# Comparing vector-based and ACT-R memory models of tag retrieval

# User-customized tag prediction on Twitter and StackOverflow

**Clayton Stanley**
Department of Psychology
Rice University
Houston, TX
clayton.stanley@rice.edu
http://chil.rice.edu

THE COMPUTER-HUMAN
INTERACTION LABORATORY AT RICE UNIVERSITY

# Overview

- Motivation for research
- Motivation for task
- General approach
- Twitter and StackOverflow descriptions
- Model descriptions
- Methods
  - Exp. 1: How does past tagging history influence future behavior?
  - Exp. 2: Incorporate history into vector-based model
  - Exp. 3: Incorporate word order into Bayesian model
  - Exp. 4: Compare contextual components of each model
  - Exp. 5: Compare full models
- Conclusions

# Motivation for Cognitive Modeling

- Recently become possible to evaluate psychological theories of declarative memory on large-scale real-world tasks
  - Theories can now be tested on hundreds of millions to billions of data points
  - Now possible due to
    - Growth of social media and user created content
    - Publicly accessible large-scale datasets of user created content
    - Improved APIs and methods for extracting information
    - Improved data mining software and faster hardware

# Motivation for Cognitive Modeling

- These information rich large-scale environments provide unique opportunities for research in declarative memory
  - Can stress test and explore the impacts of the psychological constraints of each theory on a much larger scale than has previously been possible
  - Rapidly test and evaluate different architectural constraints and their impact on retrieval accuracy
  - Begin to test the declarative memory equations on a scale that is closer to the magnitude of chunks that are stored in human memory

# Motivation for Task

- Social media sites are composed almost entirely of human-generated content
- There's a lot of it (over 400M Twitter tweets created per day)
- Users are searching for fresh information (real-time search)
- In addition to actively searching, they are subscribing to information streams
  - Followers (Twitter), friends (Facebook), tags (StackOverflow)
- To support a user on these sites, how can we quickly and effectively connect users to the streams of content that they care about?

# General Approach

- Try to figure out a user's goals on the site
- Identifying goals is easier for social media sites compared to traditional search
  - Social media sites have each user's past behavior
  - Users also choose "tags" when creating content
- Seems plausible that these user-chosen tags contain important information concerning the goals of each user on a site
- Frame the task of choosing a tag as a memory retrieval problem
  - Test how well two declarative memory theories can predict the correct tags
- If we can predict the chosen tags, then we can better predict the goals
  - And use this to recommend to them new information streams that match their interests

# Research Questions

- What kinds of cognitively plausible models can predict user chosen tags on social media sites?
  - Look at StackOverflow and Twitter
  - Compare two state-of-the-art declarative memory retrieval models
    - ACT-R's Bayesian model and a vector-based model
- Can strengths from one model be incorporated into the other?
  - Word order from vector-based models
  - Prior likelihood and user customization from Bayesian models
- Evaluate accuracy differences between the models
  - How accurate is each model in predicting the correct tags?
  - Is one particular memory model better than the other in prediction?
  - What about computational efficiency and scale?
- Output: Two improved and task-agnostic user-customized tag prediction models

# StackOverflow Example

stackoverflow | Questions | Tags | Users | Badges | Unanswered

## Why this is undefined behavior?

13

My answer to this question was this function:

```
inline bool divisible15(unsigned int x)
{
    //286331153 = (2^32 - 1) / 15
    //4008636143 = (2^32) - 286331153
    return x * 4008636143 <= 286331153;
}
```

It perfectly worked on my machine with VS2008 compiler, however here it doesn't work at all.

Does anyone has an idea, why it I get different results on different compilers? `unsigned` overflow isn't undefined behavior.

**Important note:** after some test it was confirmed it is faster than taking the remainder of the division by 15.

c++ | c | undefined-behavior

share | edit | flag

edited 13 mins ago
H2CO3
93.4k •11 •59 •127

asked 54 mins ago
user2623967
2,033 •5 •29

3 Is this faster than `(x % 15) == 0`? – asveikau 50 mins ago

1 It doesn't show as undefined behavior to me? It probably integer overflows though. – PherricOxide 50 mins ago

@asveikau depends on compiler optimizations – user2623967 50 mins ago

1 Does `x * 4008636143` fit inside an int? – andre 49 mins ago

3 @millimoose Well... these are *unsigned* ints. The overflow behavior is specified. – Mysticial 44 mins ago

add / show **9** more comments

# StackOverflow Dataset

- Use newest StackOverflow dataset released to public
  - Entire dataset is available, so use entire dataset. No need to extract subsets when evaluating models on StackOverflow
  - ~5M posts, ~1B co-occurrence observations

# Twitter Example

**Oprah Winfrey** @Oprah 29 Aug

RT @BishopJakes: Getting ready for #Lifeclass with @oprah. Tweet me if you're in the house! pic.twitter.com/XQMu1W9kw5

View photo

**Barack Obama** @BarackObama 6 Sep

Protection against employment discrimination is a no-brainer. Show your support for #ENDA today: OFA.BO/6rkVHF

Collapse Reply Retweet Favorite More

| 688 | 384 |
|---|---|
| RETWEETS | FAVORITES |

3:06 PM - 6 Sep 13 · Details

**Teen Vogue** @TeenVogue 20h

It'll take a lot to top last year's Superbowl halftime show extravaganza, but if anyone can do it, @BrunoMars can: teenv.ge/17Si69C

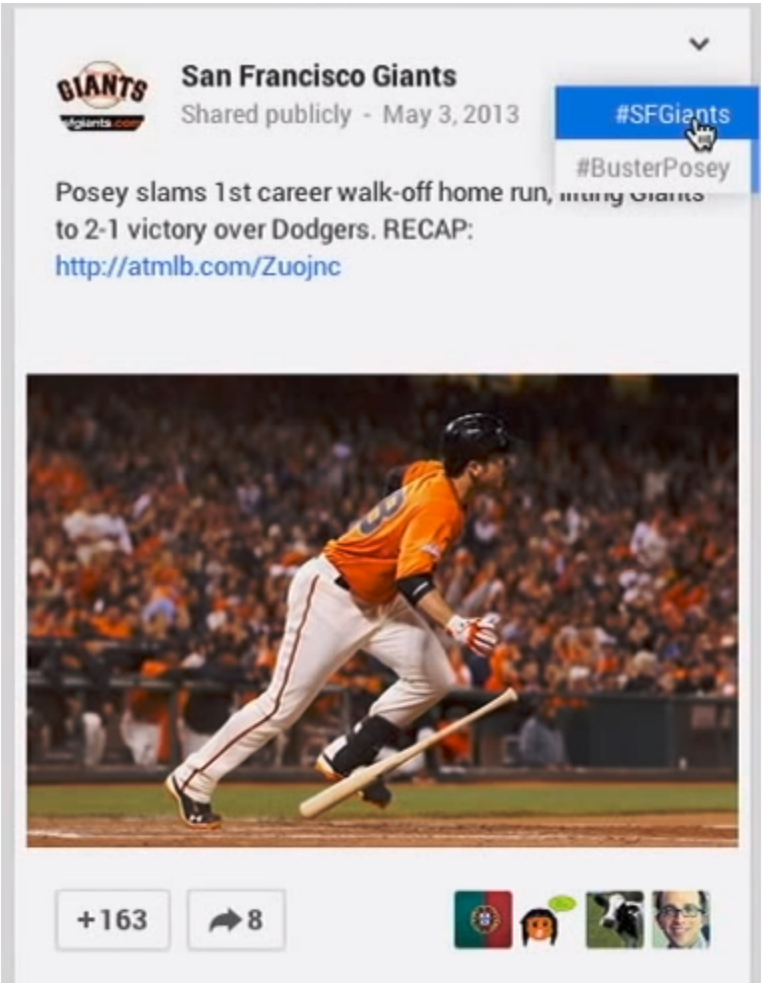
Expand

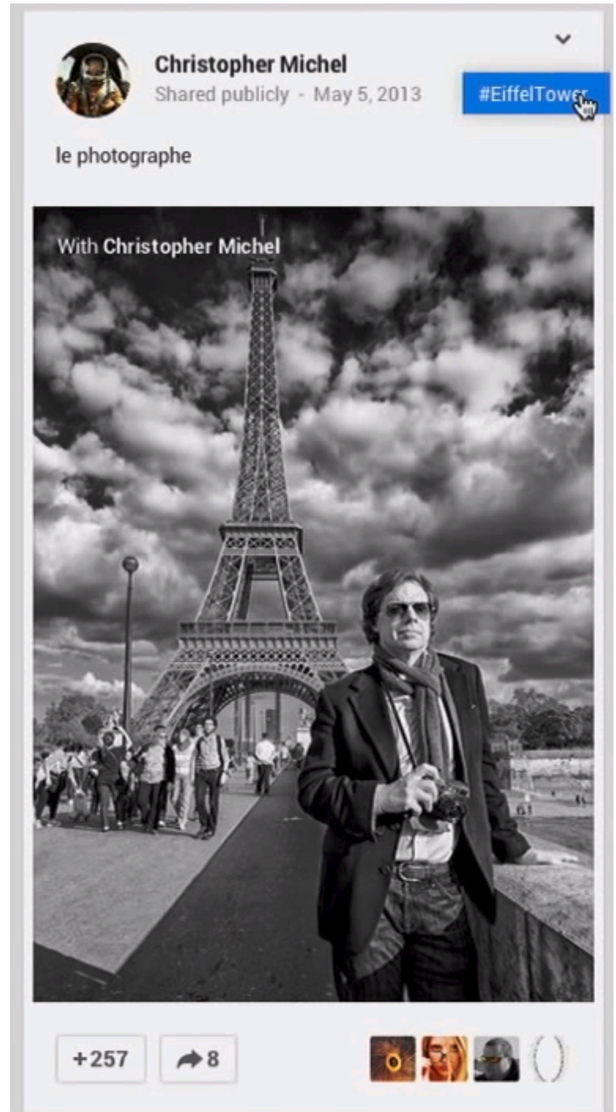
# Twitter Dataset

- Entire dataset is not publicly available. Must extract subsets
  - Popular-users subset
    - As a starting point, current top ~2,000 users
    - Used to evaluate and develop model of prior likelihood of a tag that is customized to the user's past history
  - Popular-hashtags subset
    - As a starting point, current top ~500 hashtags
    - Used to evaluate and improve the contextual component of each model

- ~4M posts per dataset

# Example Application: Google+



# Models: ACT-R Bayesian DM Model

- Naturally incorporates user prior tag use, frequency and recency effects
  - But not word order (yet)

| Common Name | Equation |
| --- | --- |
| Activation | $A_i = B_i + \sum_{j \in c} W_j S_{ji}$ |
| Attentional Weight | $W_j = \frac{W}{n}$ |
| Base Level | $B_i = log \sum_{j=1}^{n} t_j{}^{-d}$ |
| Constant Base Level | $B_i = log \frac{p(i)}{p(\bar{i})}$ |
| Strength of Association | $S_{ji} = log \frac{p(i\|j)}{p(i\|\bar{j})} \approx log \frac{p(i\|j)}{p(i)}$ |
| Recall Probability | $P_i = \left(1 + e^{\frac{\tau - A_i}{s}}\right)^{-1}$ |

# Random-Permutation Vector Model

- Compressed representation of full co-occurrence matrix used for Bayesian models
  - Simplified activation calculation
  - Should scale much better
  - Can also naturally incorporate word-order information

| Common Name | Equation |
|---|---|
| Activation | $A_i = r(m_C, m_i)$ |
| Memory Vector | $m_i = \sum_{i \in allpast} c_i + \sum_{i \in allpast} \sum_{l \in locations} o_{i,l}$ |
| Unordered Context | $c_i = e_i$ |
| Ordered Context | $o_{i,l} = e_{i-l}$ |
| Context Memory Vector | $m_C = \sum_{i \in C} c_i + \sum_{i \in C} \sum_{l \in locations} o_{i,l}$ |
| Environment Vector | $e_i = rand$ |

# Methods: Experiment 1

- Analyze how a user's tag history influences chosen hashtags

- Use StackOverflow and Twitter popular-users dataset
  - Do not use popular-hashtags dataset since this set does not contain enough information about each user's history

- Test results against ACT-R decay rate equations
  - For each user, go from oldest to newest post and predict tags in the post based purely on base-level activation
    - Accumulate tag occurrences and adjust base levels
    - Collect each tag's activation for each post
    - Note if the tag was used (signal) or wasn't used (noise) for that post

  - Monte Carlo search to find best fitting base level activation term
    - Use signal detection measures of performance as DV(s)

# Methods: Experiment 2

- How can a user's past tagging history be incorporated into the random permutation vector-based model?
- Use StackOverflow and Twitter popular-users datasets
- Explore modifications to vector-based model to incorporate a user's tag history
- Evaluate modification based on improvement in model accuracy after adding prior term
  - Use logistic regression statistical technique to optimize term weights
  - Evaluate DVs such as pseudo Rsq, positive predictive value, model accuracy vs. number of tags requested

# Methods: Prior for Vector Models

- One approach to incorporate the user's past tagging history
  - Vector-based models produce a distribution of correlations
    - Chooses the tag with the highest correlation
  - Use the distribution of correlations to compute a probability for each correlation | distribution
  - Use that probability in the Hick-Hyman law to compute a log odds that a tag with that correlation is the correct tag
  - Then add that log odds value to the log odds that the tag is correct | past user history (i.e., ACT-R's base-level activation term)

# Methods: Experiment 3

- How can word order information be incorporated into the ACT-R Bayesian model?

- Use Twitter popular-hashtag dataset
  - Cannot use StackOverflow dataset since tags always occur at the very end of the post

- Explore ways that incorporating word order into the Bayesian model improves tagging accuracy

- Evaluate based on improvement in model accuracy after incorporating word order into the context term
  - Same DVs as for prior experiments (e.g., model accuracy vs. number of tags requested)

# Methods: Word Order for Bayesian

- One approach to incorporate word order into Bayesian co-occurrence models
  - Add an additional dimension in the co-occurrence matrix for word position
  - Compute probability of a tag | context *and* word order
  - Most likely will require smoothing and computational shortcuts to make this practical
    - Possibly bin and store three separate co-occurrence matrices, for words that appear “close”, “mid”, and “far” from hashtags
    - Possibly use a model with a few parameters that describes the distribution of a word’s positions relative to a hashtag

# Methods: Experiment 4

- Predict tags based purely on context (i.e., no prior user history), using all datasets
- Results will show performance differences for the contextual component of each model

# Methods: Experiment 5

- Test the full models (prior and context) on the StackOverflow and Twitter popular-users dataset.
- Will directly compare performance differences between the two full models, after the strengths of each model have been incorporated into the other

# Conclusions: End Results

- Better understanding of how a user's past tagging history influences future tag use
- More thorough evaluation of the strengths/weaknesses of two cognitively-plausible memory models
- Improvements to each memory model where strengths of each are incorporated into the other
- Two efficient implementations of the models

# Conclusions: Broader Impacts

- Produces implementations of two task-agnostic memory models that have numerous potential real-world applications
- Explores how well each memory model scales to datasets with billions of co-occurrence observations
- Explores and tests architectural modifications to the memory models
- If model accuracy improves for these tasks, these modifications may improve accuracy more generally in other memory tasks
- Increases our understanding of how imposing specific architectural constraints on memory models influences what information is retrieved

# Questions

- Thanks for your time
  - Particularly to my advisor, Dr. Byrne, for all of his excellent guidance and feedback for this research