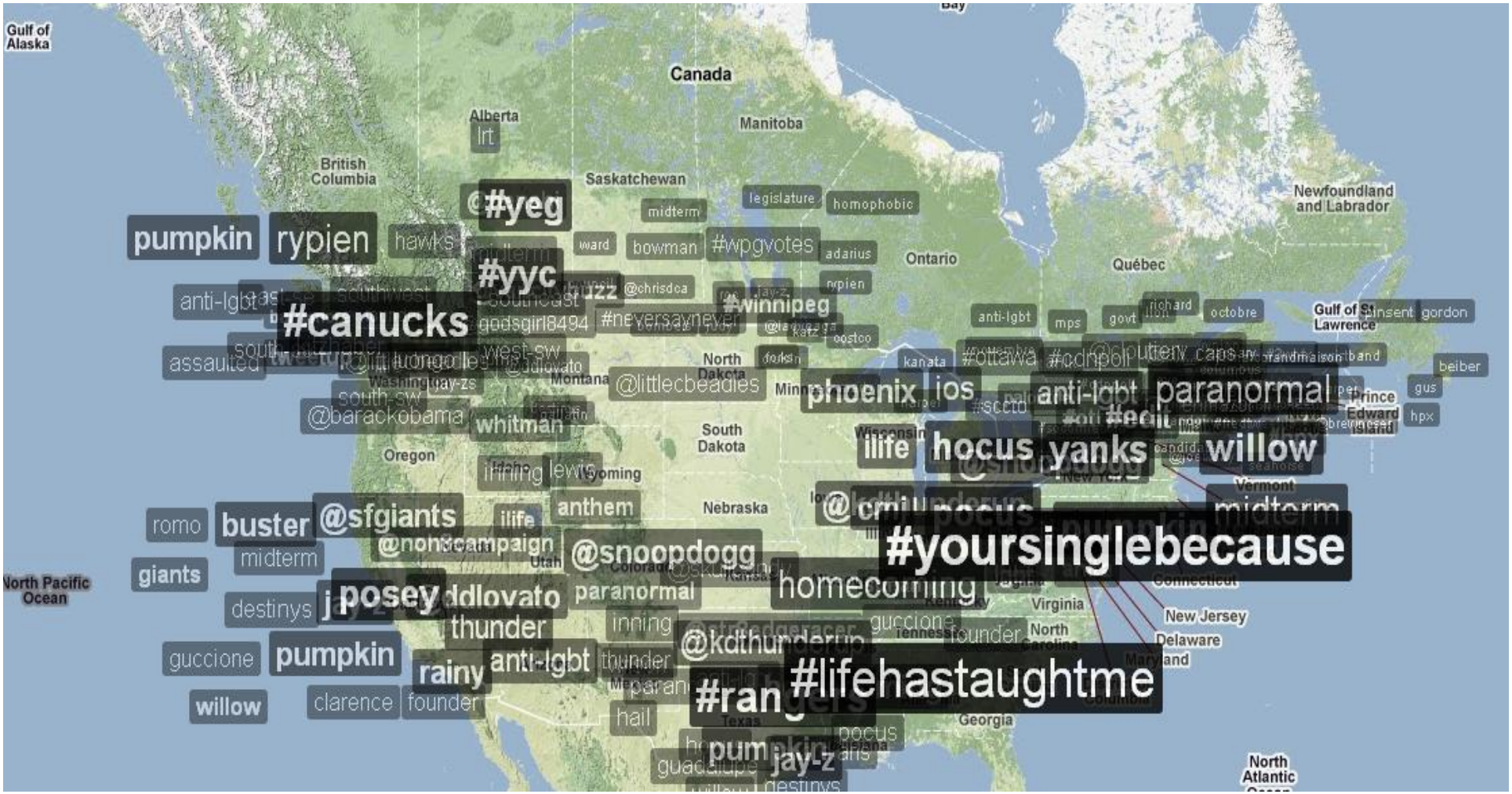
# Example: Earthquake Detection

- Earthquake shakes Twitter users, Sakaki et al, WWW 2010



# Example: Buzz Tracking

- http://trendsmap.com/



# And on and on …

- New personalized location-based information services
  - Local news service based on Twitter
  - Targeting Ads based on user's geo-location
- Emergency management:
  - Project EPIC at UC-Boulder: (http://epic.cs.colorado.edu/)
  - Earthquake and fires detection
- Tracking the diffusion of infectious diseases
  - Google Flue-Trend like service based on Twitter

# But … Location Sparsity is a Severe Problem

- Dataset:
  - Random sampling from public timeline.
  - BFS sampling from 20 random seeds.
  - → 1M user profiles and 30M tweets


- **Only 21% of users** list a location as granular as a **city name**

- **Only 5% of users** list a location as granular as **latitude/longitude** coordinates

- Rest are overly general, missing, or nonsensical
  - "California", "worldwide", "Wonderland" …

# But … Location Sparsity is a Severe Problem


```
{
    "text": "Time for the States to fight back !!!    Tenth Amendment Movement: Taking On the
Feds http://bit.ly/14t1RV   #tcot #teaparty",
    "created_at": "Tue Nov 17 21:08:39 +0000 2009",
    "geo": null,
    "id": 5806348114,
    "in_reply_to_screen_name": null,
    "in_reply_to_status_id": null,

    "user": {
        "screen_name": "TPO_News",
        "created_at": "Fri May 15 04:16:38 +0000 2009",
        "description": "Child of God - Married - Gun carrying NRA Conservative - Right Winger
hard Core Anti Obama (Pro America), Parrothead - www.ABoldStepBack.com #tcot #nra #iPhone",

TPO_Ba
        "protected": false,
        "statuses_count": 21147,
        "location": "Las Vegas, Baby!!",
        "time_zone": "Pacific Time (US & Canada)",
        "url": "http://www.tpo.net/1dollar",
        "utc_offset": -28800,
     }
}
```


**Only 0.42% of tweets** contain **geocodes** (i.e., geo/coordinate field of a tweet)

# Goal: Predict Twitter User Location at a City Level


- Requirements:
  - •**Generalizable** across social media sites and future human-centric sensing systems
  - •Provide **accurate and reliable** location estimation
  - •**No need** for proprietary data from system operators (e.g., backend database) or privacy-sensitive data from users (e.g., IP or user/pass)

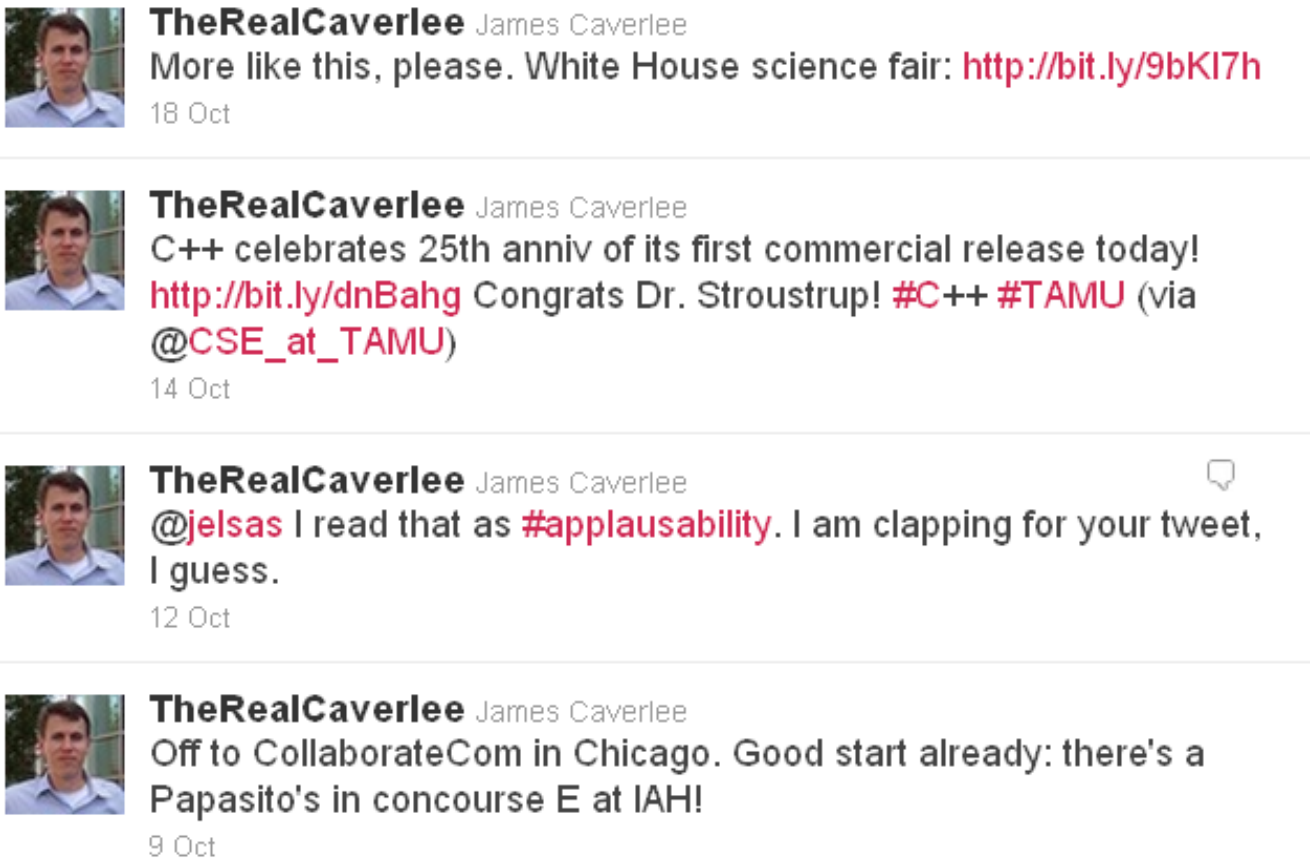
- Approach:
  - •Based purely on **public content posted by user**

# Challenges

- What are the challenges of doing content-based location estimation for Twitter data?

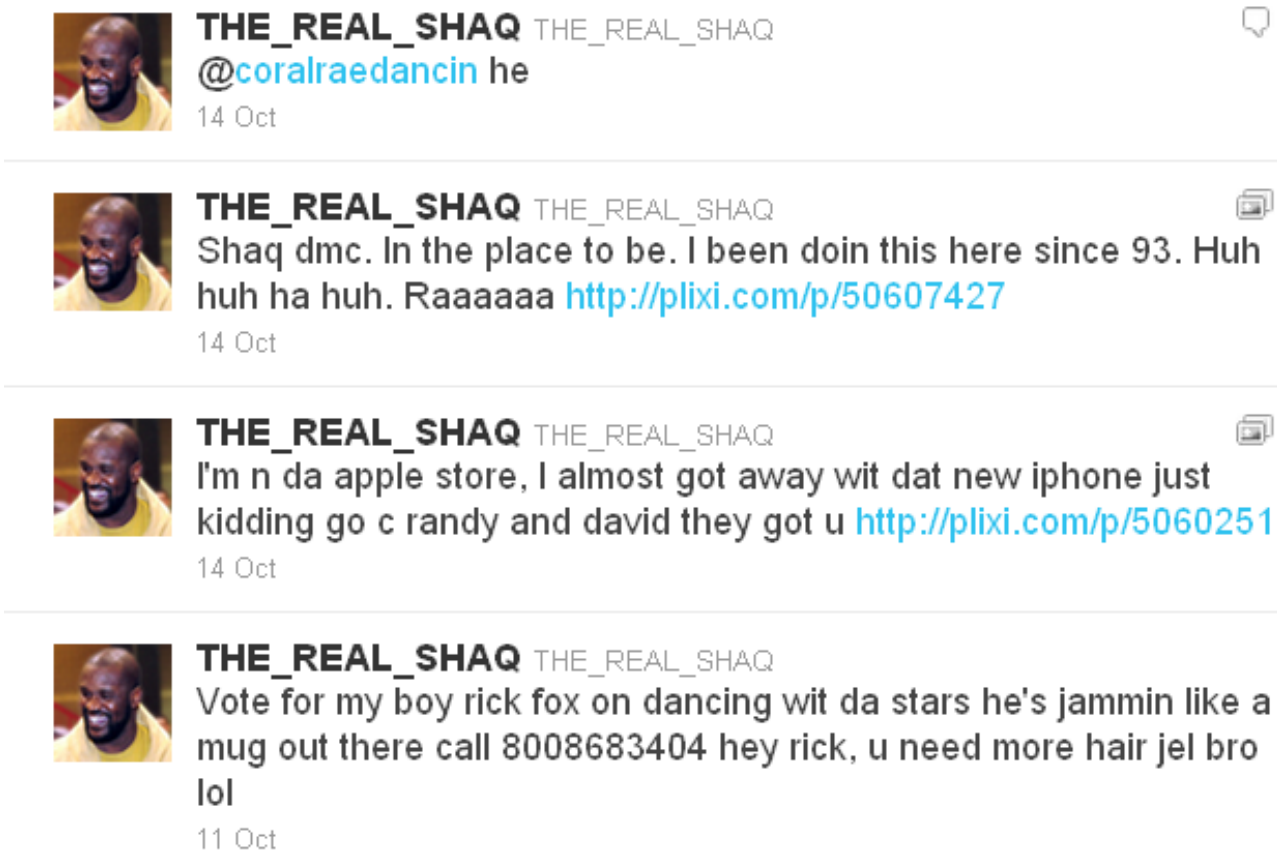
# Content-Based Location Estimation: Challenges

- Tweets are noisy, mixing a variety of daily interests.



# Content-Based Location Estimation: Challenges

- Prevalence of shorthand and non standard vocabulary for informal communication




What?

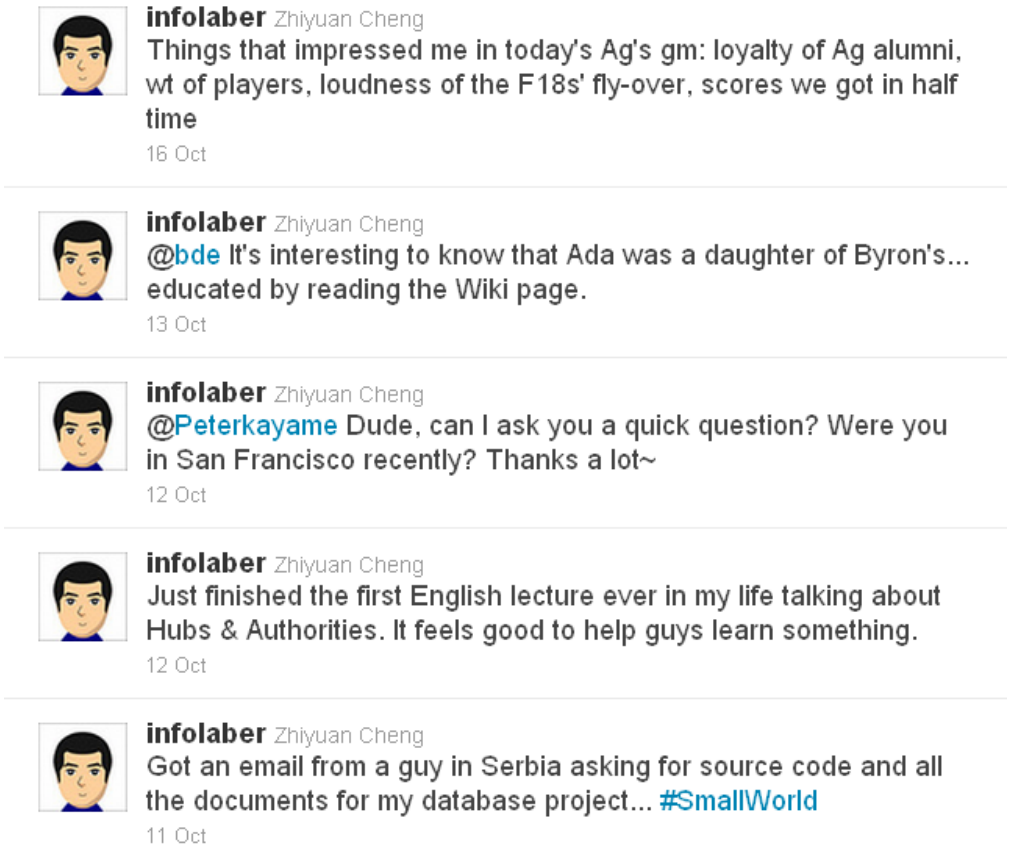
dmc? Huh huh ha huh?
Raaaaaa?

n da? wit dat?
c? u?

jel? bro? lol?

# Content-Based Location Estimation: Challenges

- A user may have interests that span multiple locations beyond their immediate home location




Texas A&M?

San Francisco??

Serbia???

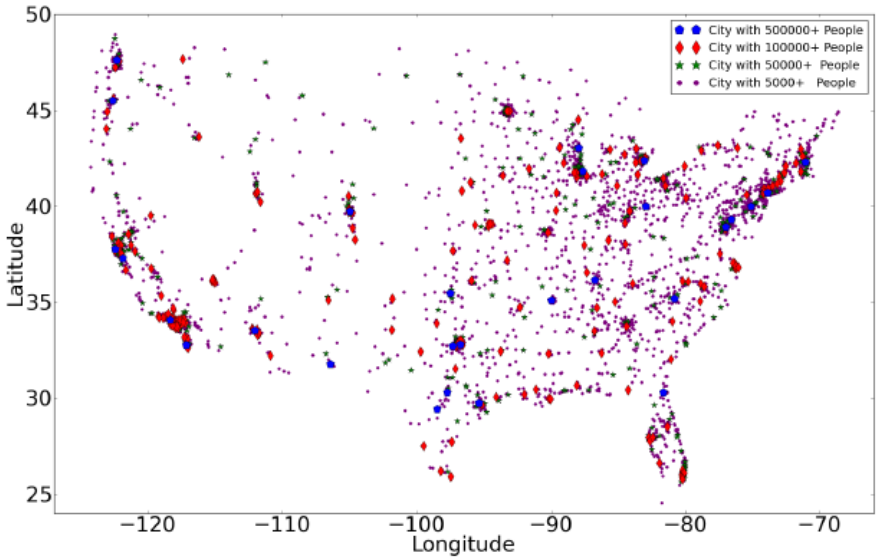
# Content-Based Location Estimation: Challenges

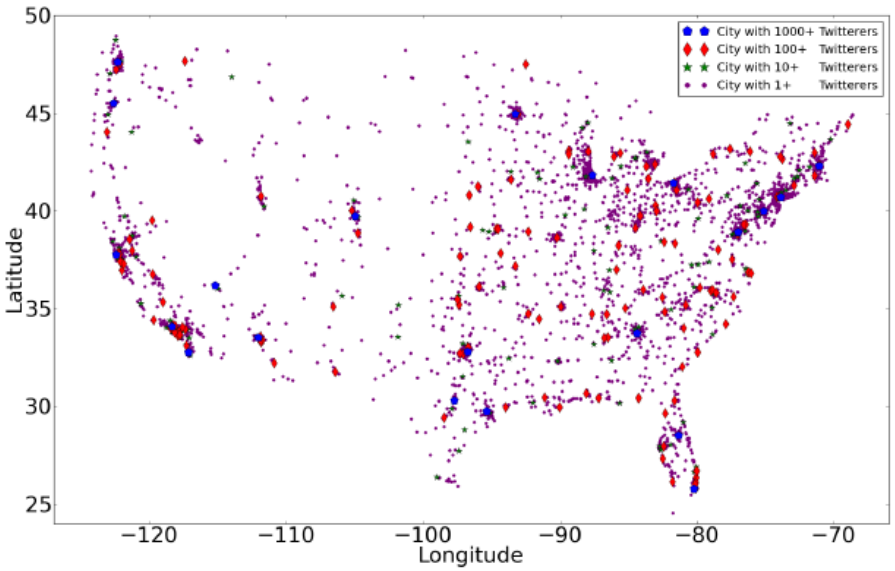
- A user may have more than one natural location, e.g., travel, commute, etc.

# Training Datasets

- Focus on users within the continental U.S.
- Using city names in Census 2000 U.S. Gazetter to filter user profile with valid city names
- For ambiguous city names, only consider "Cityname, StateName" or "Cityname, StateAbbreviation"
  - E.g., 3 city named "Anderson", 6 named "Madison"
- 12% of all sampled users (130,689) have valid and unambiguous city names

# Comparison of US Population and Sample Twitter User Population




US Population ⟷ Sample Twitter User Population

*Match Well!*

*The authors at least consider the sampling bias problem on Twitter.*

# Experimental Setup

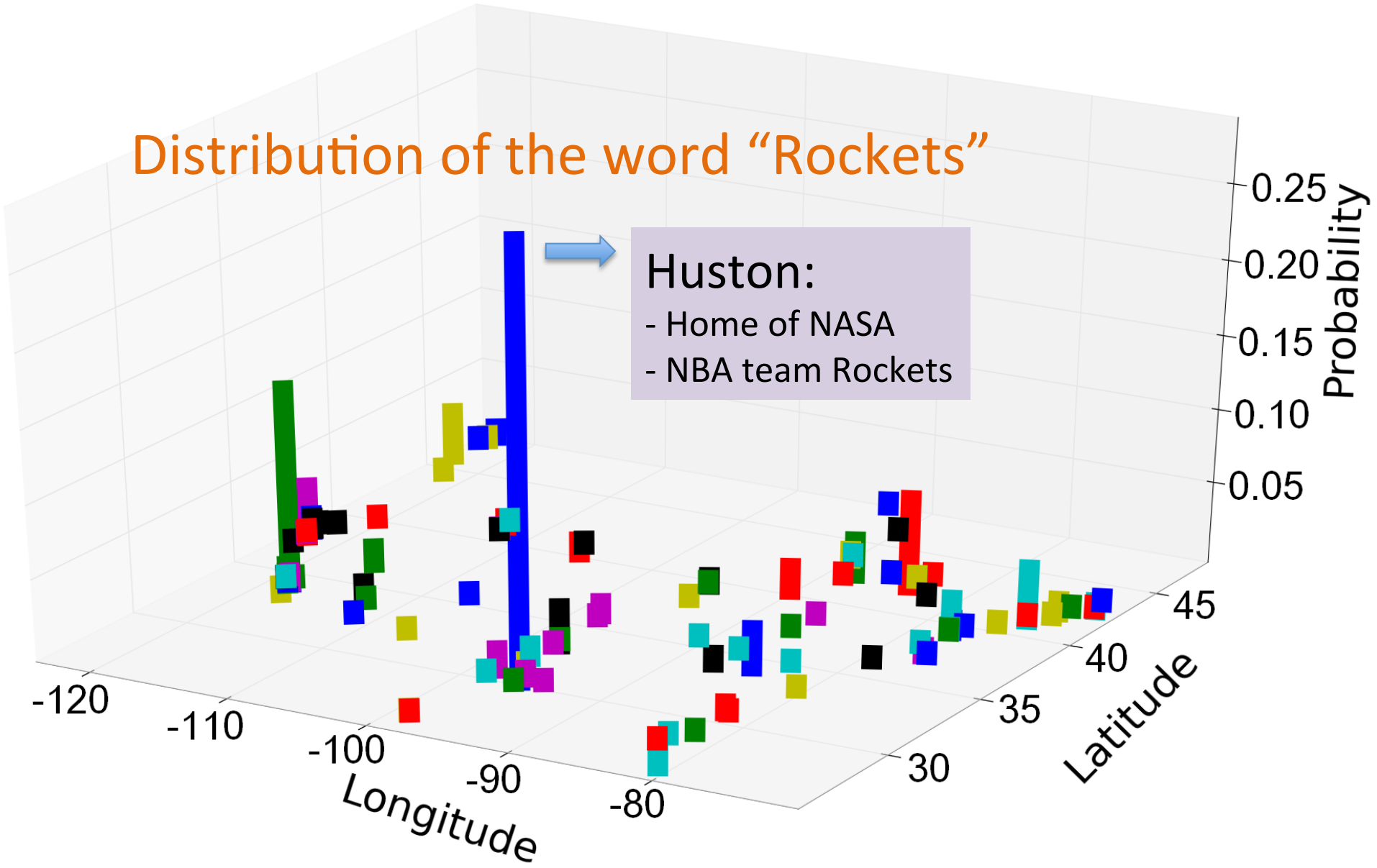
**Location Estimation Problem:** Given a set of tweets $S_{tweets}(u)$ posted by a Twitter user $u$, estimate a user's probability of being located in city $i$: $p(i|S_{tweets}(u))$, such that the city with maximum probability $l_{est}(u)$ is the user's actual location $l_{act}(u)$.

**[Training Set]:** 130K Twitter users with self-labeled city-level location within United States, and 4M tweets sampled from them.
**[Test Set]:** 5K Twitter users with 1K tweets each + lat/long (separate from the training set)

**[Metrics]:**

- **Average Error Distance:** distance in miles between the actual location of the user and the estimated location
- **Accuracy (ACC):** percentage of users with an error distance between 0-100 miles.

# Probability of a Location Given a Word



# Baseline Estimator: Aggregate Over All Words

$$p(i|S_{words}(u)) = \sum_{w \in S_{words}(u)} p(i|w) * p(w)$$

**[Method]:** Given the set of words extracted from user *u*'s tweets, aggregating the probability of city i given individual word *w*, will give us the probability of the user to be located in city i.

# Baseline Location Estimation

- **[Results]:**
  - **Average Error Distance:** 1,773 miles
  - **Accuracy:** 10.12% of users in the test data set are geo-located within 100 miles to the real locations

# Two Key Observations

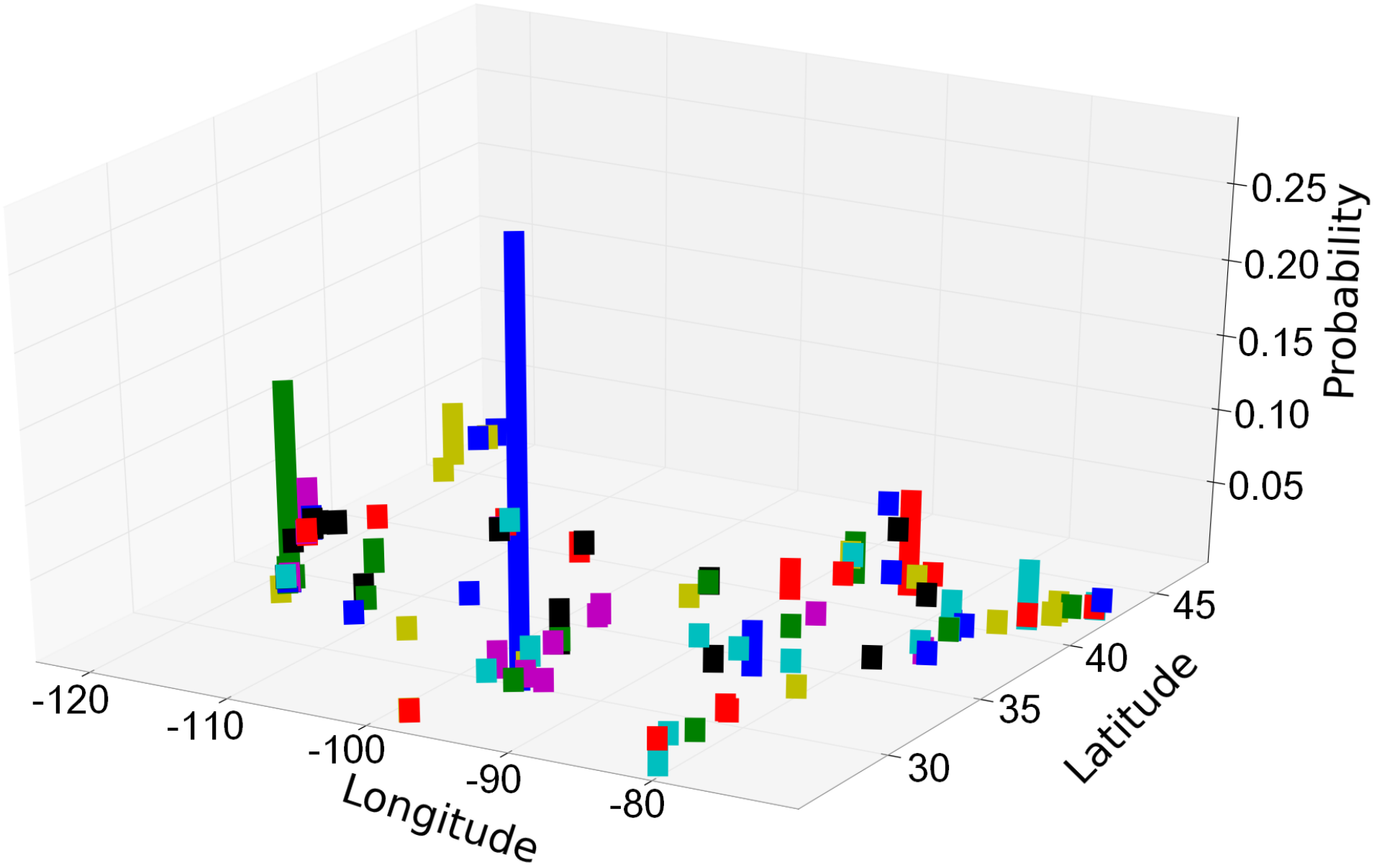
- Observation #1:
  - Most words provide **very little power** at **distinguishing the location** of a user. (e.g., August, peace, world)

  - ➔Need a method to isolate "local words" (e.g., "howdy" is a typical greeting word used in Texas)

- Observation #2:
  - Per-city word distributions for small cities are **under-specified** leading to large estimation errors.

  - ➔Need a method to overcome this sparsity

# How to identify the local words?



# Identifying Local Words in Tweets

- **Local Words**: A high local focus and fast dispersion
- **Non-local Words**: many multiple central points with no clear dispersion
- Q: How do we assess spatial focus and dispersion?

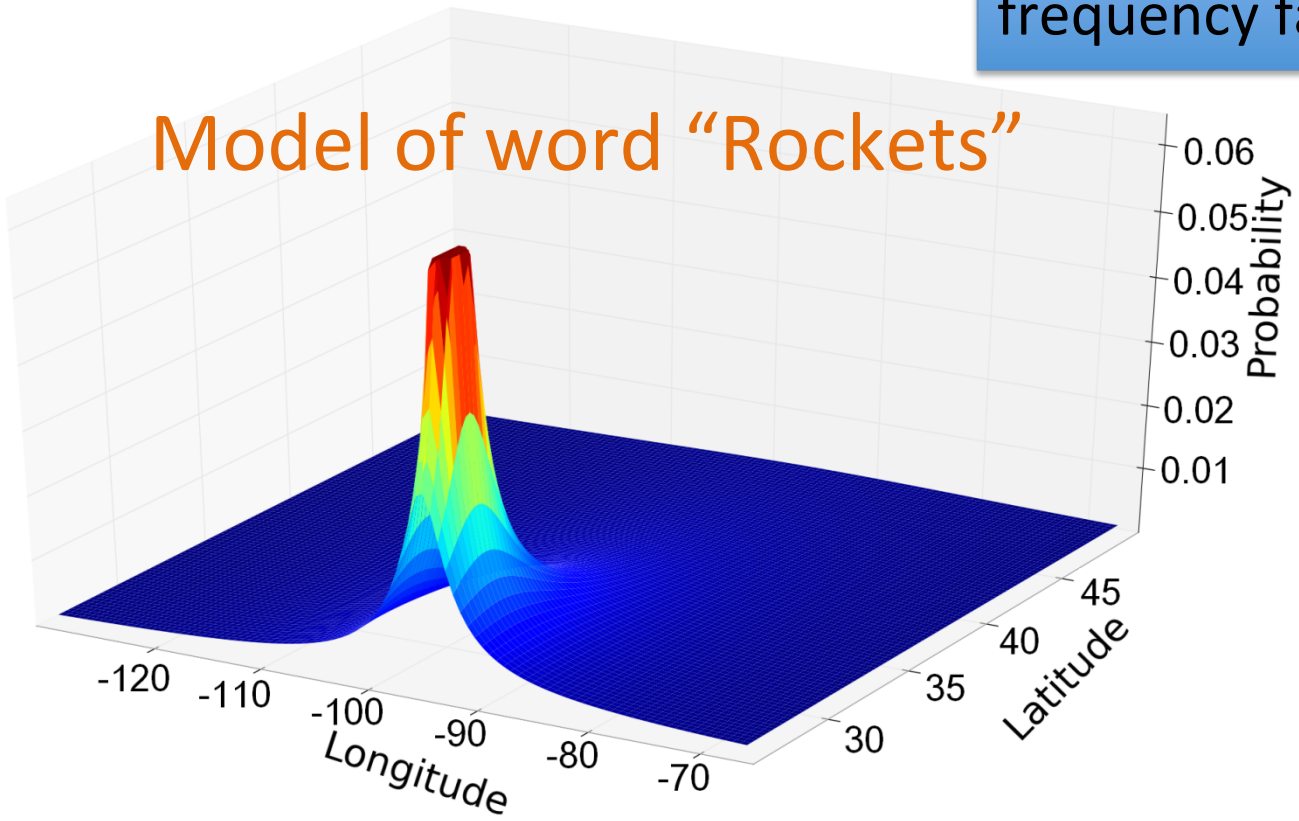
# Identifying Local Words in Tweets

Enlightened by Backstrom's model [Backstrom et al. WWW 08], generate a model for each word according to the observed probabilistic distribution, in the form of:

$$p = Cd^{-a}$$

C: central frequency
a: how fast the frequency falls

Model of word "Rockets"



# Identifying Local Words in Tweets

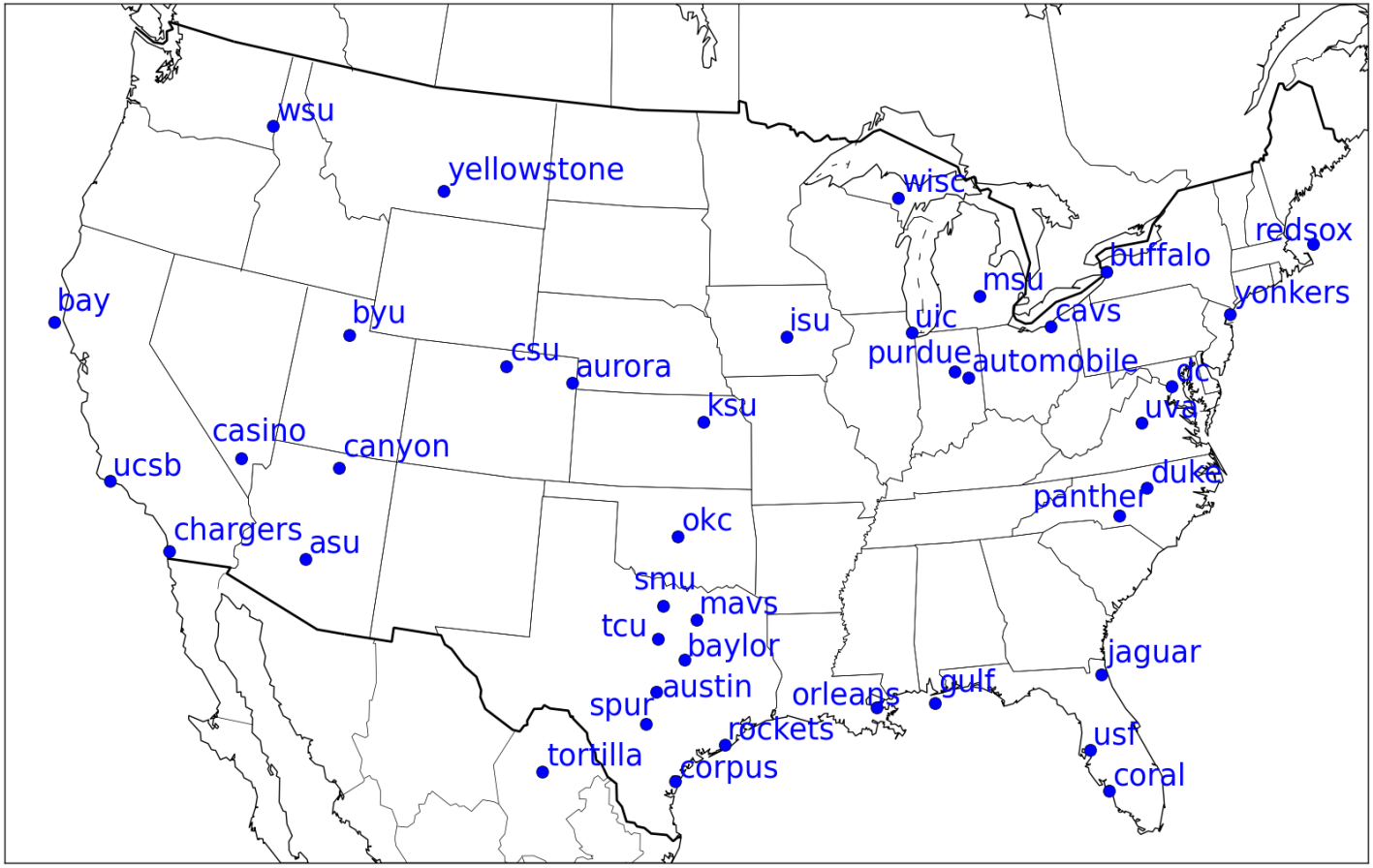
Manual classified some models as local words, and train classifiers with the labeled set to categorize other models. Finally, **3,615 words** are classified as **local words**.

**Table 1: Example Local Words**

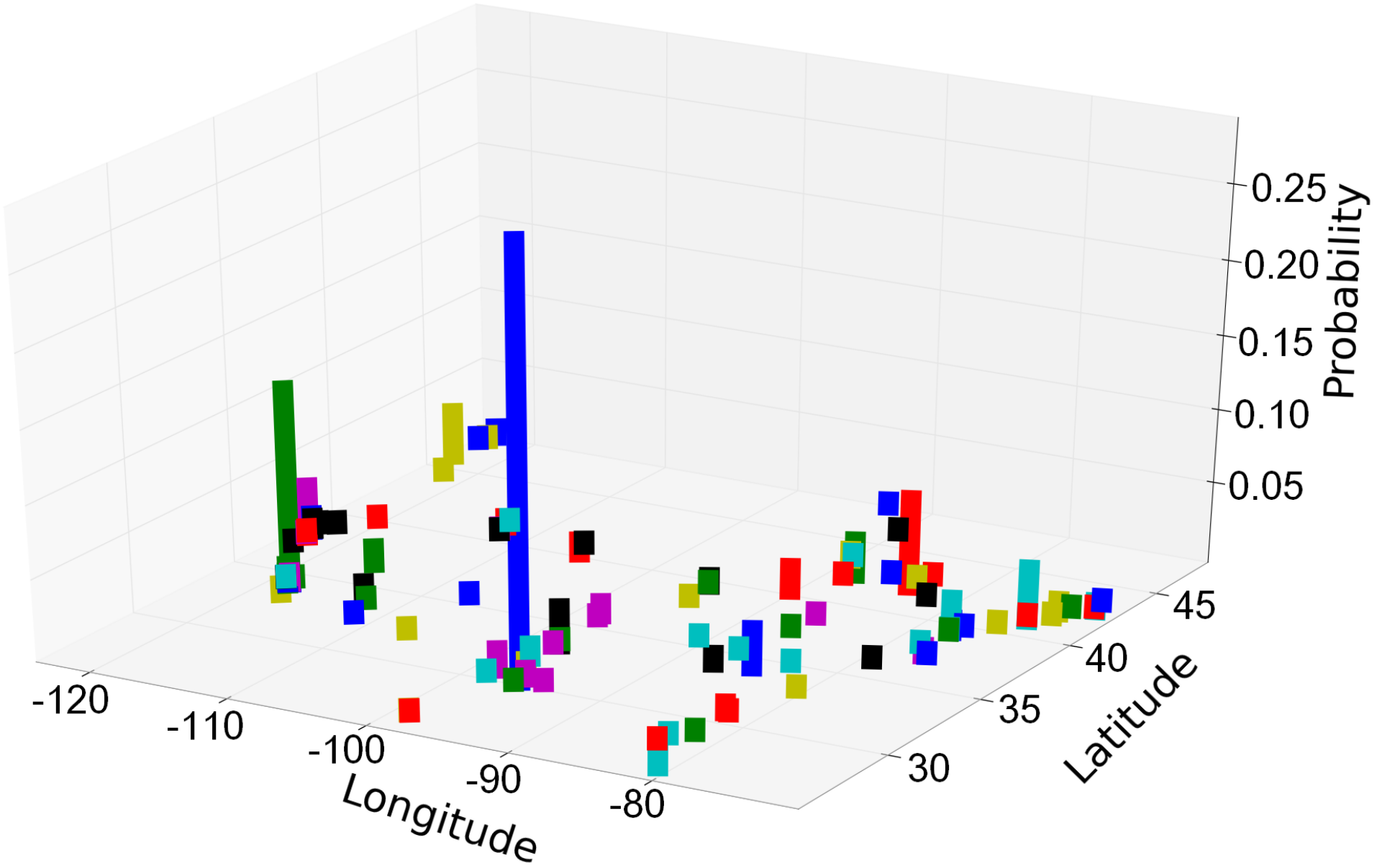
| Word | Latitude | Longitude | $C_0$ | $\alpha$ |
|---|---|---|---|---|
| automobile | Detroit | | 0.5018 | 1.8874 |
| casino | Las Vegas | | 0.9999 | 1.5603 |
| tortilla | Border of Texas and Mexico | | 0.0115 | 1.0350 |
| canyon | Grand Canyon | | 0.2053 | 1.3696 |
| redsox | Boston | | 0.1387 | 1.4516 |

# Identifying Local Words in Tweets

Manual classified some models as local words, and train classifiers with the labeled set to categorize other models. Finally, **3,615 words** are classified as **local words**.



# How to overcome tweet sparsity?



# Overcoming Tweet Sparsity

- **Laplace Smoothing:**

$$p(i|w) = \frac{1 + count(w, i)}{V + N(w)}$$

Ignore geographic information

- **State-Level Smoothing:**

$$p'(i|w) = \lambda * p(i|w) + (1 - \lambda) * \frac{\sum_{i \in S_c} p(i|w)}{|S_c|}$$

Coarse grained: State

- **Lattice-Based Neighborhood Smoothing:**

$$p'(lat|w) = \mu * p(lat|w) + (1.0 - \mu) * \sum_{lat_i \in S_{neighbors}} p(lat_i|w)$$

$$p'(i|w) = \lambda * p(i|w) + (1.0 - \lambda) * p'(lat|w)$$

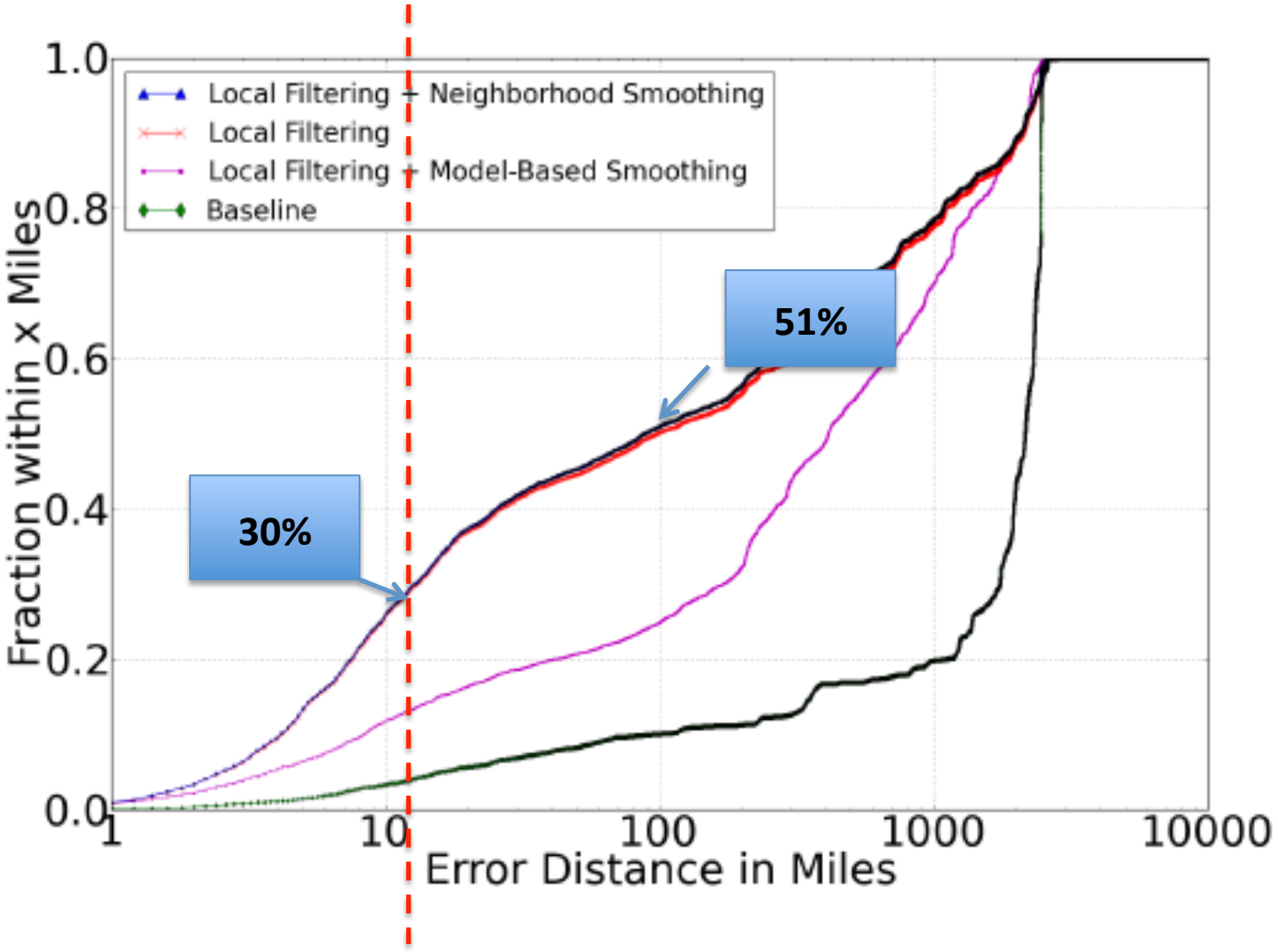
Fine grained: Lattice

# Impact of Refinements

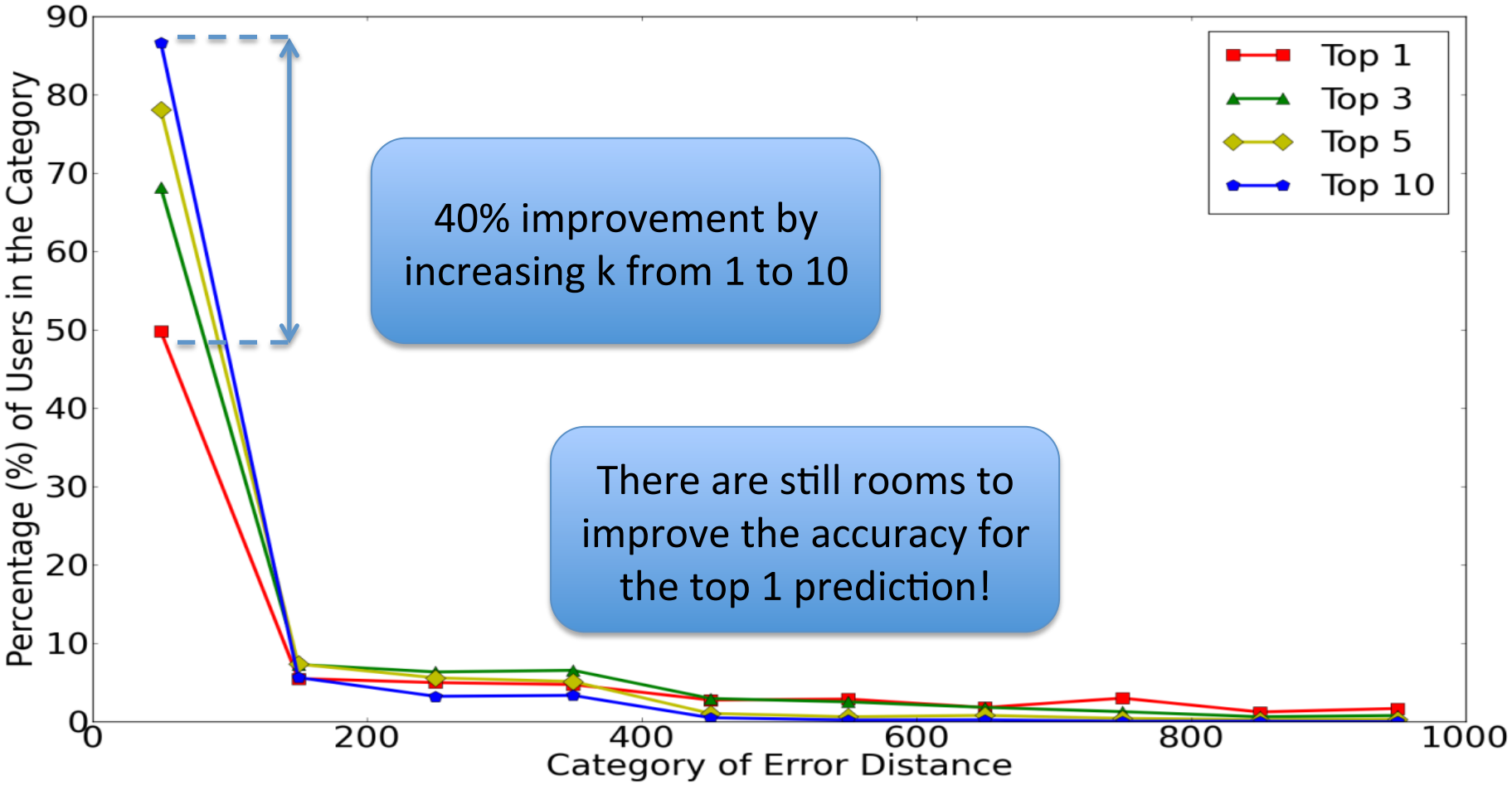
| Method | ACC | AvgErrDist (Miles) |
|---|---|---|
| Baseline | 0.101 | 1773.146 |
| + Local Filtering (LF) | 0.498 | 539.191 |
| + LF + Laplace | 0.480 | 587.551 |
| + LF + State-Level | 0.502 | 551.436 |
| **+ LF + Neighborhood** | **0.510** | **535.564** |
| + LF + Model-based | 0.250 | 719.238 |

- **Key: Local Filtering.**
- **#1 Smoothing Technique: Neighborhood-based smoothing.**

# Comparison Across Estimators

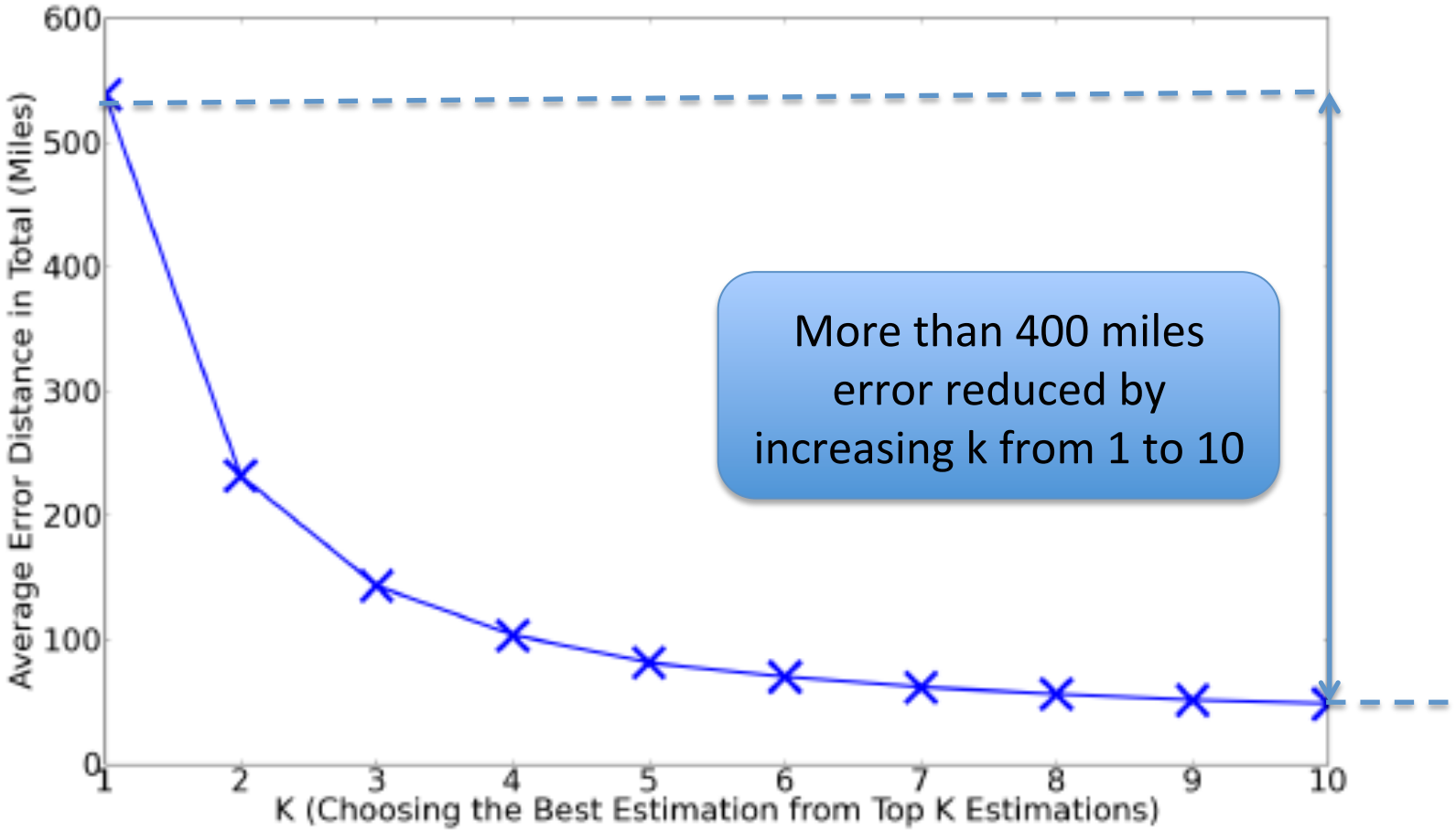


# Capacity of the Location Estimator



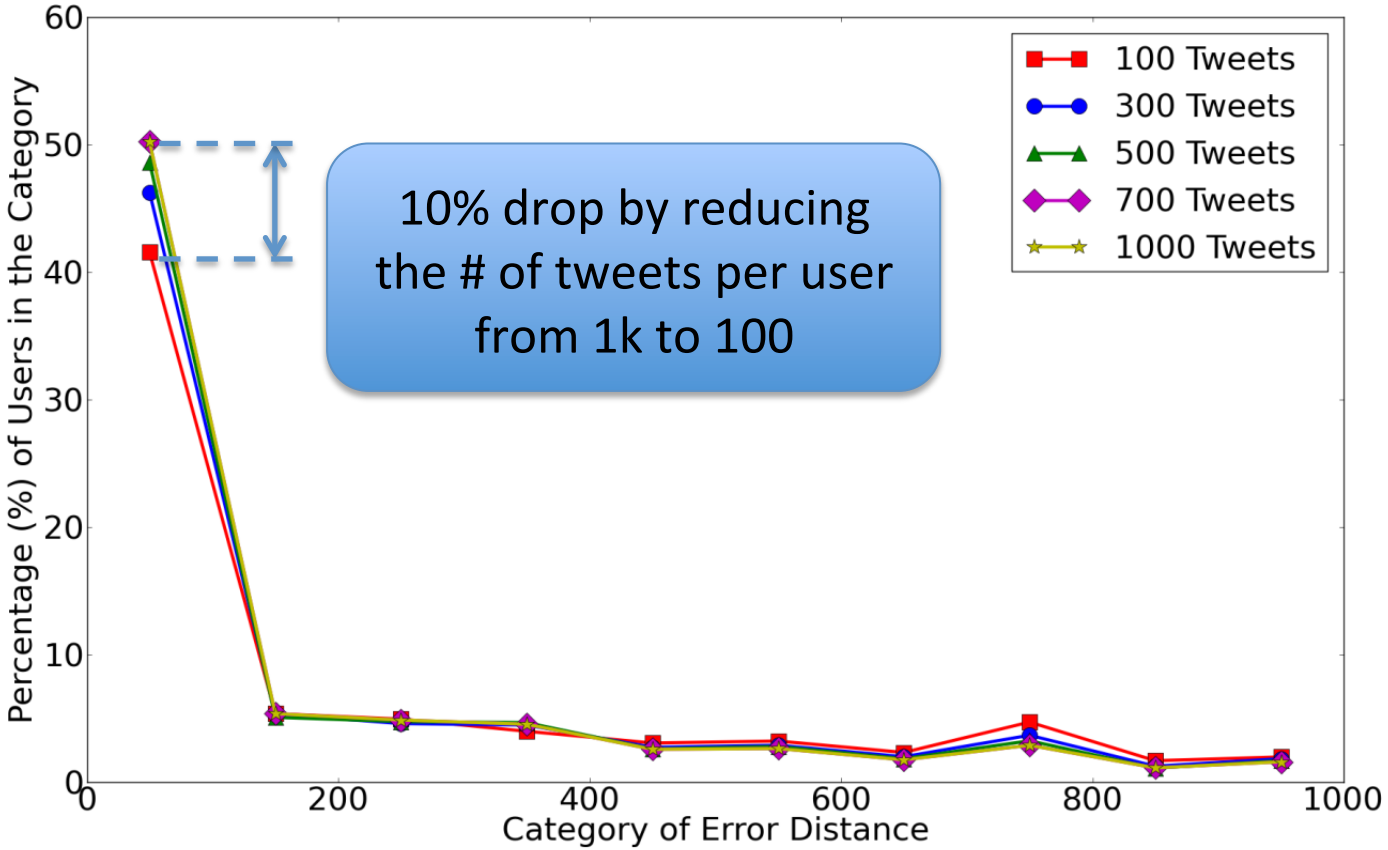

Accuracy is dramatically improved as they increase k, and almost 90% of users are located within 100 miles of their actual locations in the top 10.

# Average Error Distance vs k



# Estimation Quality: Number of Tweets

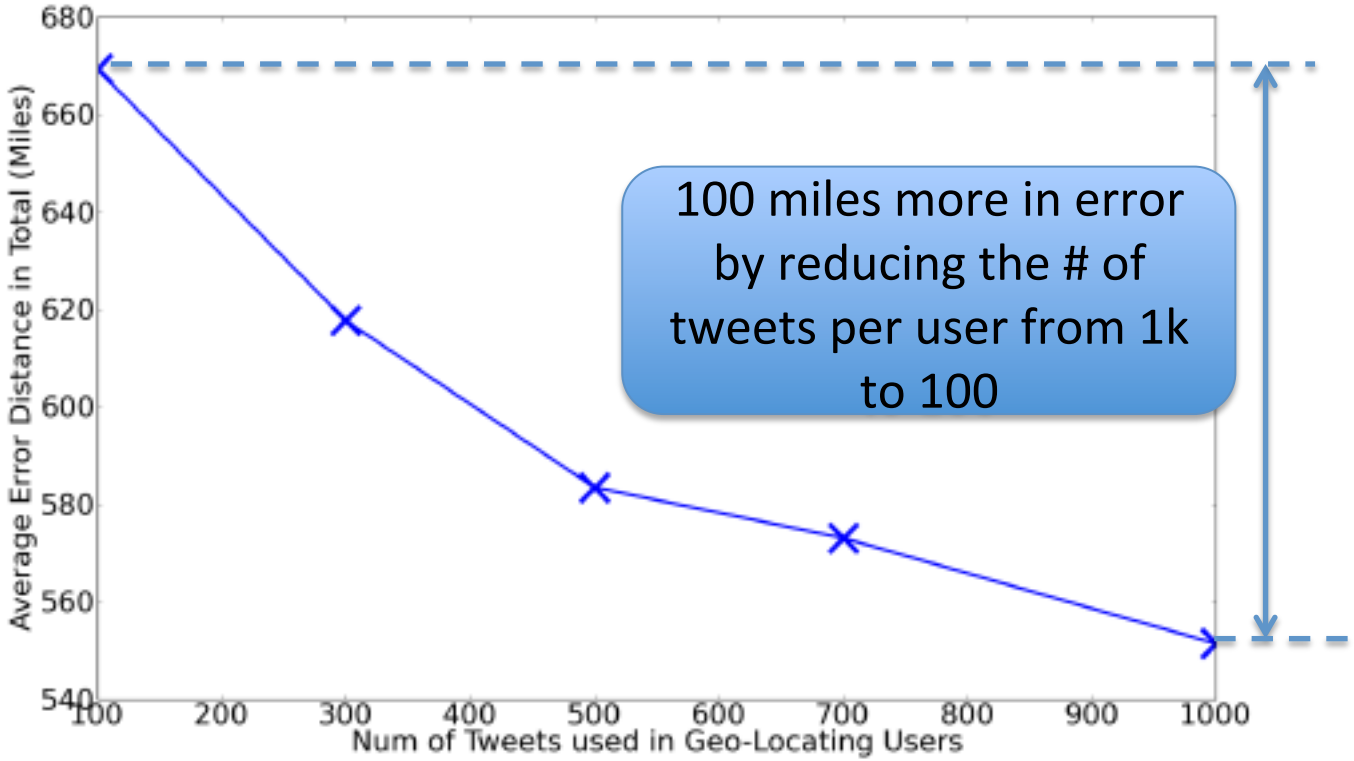
Examine whether they can achieve equally good estimation results using only 100 tweet or a few 100s.




Even with only 100 tweets, 40%+ users are located within 100 miles.

# Estimation Quality: Number of Tweets

Examine whether they can achieve equally good estimation results using only 100 tweet or a few 100s.




(b) Average Error Distance with Different # of Tweets

Even with only 100 tweets, 40%+ users are located within 100 miles.

# Conclusions and Next Steps

- Proposed and evaluated a probabilistic framework for estimating a Twitter user's city-level location based **purely on the content** of the user's tweets.
- Can place **51% of Twitter users within 100 miles** of their actual location.

- What next?
  - More data = better location estimation
  - Incorporate **social ties** into the estimator
  - Explore temporal aspect of location estimation

# Duplicate Detection on Twitter

Paper 4: "Groundhog day: near-duplicate detection on twitter." Tao, Ke, et al. Proceedings of the 22nd international conference on World Wide Web. International World Wide Web Conferences Steering Committee, 2013.

# Groundhog Day


Film released on Feb. 12 1993.

Story: A weather man finds himself in a time loop, repeating the same day again and again.

# Outline

- Search & Retrieval on Twitter
- Duplicate Content on Twitter
- Near-duplicates in Twitter Search
- Solution to Twitter Search: the Twinder Framework
- Analysis & Evaluation
- Conclusion

# Search & Retrieval on Twitter

## How do people use Twitter as a source of information?

- Twitter is more like a news media.
- How do people search on Twitter? [Teevan et al.]
  - Repeated queries & monitoring for new content
- Problems:
  - Short tweets ➔ lots of similar information
  - Few people produce contents ➔ many retweets, copied content

J. Teevan, D. Ramage, and M. R. Morris. #TwitterSearch: A Comparison of Microblog Search and Web Search. In Proceedings of the 4th International Conference on Web Search and Web Data Mining (WSDM), 2011.

# Duplicate Content on Twitter (1/3)

## Classification of near-duplicates in 5 levels

- **Exact copy**
  - Completely identical in terms of characters.

t1: Huge New Toyota Recall Includes 245,000 Lexus GS, IS Sedans - http://newzfor.me/?cuye

t2: Huge New Toyota Recall Includes 245,000 Lexus GS, IS Sedans - http://newzfor.me/?cuye

- **Nearly exact copy**
  - Completely identical except for #hashtags, URLs or @mentions

t3: Huge New Toyota Recall Includes 245,000 Lexus GS, IS Sedans - http://bit.ly/ibUoJs

# Duplicate Content on Twitter (2/3)

## Classification of near-duplicates in 5 levels

- **Strong near-duplicate**
  - Same core message, one tweet contains more information.

t4: Toyota recalls 1.7 million vehicles for fuel leaks: **Toyota's latest recalls are mostly in Japan, but they also... http://bit.ly/dH0Pmw**

t5: Toyota Recalls 1.7 Million Vehicles For Fuel Leaks **http://bit.ly/flWFWU**

- **Weak near-duplicate**
  - Same core message, one tweet contains personal views.
  - Convey semantically the same message with differing information nuggets.

t6: The White Stripes broke up. **Oh well.**

t7: The White Stripes broke up. **That's a bummer for me.**

# Duplicate Content on Twitter (3/3)

## Classification of near-duplicates in 5 levels

- **Low overlap**
  - Semantically contain the same core message, but only have a few words in common

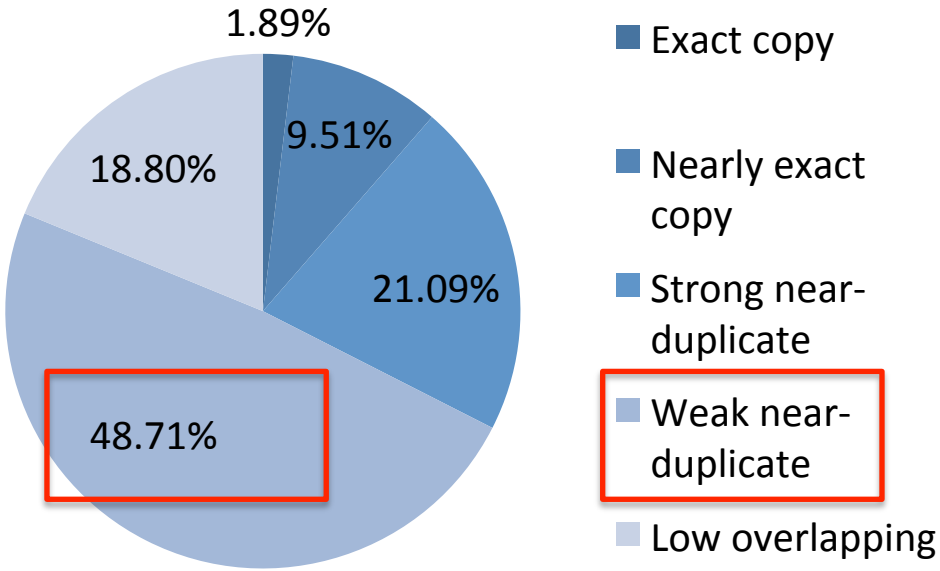
t8: **Federal** Judge **rules** Obamacare **is** unconsitutional**...**

t9: **Our man of the hour:** Judge **Vinson gave** Obamacare **its second** unconstitutional **ruling. http://fb.me/zQsChak9**

# Near-Duplicates in Twitter Search (1/2)

## Analysis of the Tweets2011 corpus

- For the 49 topics (queries), 2,825 topic-tweet pairs are relevant, 57 tweets/topic
- They manually labeled **55,362 tweet pairs**
- They found **2,745 pairs of duplicates** in different levels.




*People like to tweet their personal opinions.*

# Near-Duplicates in Twitter Search (2/2)

## Motivation

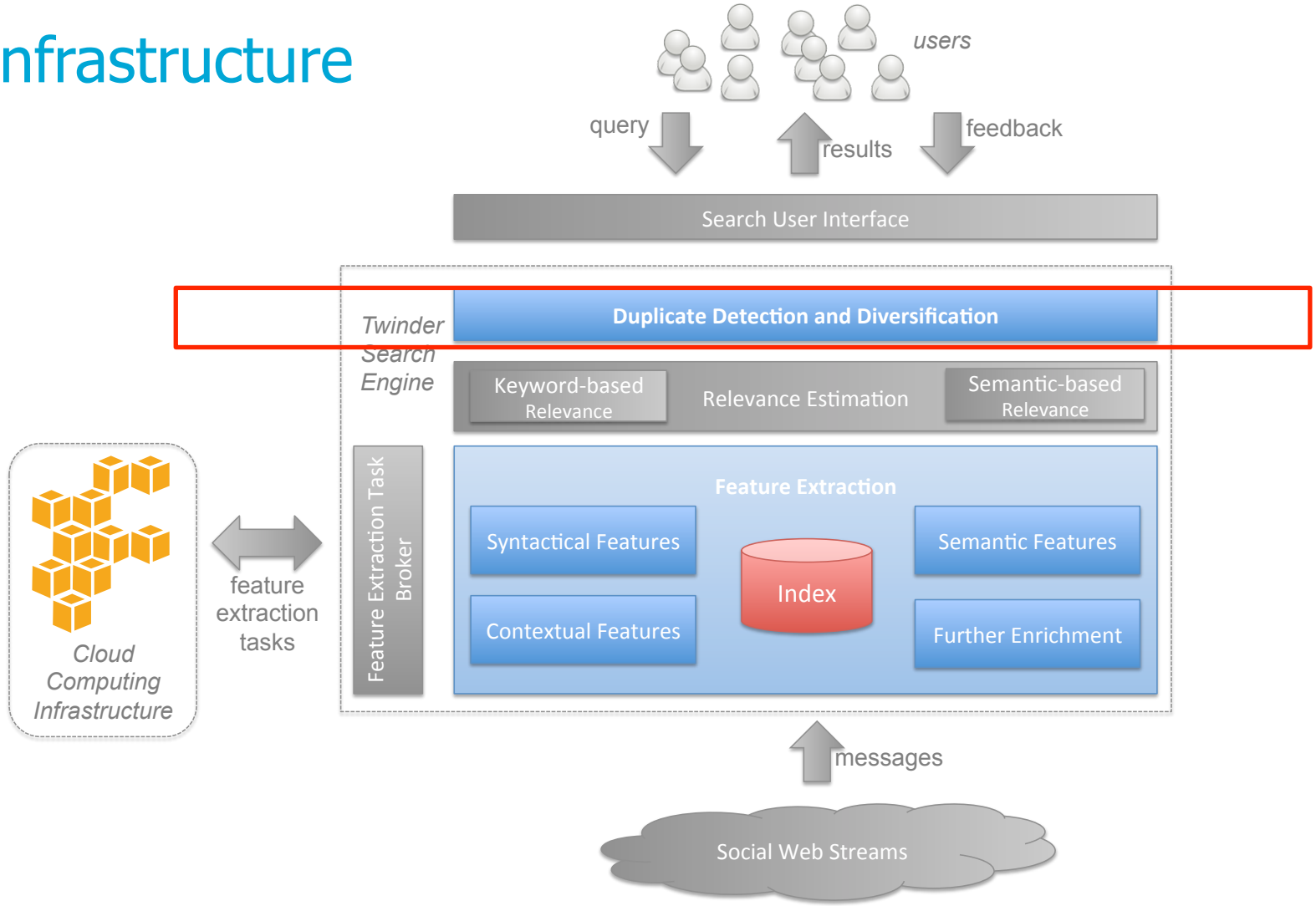
- For each of the 49 topics, rank the tweets according to their relevance to the topic (using previous work)

- On average, they found around **20%** duplicates in the search results.

| Range | Top 10 | Top 20 | Top 50 | All |
|---|---|---|---|---|
| Duplicate % | 19.4% | 22.2% | 22.5% | 22.3% |

# Twinder Framework

## Search Infrastructure



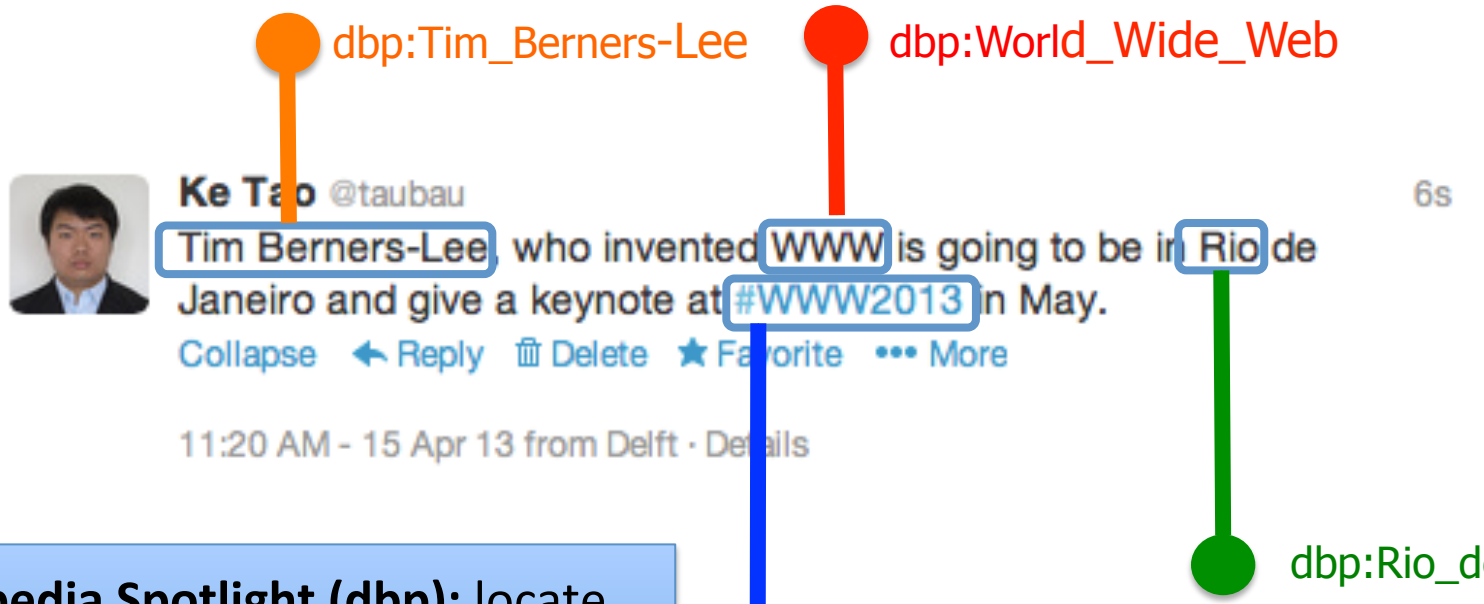
# Building a Classifier … (1/5)

## Overview of the **syntactic** features

| Features | Description |
|---|---|
| Levenshtein distance | Number of characters required to change (substitution, insertion, deletion) one tweet to the other |
| Overlap in **terms** | Jaccard similarity between two sets of words in tweets. |
| Overlap in **#hashtags** | Jaccard similarity between two sets of #hashtags in tweets. |
| Overlap in **URL** | Jaccard similarity between two sets of URLs in tweets. |
| Overlap in expanded URL | Recomputed “Overlap in URL” after expanding shortened URLs in both tweets. |
| Length difference | The difference in length between two tweets. |

# Building a Classifier … (2/5)

Extract **semantics** from tweets (using information extraction tool to extract **entities**)




**Dbpedia Spotlight (dbp):** locate and classify elements in text into pre-defined categories such as the names of persons, organizations, locations,, etc.

# Building a Classifier … (3/5)

## Overview of the **semantic** features

**WordNet**: a large lexical database of English. Nouns, verbs, adjectives and adverbs are grouped into sets of cognitive **synonyms (synsets)**, each expressing a distinct concept.
http://wordnet.princeton.edu/

**Examples:**

-Fire, flame, blaze

-Explosion, blast, blow up

# Building a Classifier … (3/5)

## Overview of the **semantic** features

| Features | Description |
|---|---|
| Overlap in Entities | Jaccard similarity between two sets of **entities** extracted from two tweets |
| Overlap in Entities Types | Jaccard similarity between two sets of **types of entities** from two tweets |
| Overlap in Topics | Jaccard similarity between two sets of detected **topics** in two tweets |
| Overlap in WordNet Concepts | Jaccard similarity between two sets of WordNet Nouns in tweets |
| Overlap in WordNet Synset Concepts | Recomputed Overlap in WordNet Concepts after Combining interlinked Concepts in Synsets |
| WordNet similarity | The similarity calculated based on semantic relatedness* between concepts from two tweets |

# Building a Classifier … (4/5)

## **Enriched** semantic features

- Integrate content from external resources and construct the same set of semantic features

**t3: Huge New Toyota Recall Includes 245,000 Lexus GS, IS Sedans - http://bit.ly/ibUoJs**

# Building a Classifier … (5/5)

## Overview of **contextual** features

| Features | Description |
|---|---|
| Temporal difference | The difference of posting time of two tweets |
| Difference in **#followees** | The difference in number of **followees** of the author of the tweets |
| Difference in **#followers** | The difference in number of **followers** of the author of the tweets |
| Same client | Indicator of whether the two tweets are posted from the same client application |

# Summary of Features

- What feature categories do they have?
  - **Sy**ntactical features (6)
  - **Se**mantic features (6)
  - **En**riched semantic features (6)
  - **Co**ntextual features (4)
- Classification strategies → different feature combinations
  - Dependent on available resources and time constraints

# Classification Strategies

Using **different sets of features** for near-duplicate detection on Twitter

| Strategy | Description |
|---|---|
| **Baseline** | Based on Levenshstein distance |
| **Sy** | Only Syntactical features |
| **SySe** | Add semantics from tweets |
| **SyCo** | Without Semantics |
| **SySeCo** | Without Enriched Semantics |
| **SySeEn** | Without Contextual features |
| **SySeEnCo** | **All Feature included** |

# Analysis and Evaluation

- Research Questions:

  R1: How accurately can the **different duplicate detection strategies** identify duplicates?

  R2: What kind of **features are of particular importance** for duplicate detection?

  R3: How does the accuracy vary for **the different levels** of duplicates?

  R4: How do the duplicate detection strategies **impact search effectiveness on Twitter**?

# Data set: Tweets2011

TREC 2011 Microblog Track

- Twitter corpus
  - 16 million tweets (Jan. 24th, 2011 – Feb. 8th)
  - 4,766,901 tweets classified as English
  - 6.2 million entity-extractions (140k distinct entities)
- Relevance judgments
  - 49 topics
  - 40,855 (topic, tweet) pairs, 2,825 judged as relevant
  - 57.65 relevant tweets per topic (on average)
- Duplicate level labeling
  - 55,362 tweet pairs labeled
  - 2,745 labeled as duplicates (in 5 levels)
  - **Publicly available at http://wis.ewi.tudelft.nl/duptweet/**

# Classification Accuracy

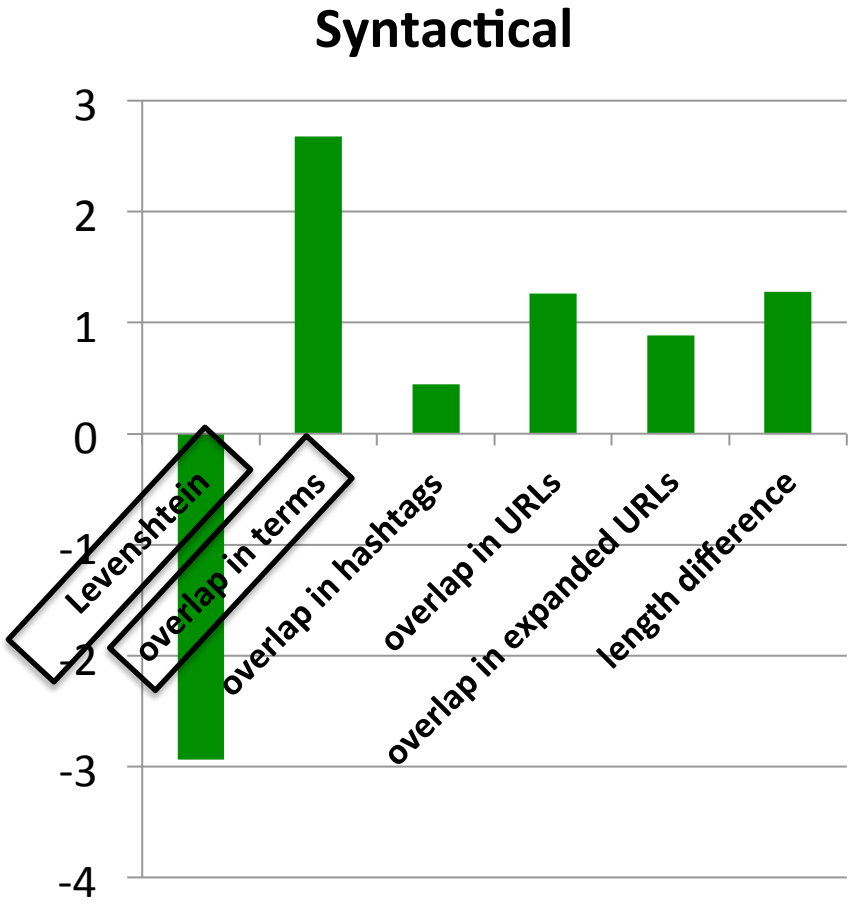
## Duplicate or not? → RQ1

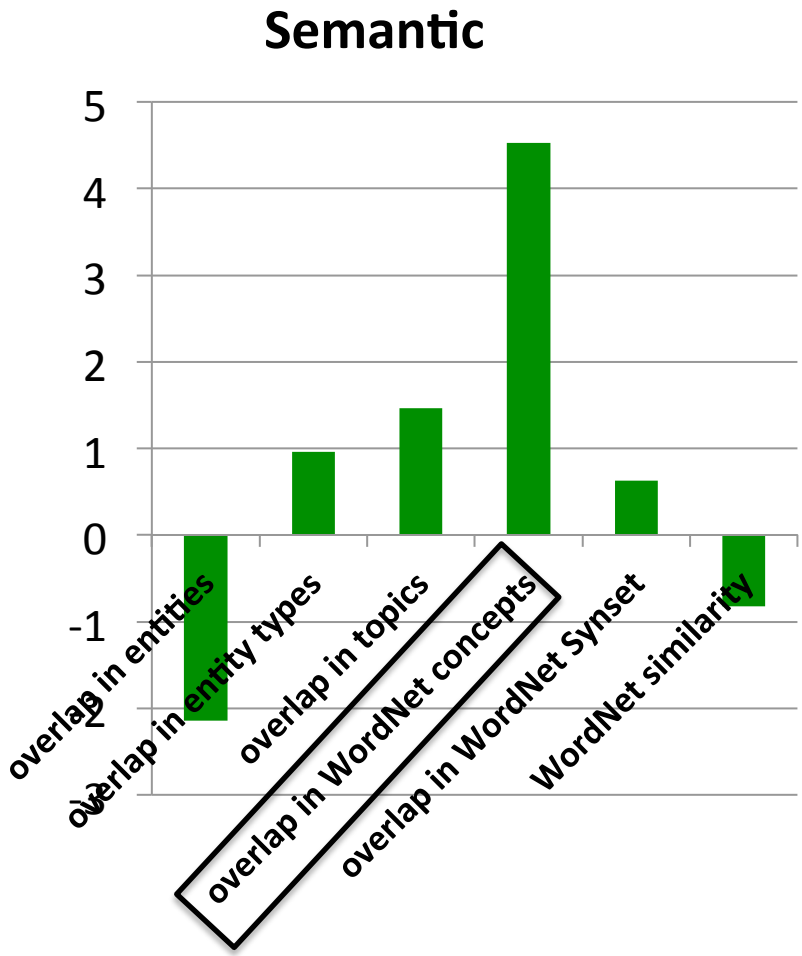
| Features | Precision | Recall | F-measure |
|---|---|---|---|
| Baseline | 0.5068 | 0.1913 | 0.2777 |
| Sy | 0.5982 | 0.2918 | 0.3923 |
| SyCo | 0.5127 | 0.337**0** | 0.4067 |
| SySe | **0.5333** | 0.3679 | 0.4354 |
| SySeEn | 0.5297 | 0.3767 | 0.4403 |
| SySeCo | 0.4816 | 0.4200 | 0.4487 |
| SySeEnCo | **0.4868** | **0.4299** | **0.4566** |

**Overall, they can achieve a precision and recall of about 49% and 43% respectively by applying all possible features.**

# Feature Weights (1/2)

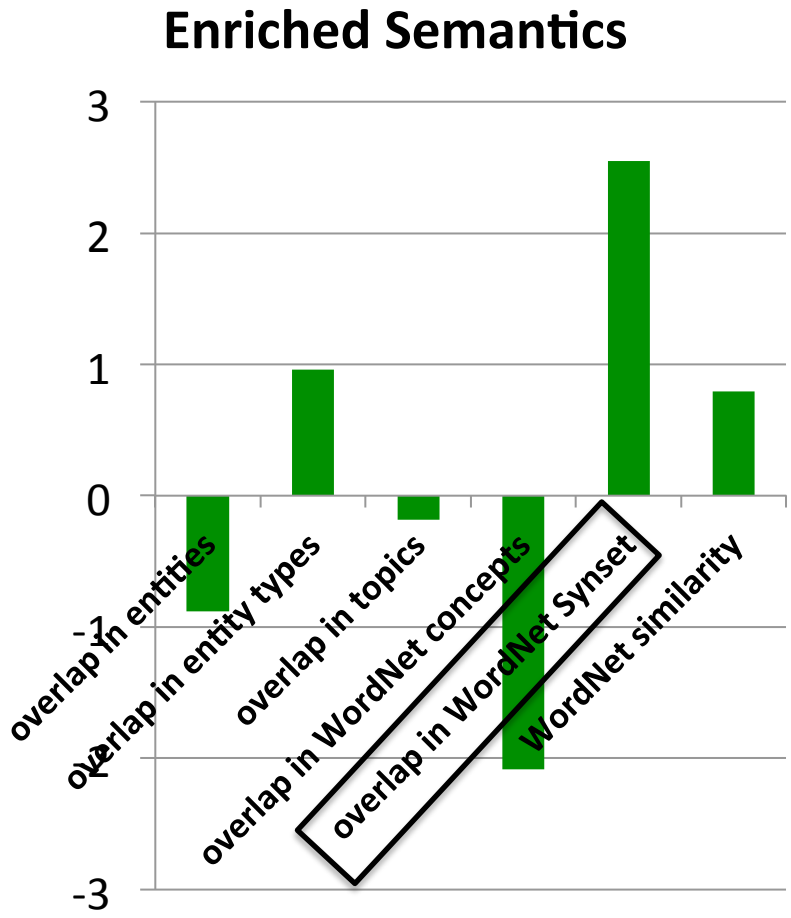
## Which features matter the most? → RQ2

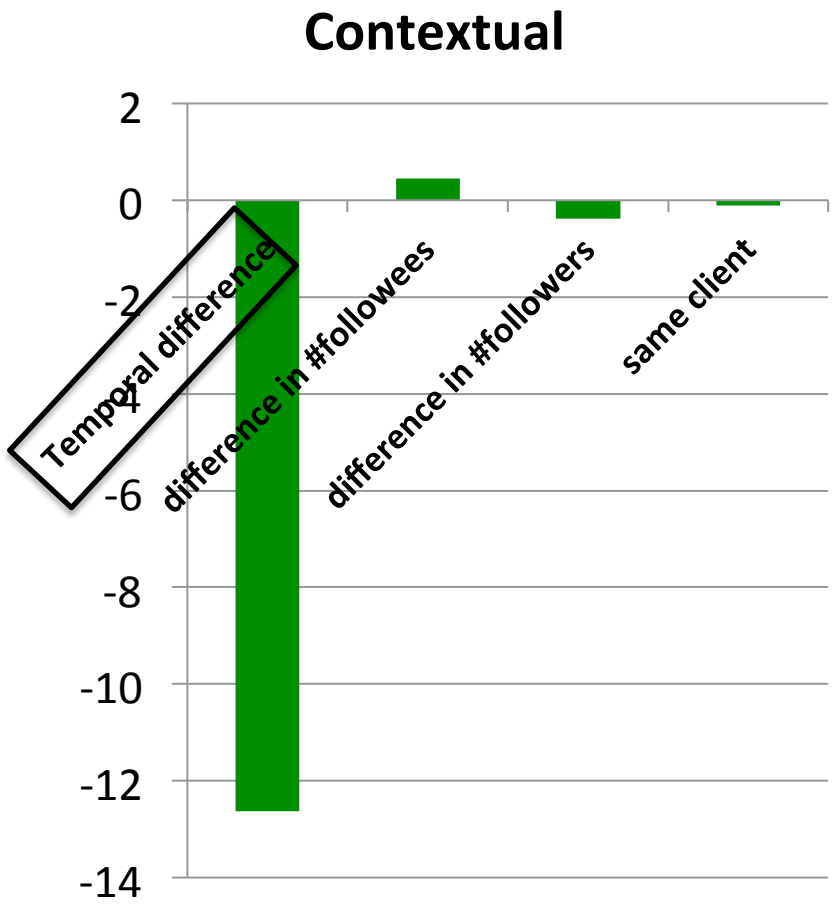


# Feature Weights (2/2)

## Which features matter the most? → RQ2



# Results for Predicting Duplicate Levels (1/2)

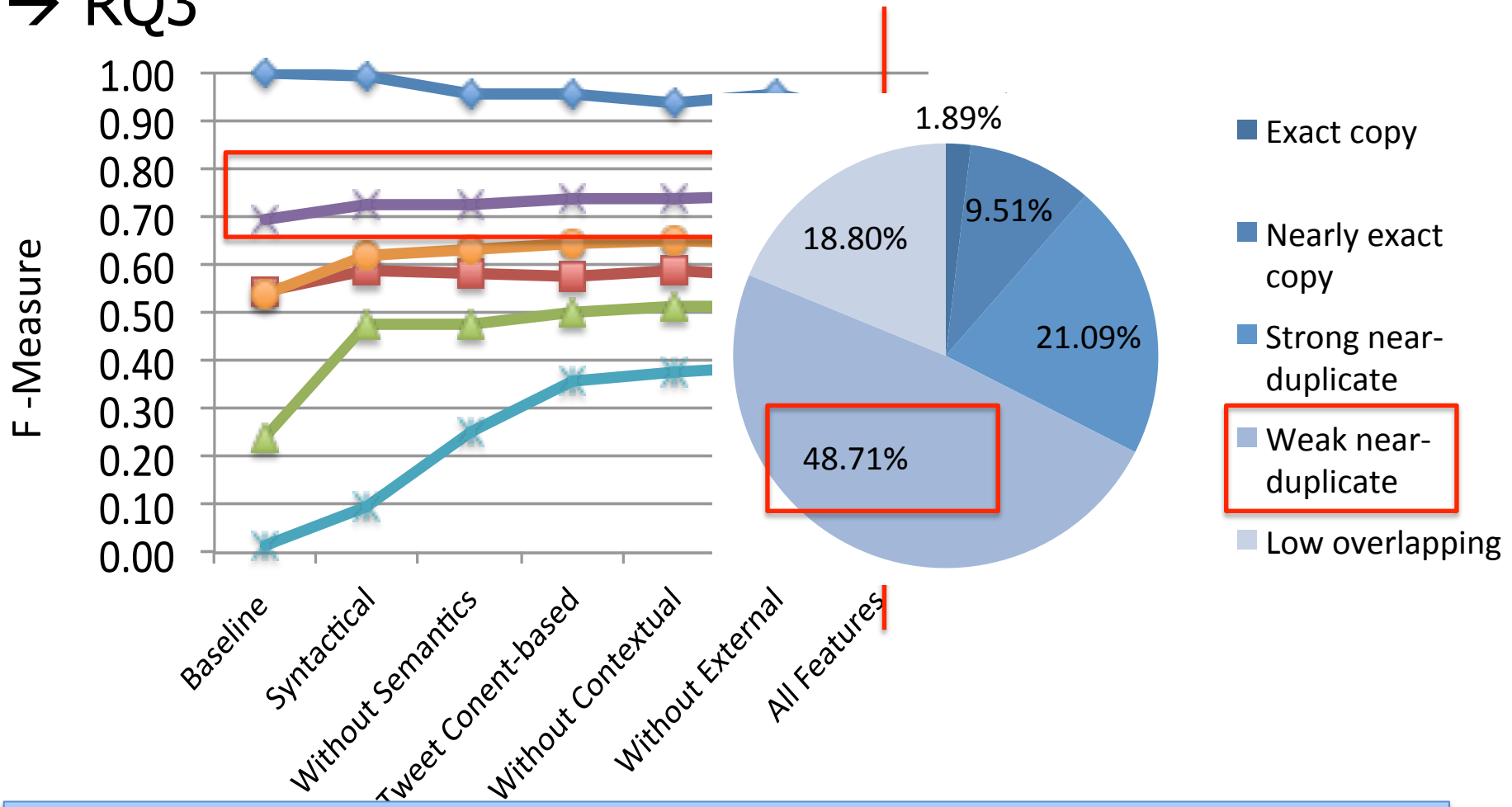
Exact copy, weak near-duplicate, … or low overlap? → RQ3

| Features | Precision | Recall | F-measure |
|---|---|---|---|
| Baseline | 0.5553 | 0.5208 | 0.5375 |
| Sy | 0.6599 | 0.5809 | 0.6179 |
| SyCo | 0.6747 | 0.5889 | 0.6289 |
| SySe | 0.6708 | 0.6151 | 0.6417 |
| SySeEn | 0.6694 | 0.6241 | 0.6460 |
| SySeCo | 0.6852 | 0.6198 | 0.6508 |
| SySeEnCo | **0.6739** | **0.6308** | **0.6516** |

**Overall, they achieve a precision and recall of about 67% and 63% respectively by applying all features.**

# Results for Predicting Duplicate Levels (2/2)

## Exact copy, weak near-duplicate, ... or low overlap?
## → RQ3




Q: What information can you extract from this figure?

# Search Result Diversification

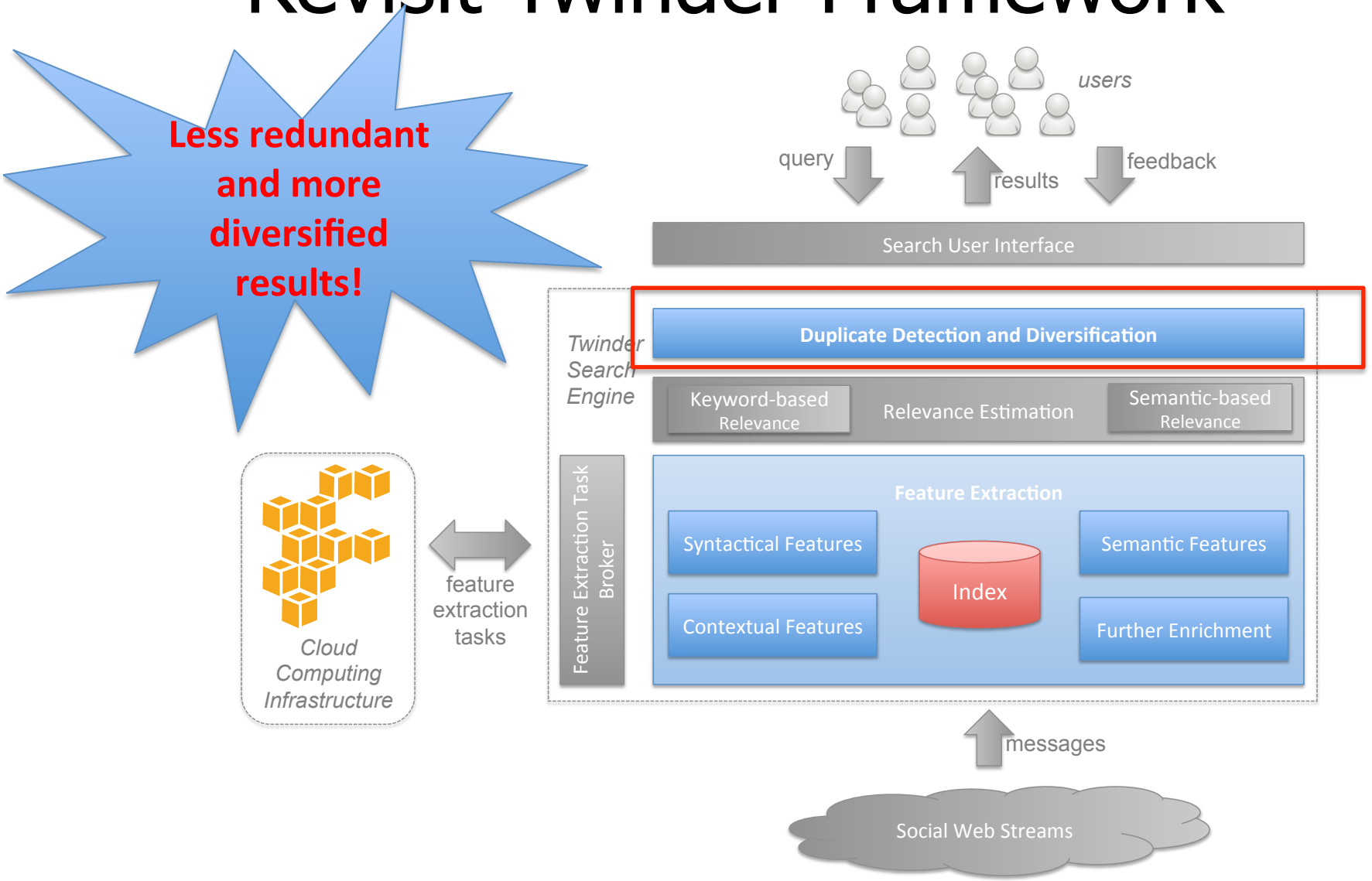
How much redundancy can they detect and remove? → RQ4

- A core application of near-duplicate detection strategies is the diversification of search results. They simply remove the duplicates that are identified by our method.

- Near-duplicates after filtering:

| Range | Top10 | Top20 | Top50 | All |
|---|---|---|---|---|
| **Baseline** | 19.4% | 22.2% | 22.5% | 22.3% |
| **After Filtering** | 9.1% | 10.5% | 12.0% | 12.1% |
| **Improvement** | +53.1% | +52.0% | +46.7% | +45.7% |

# Revisit Twinder Framework



# Conclusions

1. Conduct an **analysis of duplicate content** in Twitter search results and **infer a model for categorizing** different levels of duplicity.
2. Develop a **near-duplicate detection framework** for microposts that provides functionality for analyzing **4 categories of features**.
3. Given duplicate detection framework, the paper performs **extensive evaluations** and **analysis of different duplicate detection strategies** on a large, standardized Twitter corpus to investigate the quality of (i) detecting duplicates and (ii) categorizing the duplicity level of two tweets.
4. The proposed approach enables **search result diversification** and analyzes the impact of the diversification on the search quality.

- The progress on **Twinder** on be found at:

  http://wis.ewi.tudelft.nl/twinder/