Science Concierge: A real-time content-based recommendation system for scientific publications

Titipat Achakulvisut1,\*, Daniel E. Acuna1,2,[[1]](#footnote-1), Tulakan Ruangrong3, Konrad Kording1,2

1 Northwestern University, 2Rehabilitation Institute of Chicago, 3 Mahidon University

**Abstract**

An important part of being a professional scientist is to be able to keep a current account of the scholarly material. This task, however, has become increasingly impossible to perform manually due to the large number of publications produced. We could improve this process by adapting algorithms that are known to work relatively well for music and movie recommendations. However, how these algorithms work with scholarly material has largely been unexplored. Here, we develop a Python library that implements a recommendation system based on the content of voted relevant and irrelevant articles. The library provides real-time and accurate suggestions of new articles based on these votes by using a large scale nearest neighbor search. We tested the library on 15K posters from the Society of Neuroscience Conference 2015. We tuned the algorithm to provide suggestions that were closed in topic classifications to human curated poster sessions. We show that our algorithm significantly outperforms suggestions based on keywords. The work presented here promises to make the exploration of scholarly material faster and more accurate than what it is possible today. We also discuss applications to other domains such as figures.

# Introduction

Since the inception of Internet and large online businesses, recommendation systems are routinely used to suggest new items to users based on past preferences. These systems have been proven useful for music, movies, news, and retail in general [1]. In scientific literature search, to search for new publications, researchers rely mostly on author-provided keywords and citations to find new material. These sources of information hinder their exploration because they are known to be poor and potentially biased [2]. Moreover, this problem is more pronounced during conferences where appropriate keywords may not even exist, let alone citations. An application of recommendation systems based on the researcher’s preferences thus promises to speed up literature search and increase relevance of the findings.

There are multiple recommendation systems that use either the personal preferences of a new user (e.g., content-based recommendations) or exploit the similarity between the new users’ preferences and previous users’ preferences (e.g., collaborative filtering). A large portion of this research is available in commercial software, such as news [3], movie [4], and music [5] applications. There are also more specific systems for scientific literature search. In [6], the authors present a content-based recommendation system that works on PubMed datasets. In [7], the *Scienstein* system combines a large set of criteria for providing literature recommendation. In [8], the authors present a topic-based recommendation system based on a Latent Dirichlet Allocation (LDA) model. It is unclear, however, how these systems scale and the test of their performance is generally indirect.

Here we introduce *Science Concierge* (http://www.github.com/titipata/science\_concierge), an open source Python library that implements a fast and accurate recommendation system for literature search. Briefly, the library uses a scalable vectorization of documents through online Latent Semantic Analysis (LSA) [1]. For the recommendation part, it pairs the Rocchio Algorithm [9] with a large-scale approximate nearest neighbor search based on ball trees [10]. The library aims at providing responsive content-based recommendations using only user’s votes rather than collaborative filtering. The Scholarly software, then, provides an open source solution to content-based scientific recommendation.

We tune and test the algorithm on a collection of scientific posters from the largest Neuroscience conference in the world, Society for Neuroscience (SfN) 2015. First, we cross validate the LSA model to capture most of the variance contained in the topics. Second, we tune the parameters of the algorithm to recommend posters that maximally resemble human curated classifications into poster sessions. We showed that our algorithm significantly outperforms a popular alternative based on keywords. The algorithm implementation is available online at [http://www.scholarfy.net](http://sf.scienceofscience.org) where we use data from Society for Neuroscience (SfN) conference ([http://www.sfn.org](http://sfn.org)).

# Materials and methods

## Conference dataset

Conferences are when systems like the one proposed in this article are most needed. In conferences, the time from submission, acceptance, and presentation are typically much shorter than in journals. This makes it crucial for recommendation system to quickly scan the documents of the conference and let scientific discover material fast.

We obtained a license from the Society for Neuroscience (SfN) on the Neuroscience 2015 conference. This is the largest conference in Neuroscience in the world. This dataset included XXX authors, XXX affiliations, XXX posters, and XXX talks distributed around XXX sessions spanning 5 days. Not all the content of the conference had abstracts available (e.g., talks) and therefore they were dropped from the analysis. The final number of posters analyzed was XXX. These dataset is not publicly available but an academic license can be requested from SfN.

## Methods

The are three main design princples behind *Science Concierge*, which are aimed at adapting recommendation systems to the needs of scientific literature discovery.First, we aim at using the content of the items and avoid using collaborative filter to avoid the Mathew effect in recommendation systems [11], which can detrimental effect for scientific exploration. Second, we aim at proving suggestions as fast as possible. This means that users should get feedback as soon as they vote one item as relevant or irrelevant. Finally, we aim at being accurate, which means that we have to validate the suggestions using some external input. Below we describe the methods to achieve the three goals of *Science Concierge*.

## Content-based recommendation of scientific documents

*Science Concierge* uses only a user’s votes to generate relevant suggestions instead of other users’ votes. As we are mostly applying this algorithm to text documents, the pre processing steps will be explain in text analysis terms. First, the algorithm transform abstract into a vector space using term frequency-inverse document frequency transform (Fig. 1A), dimentionality reduction (Fig. 1B), and then the Rocchio algorithm for the suggestions [9] algorithm (Fig. 1C). By using this method, Science Concierge produces recommendations based on users’ votes.

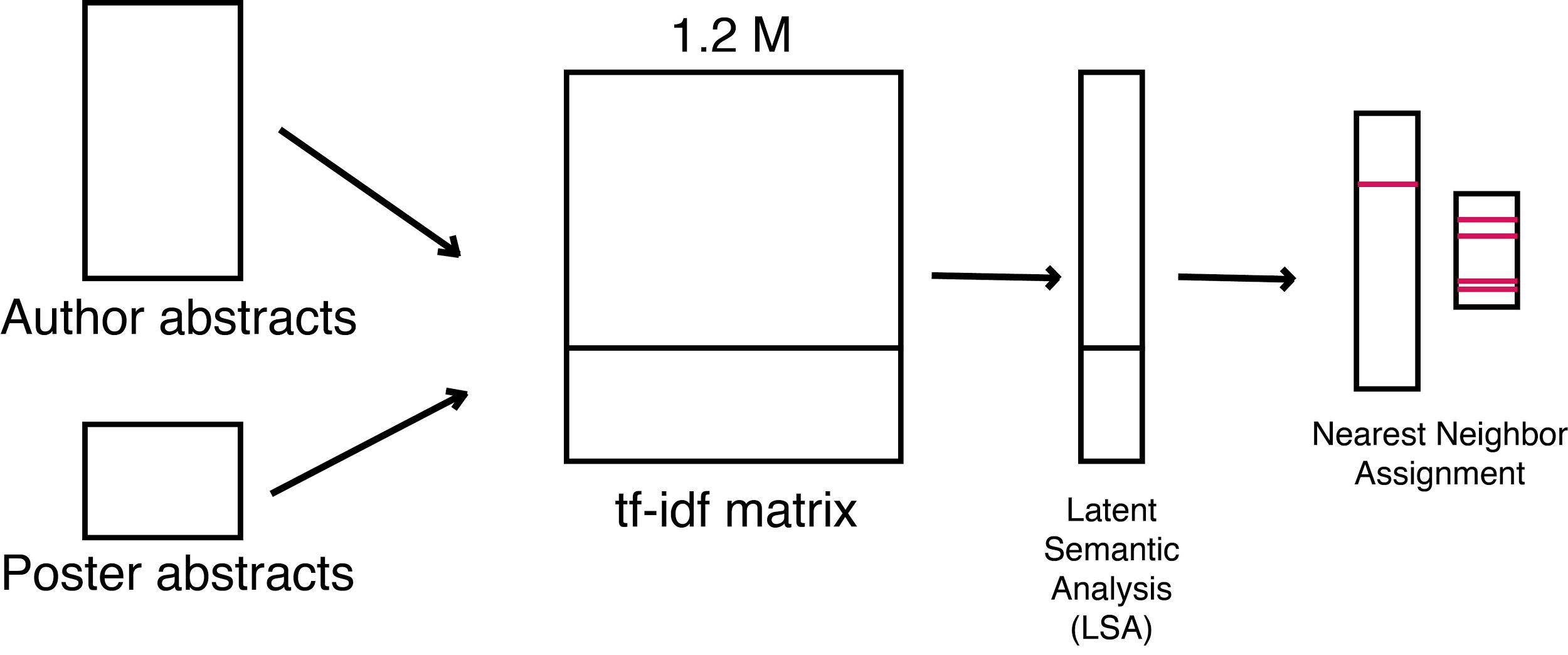


Fig. 1 The schematic of Science Concierge is provided in fig 1. (schematic of whole process)

### Text preprocessing

Documents are toquenized removing English stop words and stemming using Snowball [12]. The terms in the documents are the uni and bi-grams. This term document matrix is transformed by a tf-idf function where terms that appear fewer than XXX were removed, and terms that appear in XX% of documents were removed.

## Latent Semantic Analysis

Latent semantic analysis (LSA) is a topic modeling technique based on singular value decomposition [13]. This technique transform the document term matrix or, in our case, the tf-idf matrix Xas product of left singular vector (U), diagonal singular value matrix (S) and right singular vector (V) as X = USV, where diagonal singular value are ranked from highest to lowest value. When we select only a number of high singular value left and right vectors, then we can approximate the full original matrix X as a lower dimensional matrix X’ that essentially reduces the completely of the problem and potentially noise.

## Poster representation based on keywords

The dataset used here also contained keywords. Search by keyword is a popular way used by scientists to discover topics at conferences and use it as a based comparison. In this algorithm, the LSA analysis is applied to the keyword vector of each poster and all the rest of the analysis is the same.

## Rocchio Algorithm and Nearest Neighbor Assignment

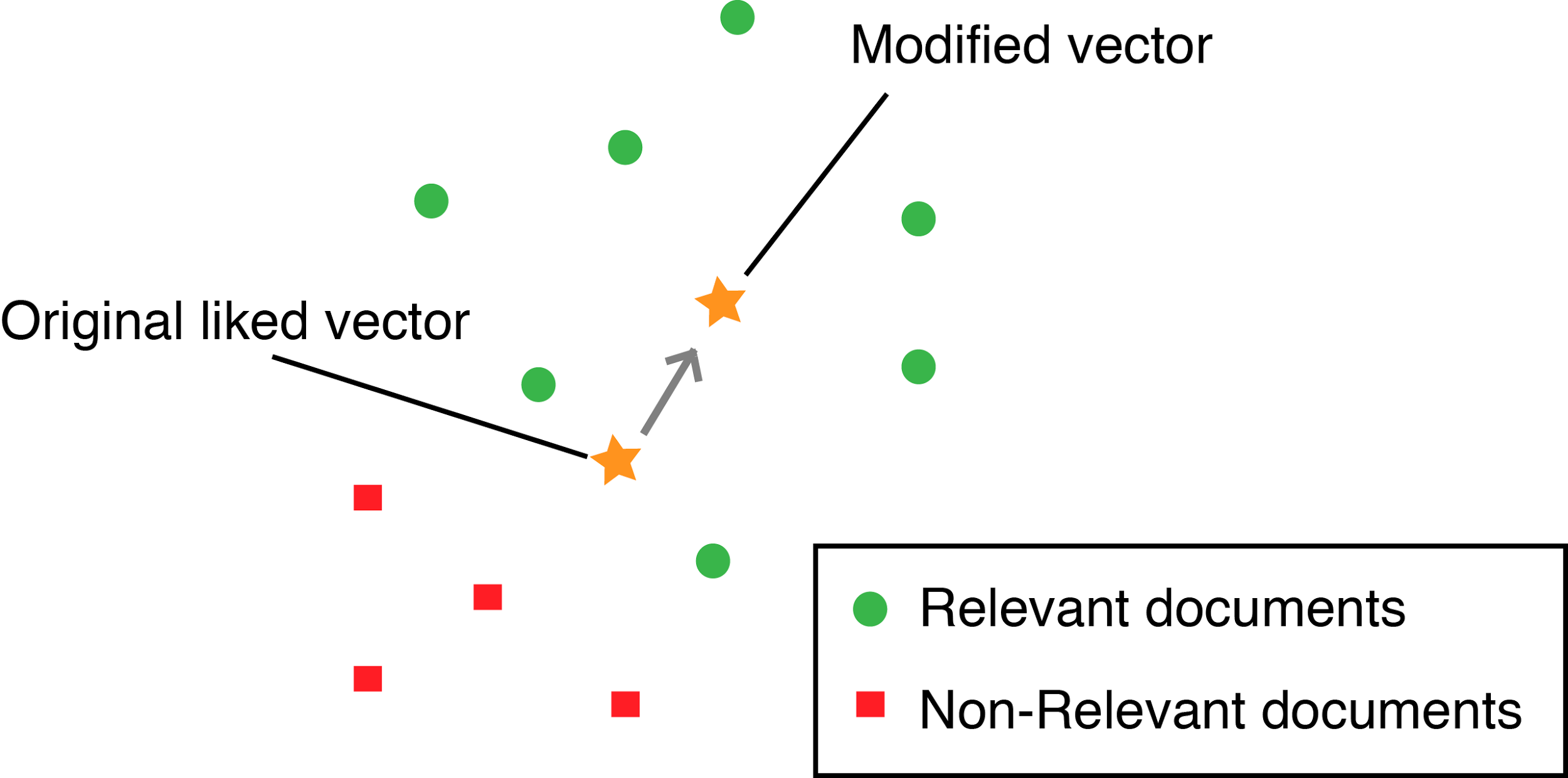
The Rocchio algorithm is used to produced recommendations based on relevant and non-relevant document previously voted by the user. It compute a vector based on the relevant and non-relevant documents. The nearest neighbors of such resulting vector would be documents that are similar to relevant while avoiding the similarity with non-relevant documents.

Mathematically, given a set of documents vectors that are found to be relevant by the user and anoter set of non-relevant documents , the Rocchio algorithm finds a document vector that combines both types of documents

|  |  |  |
| --- | --- | --- |
|  |  | (1) |

where was originaly proposed as the mean document vector of an initial query made into the system, such as a search for a particular keyword. The parameters and control how much the relevant and non-relevant documents affect the final query vector, respectively.

In the case of our system, there is no such element as an initial query and therefore this term is replaced by the mean article in the system. The parameters and are typically found by cross validation using some external validation process.



**Fig 2.** Schematic of the Rocchio algorithm

## A measure of topic distance to human curated classification

Each poster contained in this dataset has been manually classified into topic by the organizers of the conference. The topic classification used in this conference is base on a tree with three levels. To validate many of the parameters of this system, we use this topic classification as an indirect measure of how correct are the suggestions.

We assume that attendees would be highly interested in one topic only and not interested in others. While this assumption may not be accurate for a large number of attendees, we believe in captures the intention behind the classification of topics in this particular SfN conference. To measure the quality of suggestions, then, we assumed that a attendees would like a set of posters from a topic and ask the system to suggest a set of ten posters to attend. For each of the suggestions, we computed the distance of the poster topic to the of the suggestion using the lowest ansenstor measure (given that the topic space is represented hierarchically). The average of the distances between the like topic and the ten suggested posters. This measure was used to validated many of the parameters in the system.

# Results

## Parameter optimization

### Number of components for LSA

Even though the suggestion part can be done in real time, singular value decomposition is still a batch computation where we need to compute topic vectors from plain abstract which require computation power. The main

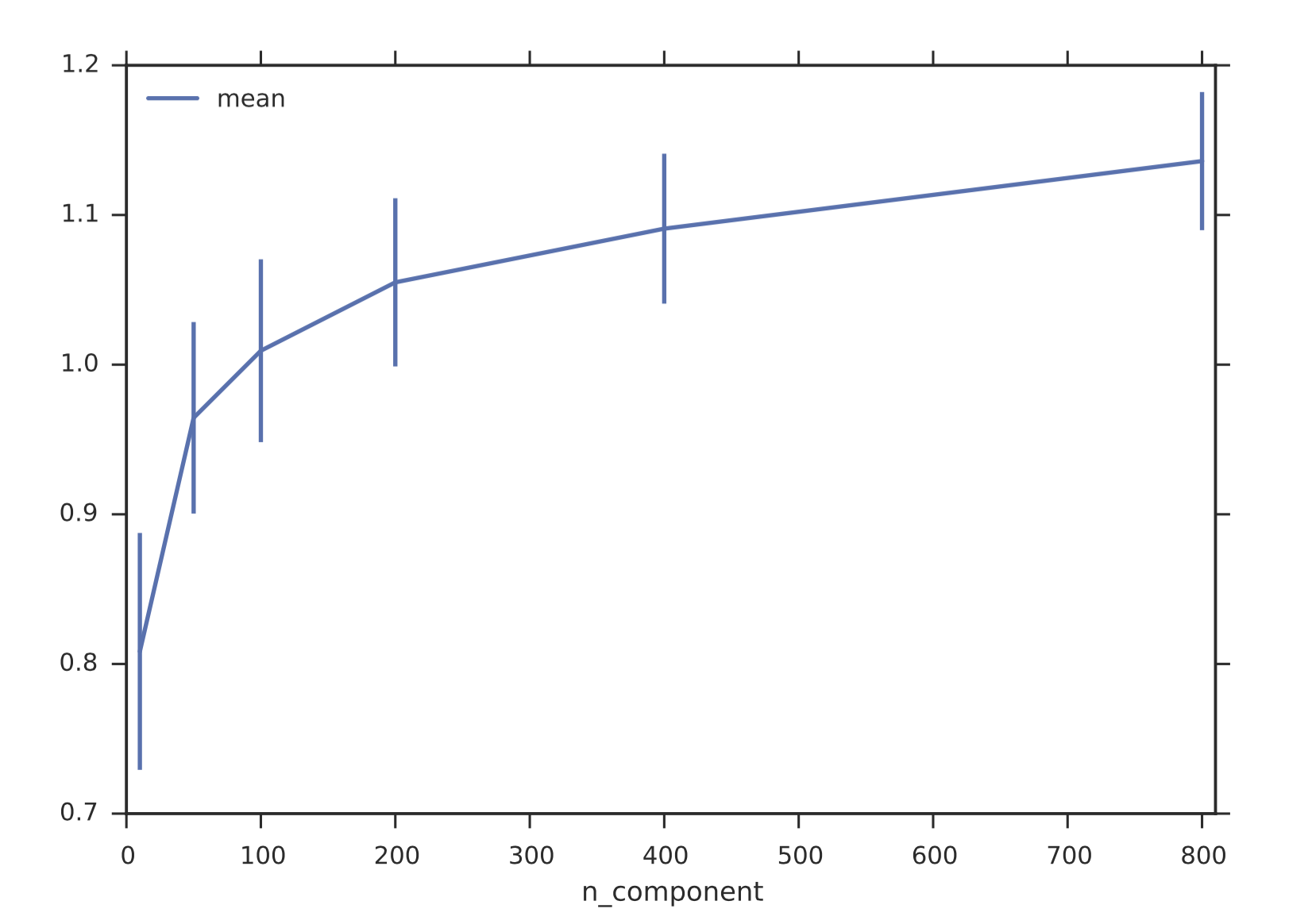


Fig 3. a

### Weight of relevant and non-relevant documents for Rocchio algorithm

We tuned the parameters and in a similar way by performing a grid search over the parameter space by using the average topic distance (see Methods). The experiment this time is done by liking one poster from a random topic and voting non-relevant a poster from a different topic that was at least 2 levels away in the hierarchy. We found that the optimal parameters were and .

It is interesting to note that the parameter for the non-relevant document is set to null. This implies that the way in which the Rocchio algorithm pushes away documents that are not relevant does not affect the final performance of the algorithm. In general, thus, finding documents as non-relevant is more like a feature to remove certain posters from the list of suggested posters.

## Algorithm comparison

Using the full abstract has advantages because we are able to capture variability that is not present in keywords.

As a baseline, we compute a null model that would suggest random papers. This is necessary because the distrubtion of human curated topic distances is not uniformly random. This is how we did it. The average random topic distance is. This will be used as a baseline.

Keywords is common modeling technique. This is how keywords were used to model this dataset. Keywords suck.

The Rocchie algorithm is significantly better. The human curated topic distance is significantly lower than keywords across number of votes (*p* values). Moreover, the reduction in topic distance as number of votes is significantly lower for Rocchio.

Therefore, our algorithm is awesome.

Figure 4. Comparison of algorithms as function of votes.

# Discussion and conclusion

Recapitulation of the aim of the paper. We propose an algorithm. We implement it. We beat another popular suggestion criteria.

We need to model unlikes better. Not liking something that is close makes the suggestions worse (weird). We are examining a particular dataset where this is the case. However, in another dataset the weight of the unlike voters may matter. Always, this is a free parameter that can be cross validated based on human curated topics, which are typically available in conferences.

We are modeling topic spaces as living a linear space. There have been other research directions that model topics in non-linear spaces, such as Non-negative Matrix Factorization [cite] and Latent Dirichlet Allocation [cite]. However, recent research has shown that this modeling approach may not necessarily be enough for capturing topics that are reproducible and accurate [cite me? Maybe? Please?]. This may make non-linear topic modeling not appropriate for the fast speed needed in a system like ours. In the future, we will address this question in more detail.

The problem addressed in this work occurs in a large range of disciplines and types of items. For example, nothing prevents this algorithm to be applied to movies, news, pictures, or music. More specifically, we plan on extending of poster suggestion website to include journal articles from the Pubmed Open Access Subset. For any of these extensions, however, we make the code freely available on Github (http://www.github.com/titipata/science\_concierge). We believe that a tool like and an approach like this solve the problem of exploring scientific work will accelerate the discovery of topics.

**Acknowledgements**

Titipat Achakulvisut was supported by Royal Thai Government Scholarship grant #50AC002. DAniel E. Acuna was supposed by the John Templeton Foundation grant #. Tulakan Ruangrong would like to thank… Konrad Kording was supposed by NIH # and John Templeton Foundation #. Thanks to SfN for providing the abstract dataset for testing.

**Author Contributions**

Titipat Achakulvisut (writing, programming, concept), Daniel Acuna (writing, programming, concept), Tulakan Ruangron (programming, concept), Konrad Kording (writing, concept).

**References**

1. Manning, C.D., P. Raghavan, and H. Schütze, *Introduction to information retrieval*. 2008, New York: Cambridge University Press. xxi, 482 p.

2. Chavalarias, D. and J.P. Ioannidis, *Science mapping analysis characterizes 235 biases in biomedical research.* J Clin Epidemiol, 2010. **63**(11): p. 1205-15.

3. Li, L., et al. *A contextual-bandit approach to personalized news article recommendation*. in *Proceedings of the 19th international conference on World wide web*. 2010. ACM.

4. Bell, R.M. and Y. Koren, *Lessons from the Netflix prize challenge.* ACM SIGKDD Explorations Newsletter, 2007. **9**(2): p. 75-79.

5. Ali, M., C.C. Johnson, and A.K. Tang. *Parallel Collaborative Filtering for Streaming Data*. 2011; Available from: <https://github.com/MrChrisJohnson/CollabStream>.

6. Yoneya, T. and H. Mamitsuka. *PURE: a PubMed article recommendation system based on content-based filtering*. in *Genome Inform*. 2007. World Scientific.

7. Gipp, B., J. Beel, and C. Hentschel. *Scienstein: A research paper recommender system*. in *Proceedings of the international conference on emerging trends in computing (ICETiC’09)*. 2009.

8. Wang, C. and D.M. Blei. *Collaborative topic modeling for recommending scientific articles*. in *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2011. ACM.

9. Rocchio, J.J., *Relevance feedback in information retrieval.* 1971.

10. Shakhnarovich, G., T. Darrell, and P. Indyk, *Nearest-Neighbors methods in Learning and Vision: Theory and Practice*. 2006: MIT Press.

11. Petersen, A.M., et al., *Quantitative and empirical demonstration of the Matthew effect in a study of career longevity.* Proc Natl Acad Sci U S A, 2011. **108**(1): p. 18-23.

12. Porter, M.F., *An Algorithm for Suffix Stripping.* Program-Automated Library and Information Systems, 1980. **14**(3): p. 130-137.

13. Landauer, T.K., P.W. Foltz, and D. Laham, *An introduction to latent semantic analysis.* Discourse Processes, 1998. **25**(2-3): p. 259-284.

1. Corresponding authors: [titipat.a@northwestern.edu](mailto:titipat.a@northwestern.edu), daniel.acuna@northwestern.edu [↑](#footnote-ref-1)