Science Concierge

a fast content-based recommendation system for scientific publications

Titipat Achakuvisut, Daniel Acuna,
Tulakan Ruangrong, Konrad Kording

Department of Biomedical Engineering, Northwestern University,
and Rehabilitation Institute of Chicago

December 21, 2015

K-Lab

Microsoft Azure

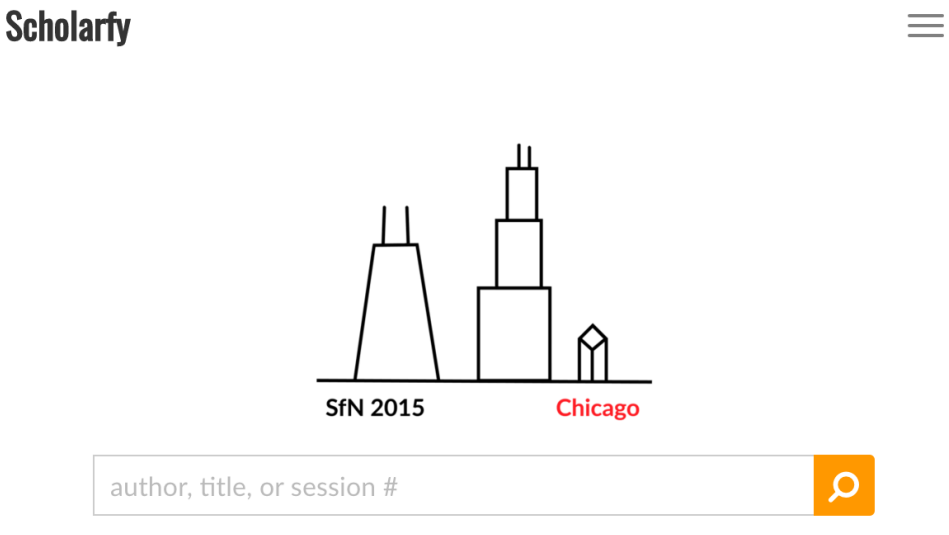
Motivations

- Growing number of publications and conferences
- To read through all documents is impossible
- To search for poster is very difficult. See SfN search website here.
- It's a large-scale problem that requires real-time answers
- Ability to search new documents, relevant documents and group of people who work on the same problem

Proposal

- **Search and like format** because people are good at liking document
- **Automated** way to analyze content/ make suggestion
- **Open-source** we need the community to try and test different approaches

Scholarfy: application



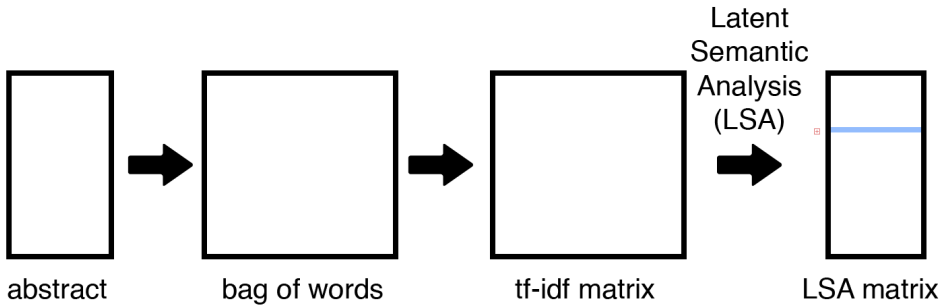
7500 sessions, 2900 users,
48000 page views, 6 minutes average per session

Data and Problems

- Data
  - Scrape data from SfN website (with permission)
  - 15k posters, 70k total authors, 53k unique authors
- The solution has 2 main parts:
  - **Abstract representation** *i.e. how to represent abstract in vector space?*
  - **Abstract recommendation** *i.e. how to suggest posters to conference-goers*?

Abstract representation

Mapping plain abstracts to vector:



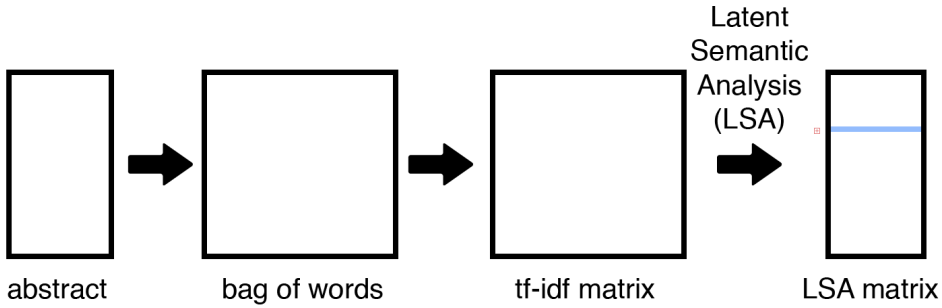
Schematic of the workflow for converting abstracts into vector representations

Applying term frequency–inverse document frequency:

$$\begin{aligned}\text{tf-idf} &= \text{tf} \times \text{idf} \\ &= (1 + \log f) \times \log\left(\frac{N}{d}\right)\end{aligned}$$

- f is total number of words occurrence in each document
- d is number of documents where the term appears

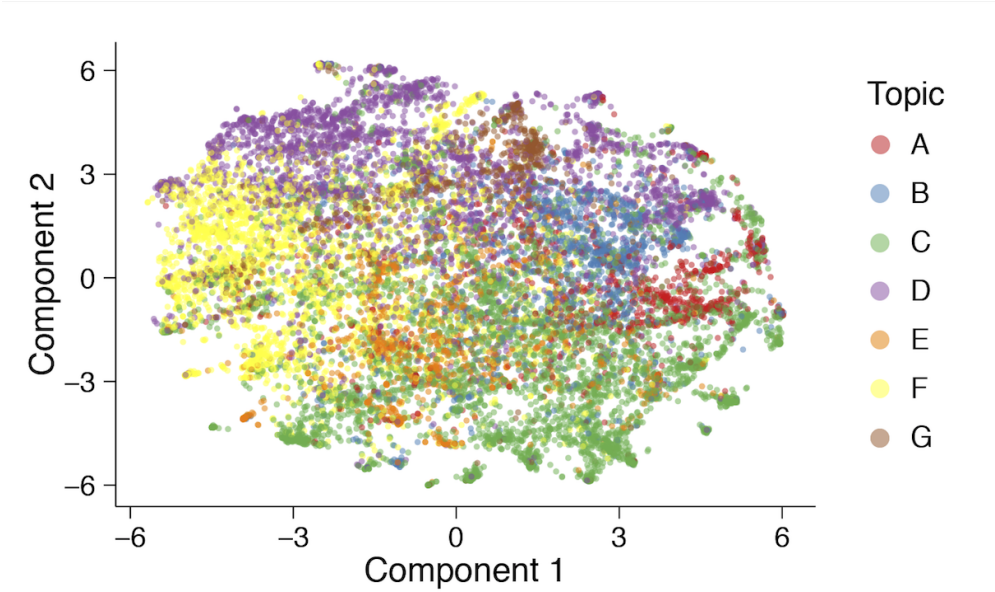
Abstract representation



Schematic of the workflow for converting abstracts into vector representations

- tf-idf vectorizer (weight more if words are distinctive of a document)
- Latent Semantic Analysis - truncated SVD
- $X = U\Sigma V^\intercal \rightarrow X_{LSA} = U_r \Sigma_r V_r^\intercal$

t-SNE in 2D

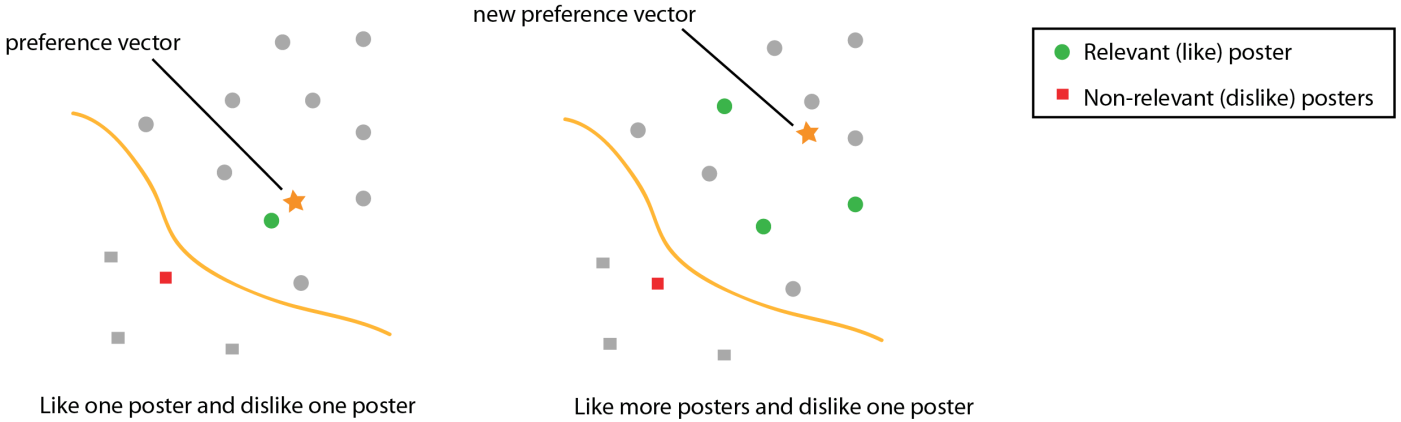


Visualization of topics colored by human curated sessions using t-SNE on abstract vectors.

2D abstract topic representation is grouped by human curated topic, see more **here**. (e.g. A.01.e. Cell migration, F.01.q. Cognitive development, F.01.r. Cognitive aging)

Abstract recommendation

We allow users to like/dislike abstracts and use nearest neighbor to provide N closest abstracts ($\mathbf{x}_a$ = topic of abstract)



Schematic of modified Rocchio Algorithm (T Joachims, 1997)

Abstract recommendation

Original Rocchio Algorithm

$$\mathbf{x}_a = \mathbf{x}_0 + (\alpha \cdot \frac{1}{|X_r|} \sum_{\mathbf{x}_j \in X_r} \mathbf{x}_j) - (\beta \cdot \frac{1}{|X_{nr}|} \sum_{\mathbf{x}_k \in X_{nr}} \mathbf{x}_k)$$

Modified Rocchio Algorithm

$$\mathbf{x}_a = (\frac{\alpha}{|X_{nr}|} \cdot \sum_{\mathbf{x}_j \in X_r} \mathbf{x}_j) - (\frac{\beta}{|X_{nr}|} \cdot \sum_{\mathbf{x}_k \in X_{nr}} \mathbf{x}_k)$$

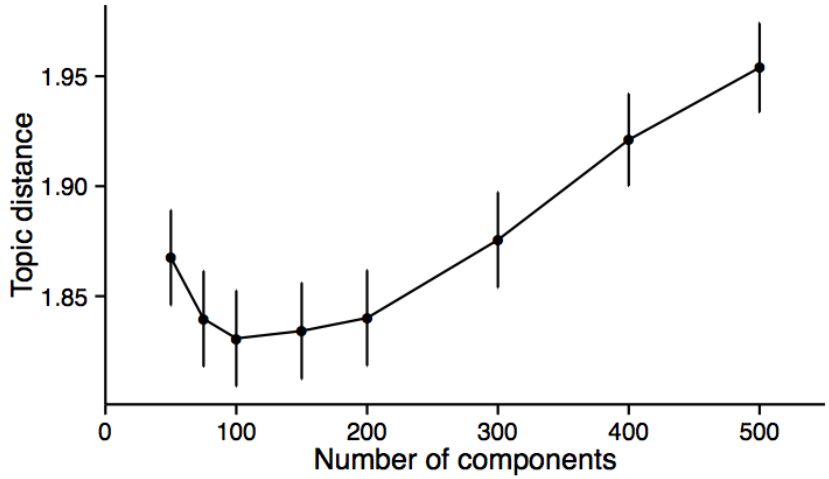
use less than 100 ms for each nearest neighbor assignment (for 50k posters)

Parameter Optimization

Experiment setup

- Each posters has corresponded human curated topic in tree structure i.e. 'F.01.r'
- Assume attendees like one topic
- One trial, randomly like one poster then compute tree distance between liked poster and first 10 suggested posters
- 'F.01.e' and 'F.01.r' has tree distance 1
- This distance is used to validate all parameter choices

Parameter Optimization: number of LSA components

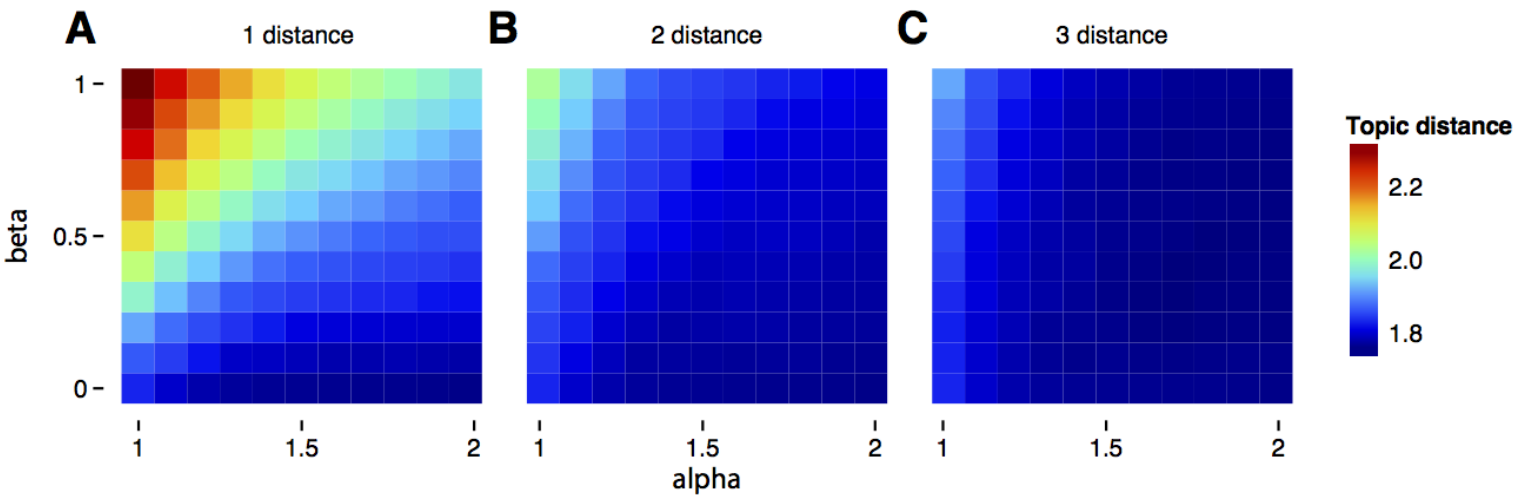


Number of SVD components vs. performance of the algorithm to capture human curated topics

Varying SVD components is not significant. Appropriate number of components is around 100.

Parameter Optimization: weight for Rocchio algorithm

Select one relevant poster and one non-relevant poster (from closest human curated to furthest).

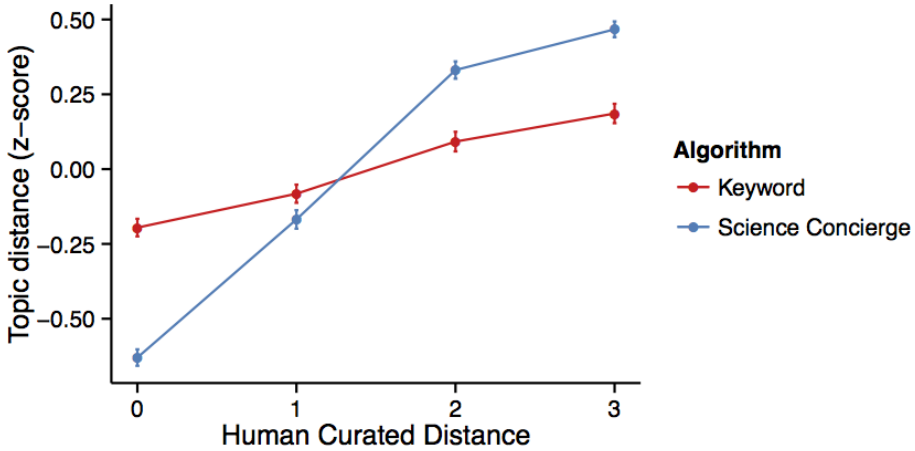


Finding best parameters to weigh relevant and non-relevant votes

Best combination of parameters for non-relevant posters is $\alpha = 1.8$ and $\beta = 0$

Parameter Optimization: algorithm comparison

Select two random posters and compute their distance then correlate that difference with human topic distance

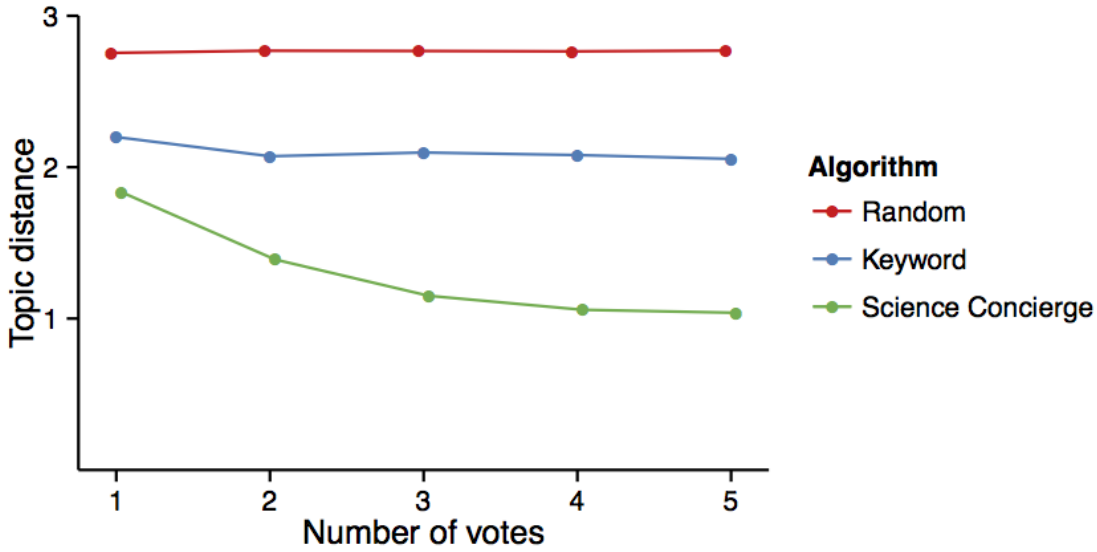


Relationship between human curated distance and topic distance induced by the keyword and Science Concierge models

Spearman's rank correlation of the Science Concierge and keywords are $\rho = 0.442$, $\rho = 0.164$ $(p < 0.001)$.

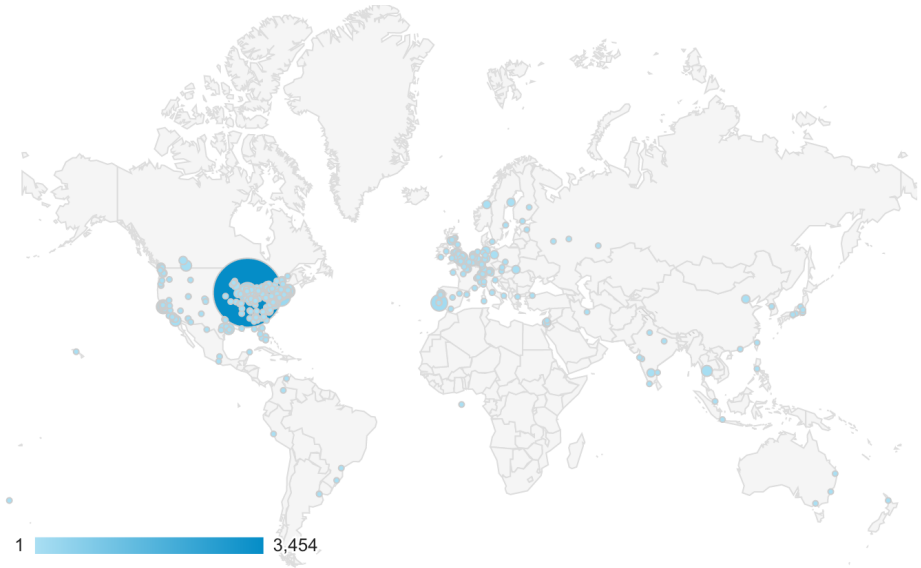
Parameter Optimization: algorithm comparison

Effect of more votes on recommendation quality



Comparison of algorithms as they learn more from a simulated user

Scholarfy: usage

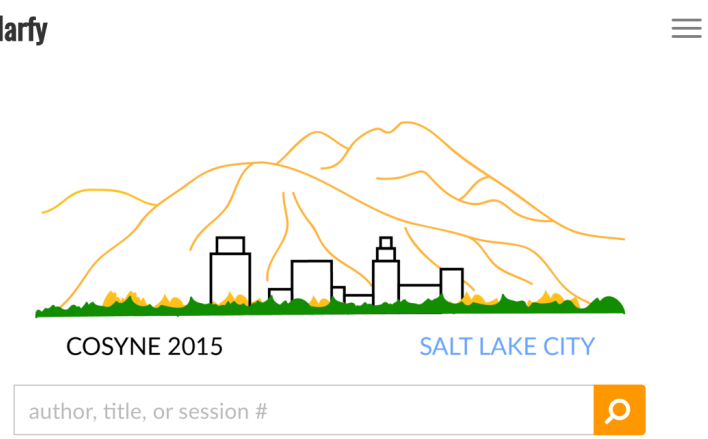


"@KordingLab I joined twitter specifically to thank you for this site. Great job!"

Discussion

- Scalability - LSA is still a batch, normal nearest neighbor will be slower for 10M Pubmed documents
- Suggestion - still assume one liking topic

Applying Scholarfy to review process: COSYNE



- Let reviewers choose preference papers from last year
- Match paper to reviewers based on preferences

Scale Scholarfy to Medline dataset

- Total number of publication = 14.3M
- Apply **Spark** to all pre-processing, allows more than 100 times faster
- Use the same tf-idf vectorizer and sparse SVD
- Apply hierarchical nearest neighbor as nearest neighbor with cosine distance reduce time from 15 second to 1 second
- See more at **pubmed.scholarfy.org**

Acknowledgement

For great mentorship and suggestion

- Daniel Acuna
- Konrad Kording

For great friendship and Github support

- Tulakan Ruangrong

Github projects

- `titipata/science_concierge`
- `tribbloid/spookystuff`

Q/A

K-Lab

RIC
Rehabilitation Institute *of* Chicago

NORTHWESTERN UNIVERSITY
QUAECUMQUE SUNT VERA
1851

Microsoft Azure

amazon
web services™

IP[y]:
IPython

GitHub