

Information Retrieval and Web Search

Cornelia Caragea

Computer Science
University of Illinois at Chicago

Web Search and Link Analysis

Required Reading

- “Information Retrieval” textbook
 - Chapter 21: Link Analysis

Searching the Web

Google Florin Marin

Web Images Maps Shopping News More Search tools

About 58,800,000 results (0.38 seconds)

[Cornell University](#)

www.cornell.edu/

Cornell University contains seven undergraduate colleges plus the College of Veterinary Medicine, the Law School, the Samuel Curtis Johnson Graduate ...

Score: **25** / 30 · [41 Google reviews](#) · [Write a review](#)

410 Thurston Ave Ithaca, NY 14850
(607) 255-5241

[Admissions](#) · [Academics](#) · [CUinfo](#)

[Cornell University - Wikipedia, the free encyclopedia](#)

en.wikipedia.org/wiki/Cornell_University

Cornell University is an American private Ivy League research university located in Ithaca, New York, United States. Founded in 1865 by Ezra Cornell and ...

[History](#) · [Ithaca, New York](#) · [List of Cornell University alumni](#) · [Arts and Sciences](#)

[Cornell University Athletics](#)

www.cornellbigred.com/

Official web site of Big Red athletics. Information about varsity sports, facilities, schedules, and the department, as well as an alumni section and Big Red Store.

[Cornell University \(Cornell\) on Twitter](#)

<https://twitter.com/Cornell>

The latest from Cornell University (@Cornell). Cornell University Twitter feed. Ithaca, NY.

[Cornell Home](#)

www.cornellcollege.edu/

Residential liberal arts college established in 1853. Operates under the distinctive One-Course-At-A-Time academic calendar.



Cornell University

80,190 followers on Google+

[Directions](#)

[Follow](#)

Cornell University is an American private Ivy League research university located in Ithaca, New York, United States. [Wikipedia](#)

Address: 410 Thurston Ave, Ithaca, NY 14850

Acceptance rate: 16.2% (2012)

Mascot: Big Red Bear


Phone: (607) 255-5241

Colors: White, Carmelian

Founders: [Andrew Dickson White](#), [Ezra Cornell](#)

Recent posts

Searching the Web



Florian Marinsek

[Web](#) [Images](#) [Maps](#) [Shopping](#) [News](#) [More ▾](#) [Search tools](#)

About 58,800,000 results (0.38 seconds)

[Cornell University](#)
www.cornell.edu/ ▾
Cornell University contains seven undergraduate colleges plus the College of Veterinary Medicine, the Law School, the Samuel Curtis Johnson Graduate ...
Score: **25** / 30 · [41 Google reviews](#) · [Write a review](#)

410 Thurston Ave Ithaca, NY 14850
(607) 255-5241



[Admissions](#) · [Academics](#) · [CUinfo](#)

[Cornell University - Wikipedia, the free encyclopedia](#)
en.wikipedia.org/wiki/Cornell_University ▾
Cornell University is an American private Ivy League research university located in Ithaca, New York, United States. Founded in 1865 by Ezra Cornell and ...
[History](#) · [Ithaca, New York](#) · [List of Cornell University alumni](#) · [Arts and Sciences](#)

[Cornell University Athletics](#)
www.cornellbigred.com/ ▾
Official web site of Big Red athletics. Information about varsity sports, facilities, schedules, and the department, as well as an alumni section and Big Red Store.

[Cornell University \(Cornell\) on Twitter](#)
<https://twitter.com/Cornell> ▾
The latest from Cornell University (@Cornell). Cornell University Twitter feed. Ithaca, NY.

[Cornell Home](#)
www.cornellcollege.edu/ ▾
Residential liberal arts college established in 1853. Operates under the distinctive One-Course-At-A-Time academic calendar.



Cornell University

80,190 followers on Google+

[Directions](#) [Follow](#)

Cornell University is an American private Ivy League research university located in Ithaca, New York, United States. [Wikipedia](#)

Address: 410 Thurston Ave, Ithaca, NY 14850
Acceptance rate: 16.2% (2012)
Mascot: Big Red Bear
Phone: (607) 255-5241
Colors: White, Carmelian
Founders: [Andrew Dickson White](#), [Ezra Cornell](#)

Recent posts

How did Google “know” that “Cornell University” is the best answer?

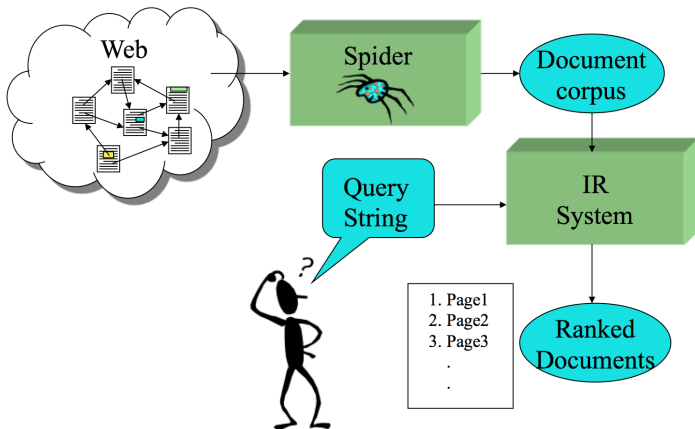
Searching the Web: The Problem of Ranking

- When issuing the single-word query “Cornell,” a search engine does not have very much to go on.
 - Did the searcher want information about the university?
 - The university’s hockey team?
 - Cornell College in Iowa?
 - The Nobel-Prize-winning physicist Eric Cornell?

Searching the Web: The Problem of Ranking

- Search engines determine how to rank pages using automated methods that look at the Web itself, not some external source of knowledge.
- There must be enough information *intrinsic* to the Web and its structure to figure out that “Cornell University” is the best answer.

Web Search System



Key issues for search engines:

- To filter, from among an enormous number of documents, the few that are most important

Web Search System

Understanding the network structure of Web pages is crucial for understanding what documents a search engine should return!

Back to the query “Cornell”:

- No internal features of the page www.cornell.edu are really helpful:
 - “Cornell” does not necessarily occur more frequently within this page content than within others, relevant to the query

Web Search System

Understanding the network structure of Web pages is crucial for understanding what documents a search engine should return!

Back to the query “Cornell”:

- No internal features of the page www.cornell.edu are really helpful:
 - “Cornell” does not necessarily occur more frequently within this page content than within others, relevant to the query
- Rather, features extracted from the [link structure](#) are more helpful:
 - When a page is relevant to the query “Cornell”, very often it links to www.cornell.edu

Link Analysis using Hubs and Authorities

Links are Essential to Ranking!

- We can use links to assess the **authority of a page on a topic**, through implicit endorsements that other pages on the topic confer through their links to it.
- **Experiment with the query “Cornell”:**
 - Collect pages that are relevant to “Cornell” using IR (text-only) techniques.
 - Let these pages “vote” through their links for pages on the Web.
 - Which page on the Web receives the greatest number of in-links from pages that are relevant to Cornell?

Links are Essential to Ranking!

- We can use links to assess the **authority of a page on a topic**, through implicit endorsements that other pages on the topic confer through their links to it.
- **Experiment with the query “Cornell”:**
 - Collect pages that are relevant to “Cornell” using IR (text-only) techniques.
 - Let these pages “vote” through their links for pages on the Web.
 - Which page on the Web receives the greatest number of in-links from pages that are relevant to Cornell?
 - Answer: www.cornell.edu

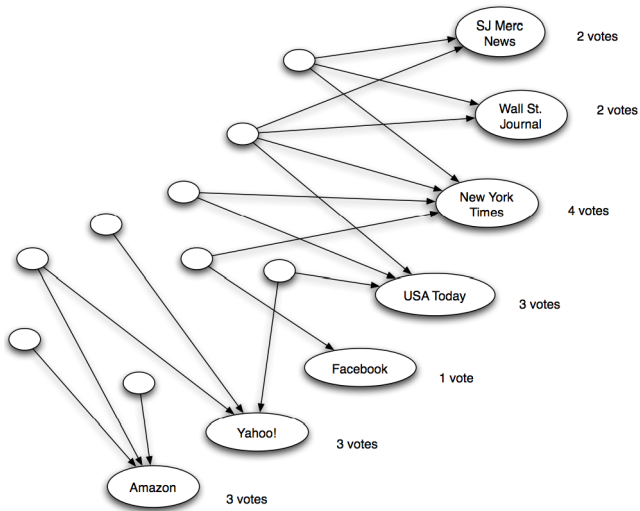
Links are Essential to Ranking!

- Experiment with the query “newspapers”:
 - What is the “best” answer to the query “newspapers”?

Links are Essential to Ranking!

- Experiment with the query “newspapers”:
 - What is the “best” answer to the query “newspapers”?
 - No single right answer
 - Best expected answer: [A list of most important ones](#)
 - Collect pages relevant to “newspapers” and let them vote through their links
 - **Result:** a mix of prominent newspapers along with pages that are going to receive a lot of in-links no matter what the query is - pages like Facebook and Amazon.

In-Links

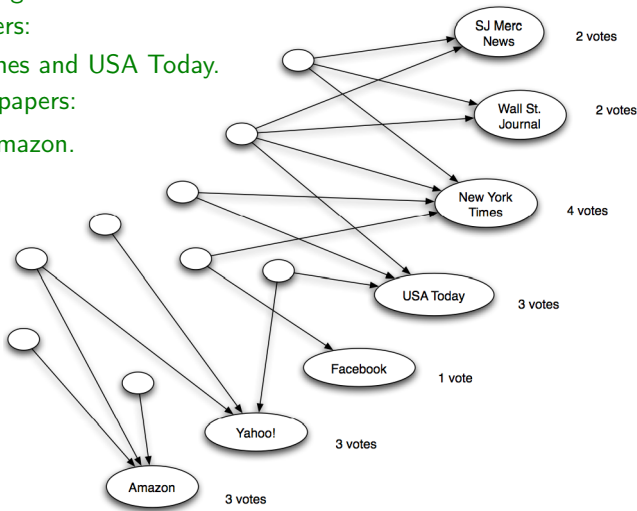


The unlabeled circles represent pages relevant to the query “newspaper.”

In-Links

Four highly-ranked pages:

- two are newspapers:
 - New York Times and USA Today.
- two are not newspapers:
 - Yahoo! and Amazon.

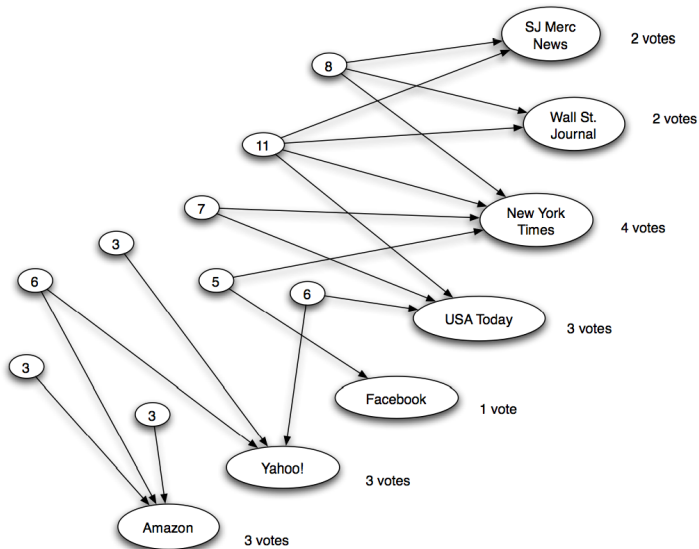


The unlabeled circles represent pages relevant to the query "newspaper."

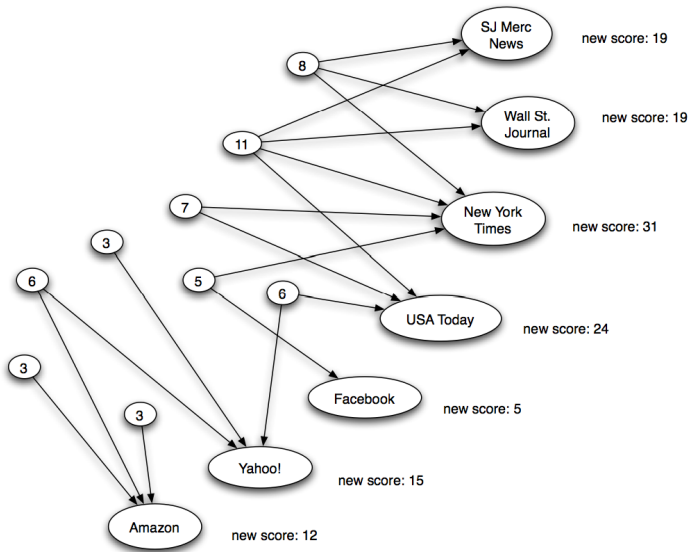
Finding Good Lists

- In addition to the newspapers themselves, there is another kind of useful answer to our query: pages that compile lists of resources relevant to the topic.
- If we could find good list pages for newspapers, we would have another approach to the problem of finding the newspapers themselves.
- Intuitively, these pages have some sense where the good answers are, and we score them highly as lists.
 - A page's value as a list is equal to the sum of the votes received by all pages that it voted for.

Finding Good Lists



Re-Weighting



Authorities and Hubs

- **Authorities for a query** are pages that are recognized as providing significant, trustworthy, and useful information on a topic
 - In-degree (number of pointers to a page) is one simple measure of authority
 - However in-degree treats all links as equal
 - Links from pages that are themselves authoritative should count more
- **Hubs for a query** are index pages that provide lots of useful links to relevant content pages (topic authorities)

Authorities and Hubs - Examples

- Authorities:
 - Newspaper home pages
 - Course home pages
 - Home pages of auto manufacturers
- Hubs
 - List of newspapers
 - Course bulletin
 - List of US auto manufacturers

Ranking by Hyperlink-Induced Topic Search (HITS) algorithm

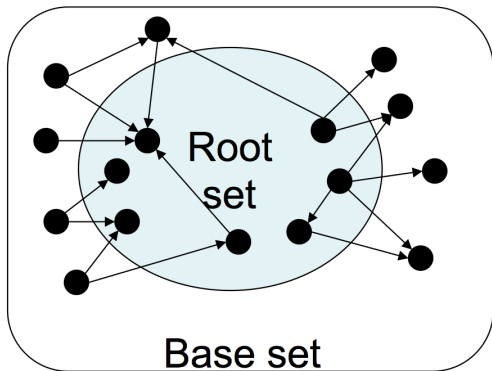
- Attempts to computationally determine hubs and authorities on a particular topic through analysis of a relevant subgraph of the web
- Based on mutually recursive facts:
 - Hubs point to lots of authorities
 - Authorities are pointed to by lots of hubs

The HITS Algorithm

- Computes hubs and authorities for a particular topic specified by a normal query
- First determines a set of relevant pages for the query called the base set S
- Analyze the link structure of the web subgraph defined by S to find authority and hub pages in this set

Constructing a Base Subgraph

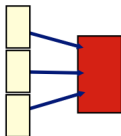
- For a specific query Q , let the set of documents returned by a standard search engine be called the root set R
- Initialize the base set S to R
- Add to S all pages pointed to by any page in R
- Add to S all pages that point to any page in R



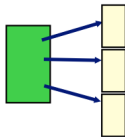
HITS

Goal: Given a query, find:

- Good sources of content (authorities)

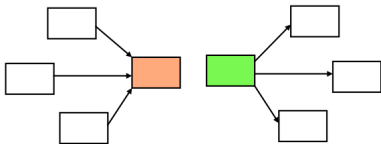


- Good sources of links (hubs)

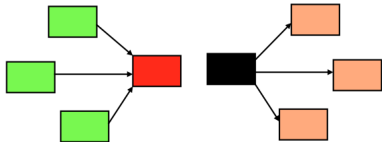


Intuition

- **Authority** comes from in-edges.
Being a **good hub** comes from out-edges.



- **Better authority** comes from in-edges from **good hubs**.
Being a **better hub** comes from out-edges to **good authorities**.



Iterative Algorithm

- Use an iterative algorithm to slowly converge on a mutually reinforcing set of hubs and authorities
- Maintain for each page $p \in S$:
 - Authority score: a_p (vector \mathbf{a})
 - Hub score: h_p (vector \mathbf{h})
- Initialize all $a_p = h_p = 1$
- Maintain normalized scores by normalizing with the sum of authorities (or hubs) in the graph.

HITS Update Rules

- Authorities are pointed to by lots of good hubs:

$$a_p = \sum_{q:q \rightarrow p} h_q$$

- Hubs point to lots of good authorities:

$$h_p = \sum_{q:p \rightarrow q} a_q$$

- Repeat until vectors **a** and **h** converge

The HITS Iterative Algorithm

- Initialize $a_p = h_p = 1$ for all $p \in S$
- For $i = 1$ to k :
 - For all $p \in S$: update authority scores (based on old hubs)

$$a_p = \sum_{q:q \rightarrow p} h_q$$

- For all $p \in S$: update hub scores (based on old authorities)

$$h_p = \sum_{q:p \rightarrow q} a_q$$

- For all $p \in S$: normalize **a**

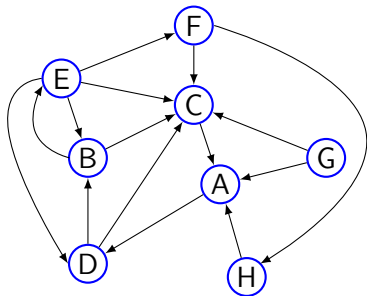
$$a_p = \frac{a_p}{\sum_{p \in S} a_p}$$

- For all $p \in S$: normalize **h**

$$h_p = \frac{h_p}{\sum_{p \in S} h_p}$$

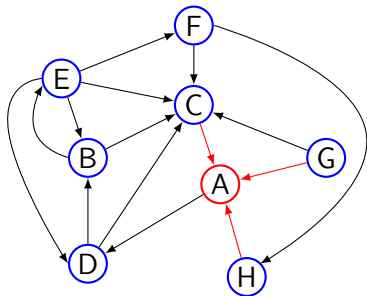
The HITS Iterative Algorithm: Example

	Old Auth	Old Hub	New Auth	New Hub
A	1	1		
B	1	1		
C	1	1		
D	1	1		
E	1	1		
F	1	1		
G	1	1		
H	1	1		



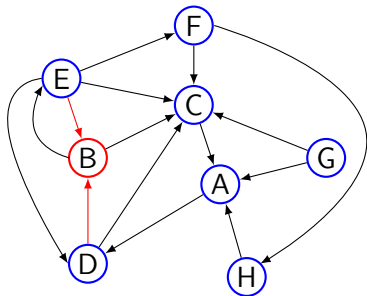
The HITS Iterative Algorithm: Example

	Old Auth	Old Hub	New Auth	New Hub
A	1	1	3	
B	1	1		
C	1	1		
D	1	1		
E	1	1		
F	1	1		
G	1	1		
H	1	1		



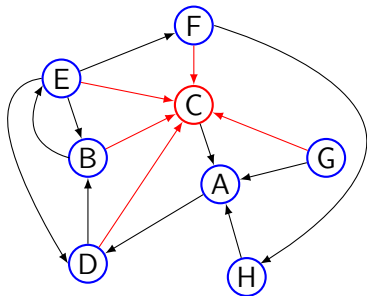
The HITS Iterative Algorithm: Example

	Old Auth	Old Hub	New Auth	New Hub
A	1	1	3	
B	1	1	2	
C	1	1		
D	1	1		
E	1	1		
F	1	1		
G	1	1		
H	1	1		



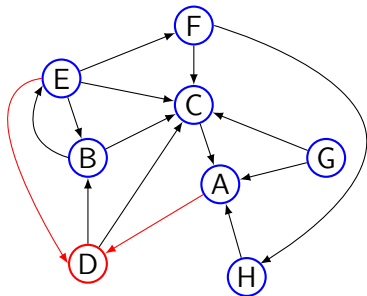
The HITS Iterative Algorithm: Example

	Old Auth	Old Hub	New Auth	New Hub
A	1	1	3	
B	1	1	2	
C	1	1	5	
D	1	1		
E	1	1		
F	1	1		
G	1	1		
H	1	1		



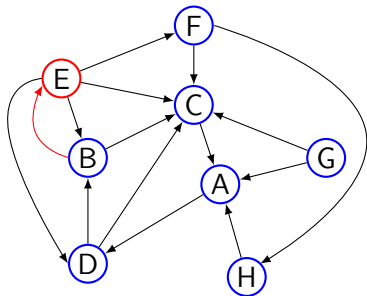
The HITS Iterative Algorithm: Example

	Old Auth	Old Hub	New Auth	New Hub
A	1	1	3	
B	1	1	2	
C	1	1	5	
D	1	1	2	
E	1	1		
F	1	1		
G	1	1		
H	1	1		



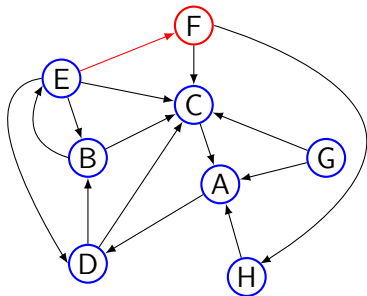
The HITS Iterative Algorithm: Example

	Old Auth	Old Hub	New Auth	New Hub
A	1	1	3	
B	1	1	2	
C	1	1	5	
D	1	1	2	
E	1	1	1	
F	1	1		
G	1	1		
H	1	1		



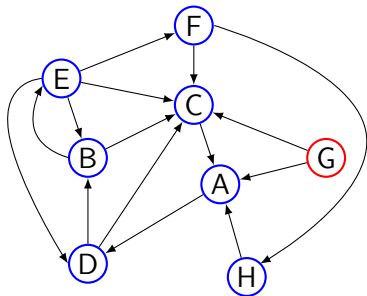
The HITS Iterative Algorithm: Example

	Old Auth	Old Hub	New Auth	New Hub
A	1	1	3	
B	1	1	2	
C	1	1	5	
D	1	1	2	
E	1	1	1	
F	1	1	1	
G	1	1		
H	1	1		



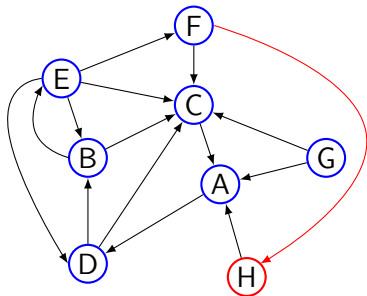
The HITS Iterative Algorithm: Example

	Old Auth	Old Hub	New Auth	New Hub
A	1	1	3	
B	1	1	2	
C	1	1	5	
D	1	1	2	
E	1	1	1	
F	1	1	1	
G	1	1	0	
H	1	1		



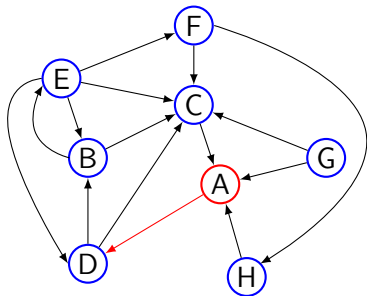
The HITS Iterative Algorithm: Example

	Old Auth	Old Hub	New Auth	New Hub
A	1	1	3	
B	1	1	2	
C	1	1	5	
D	1	1	2	
E	1	1	1	
F	1	1	1	
G	1	1	0	
H	1	1	1	



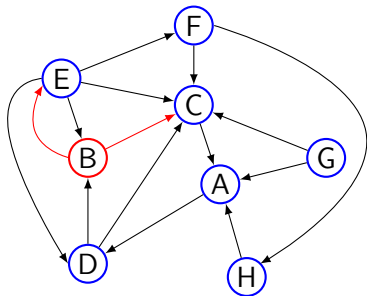
The HITS Iterative Algorithm: Example

	Old Auth	Old Hub	New Auth	New Hub
A	1	1	3	1
B	1	1	2	
C	1	1	5	
D	1	1	2	
E	1	1	1	
F	1	1	1	
G	1	1	0	
H	1	1	1	



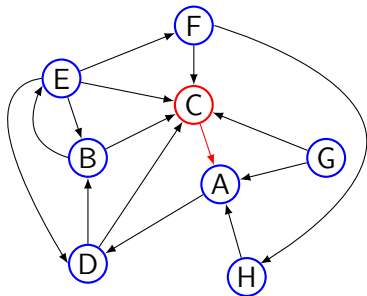
The HITS Iterative Algorithm: Example

	Old Auth	Old Hub	New Auth	New Hub
A	1	1	3	1
B	1	1	2	2
C	1	1	5	
D	1	1	2	
E	1	1	1	
F	1	1	1	
G	1	1	0	
H	1	1	1	



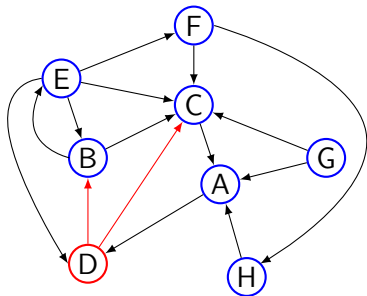
The HITS Iterative Algorithm: Example

	Old Auth	Old Hub	New Auth	New Hub
A	1	1	3	1
B	1	1	2	2
C	1	1	5	1
D	1	1	2	
E	1	1	1	
F	1	1	1	
G	1	1	0	
H	1	1	1	



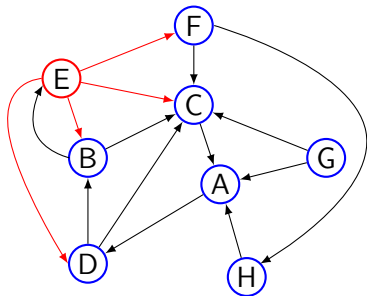
The HITS Iterative Algorithm: Example

	Old Auth	Old Hub	New Auth	New Hub
A	1	1	3	1
B	1	1	2	2
C	1	1	5	1
D	1	1	2	2
E	1	1	1	
F	1	1	1	
G	1	1	0	
H	1	1	1	



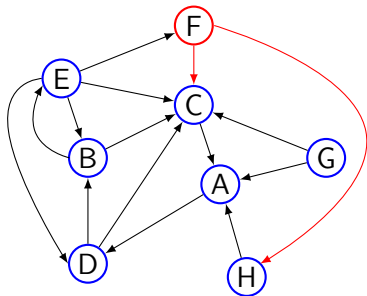
The HITS Iterative Algorithm: Example

	Old Auth	Old Hub	New Auth	New Hub
A	1	1	3	1
B	1	1	2	2
C	1	1	5	1
D	1	1	2	2
E	1	1	1	4
F	1	1	1	
G	1	1	0	
H	1	1	1	



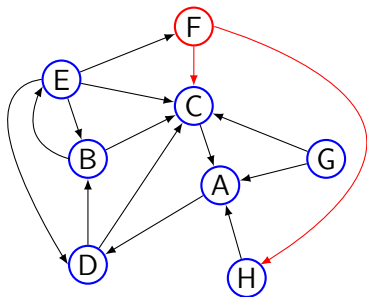
The HITS Iterative Algorithm: Example

	Old Auth	Old Hub	New Auth	New Hub
A	1	1	3	1
B	1	1	2	2
C	1	1	5	1
D	1	1	2	2
E	1	1	1	4
F	1	1	1	2
G	1	1	0	
H	1	1	1	



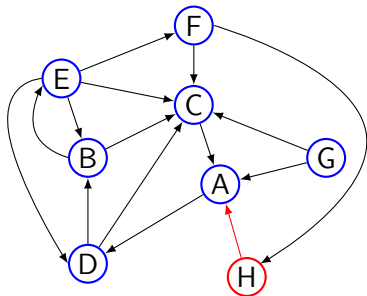
The HITS Iterative Algorithm: Example

	Old Auth	Old Hub	New Auth	New Hub
A	1	1	3	1
B	1	1	2	2
C	1	1	5	1
D	1	1	2	2
E	1	1	1	4
F	1	1	1	2
G	1	1	0	
H	1	1	1	



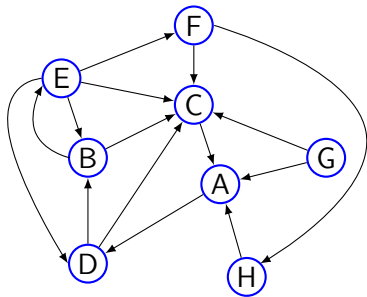
The HITS Iterative Algorithm: Example

	Old Auth	Old Hub	New Auth	New Hub
A	1	1	3	1
B	1	1	2	2
C	1	1	5	1
D	1	1	2	2
E	1	1	1	4
F	1	1	1	2
G	1	1	0	2
H	1	1	1	1



The HITS Iterative Algorithm: Example

	Old Auth	Old Hub	New Auth	New Hub
A	1	1	3	1
B	1	1	2	2
C	1	1	5	1
D	1	1	2	2
E	1	1	1	4
F	1	1	1	2
G	1	1	0	2
H	1	1	1	1

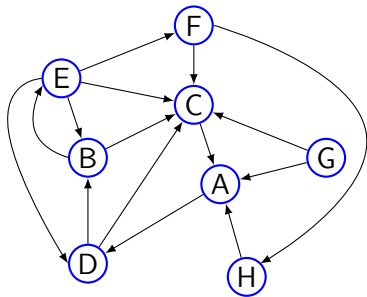


Normalize:

$$\sum_{i \in S} auth(i) = 15; \sum_{i \in S} hub(i) = 15$$

The HITS Iterative Algorithm: Example

	Old Auth	Old Hub	New Auth	New Hub
A	1	1	3/15	1/15
B	1	1	2/15	2/15
C	1	1	5/15	1/15
D	1	1	2/15	2/15
E	1	1	1/15	4/15
F	1	1	1/15	2/15
G	1	1	0/15	2/15
H	1	1	1/15	1/15

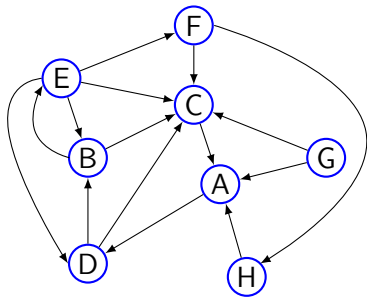


Normalize:

$$\sum_{i \in S} auth(i) = 15; \sum_{i \in S} hub(i) = 15$$

The HITS Iterative Algorithm: Example

	Old Auth	Old Hub	New Auth	New Hub
A	3/15	1/15		
B	2/15	2/15		
C	5/15	1/15		
D	2/15	2/15		
E	1/15	4/15		
F	1/15	2/15		
G	0/15	2/15		
H	1/15	1/15		

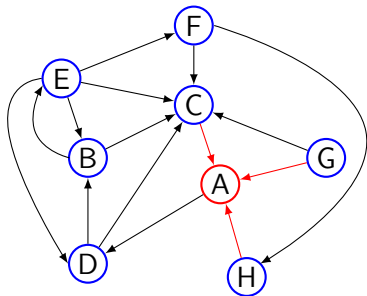


Normalize:

$$\sum_{i \in S} auth(i) = 15; \sum_{i \in S} hub(i) = 15$$

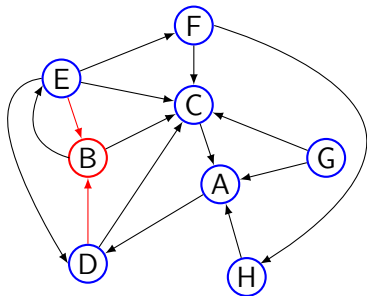
The HITS Iterative Algorithm: Example

	Old Auth	Old Hub	New Auth	New Hub
A	3/15	1/15	4/15	
B	2/15	2/15		
C	5/15	1/15		
D	2/15	2/15		
E	1/15	4/15		
F	1/15	2/15		
G	0/15	2/15		
H	1/15	1/15		



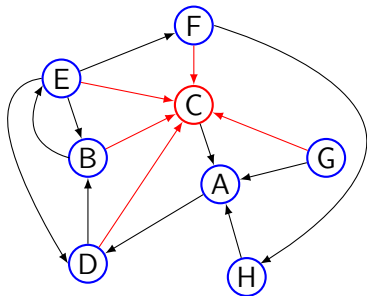
The HITS Iterative Algorithm: Example

	Old Auth	Old Hub	New Auth	New Hub
A	3/15	1/15	4/15	
B	2/15	2/15	6/15	
C	5/15	1/15		
D	2/15	2/15		
E	1/15	4/15		
F	1/15	2/15		
G	0/15	2/15		
H	1/15	1/15		



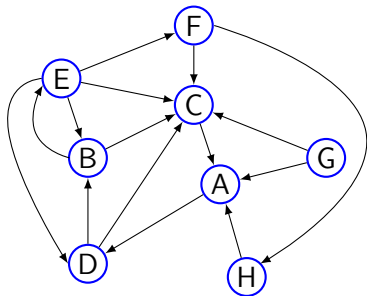
The HITS Iterative Algorithm: Example

	Old Auth	Old Hub	New Auth	New Hub
A	3/15	1/15	4/15	
B	2/15	2/15	6/15	
C	5/15	1/15	12/15	
D	2/15	2/15		
E	1/15	4/15		
F	1/15	2/15		
G	0/15	2/15		
H	1/15	1/15		



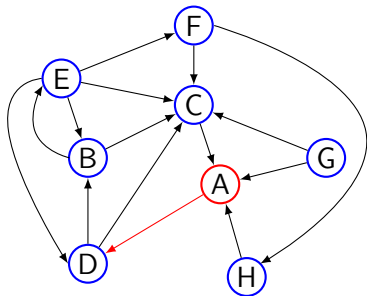
The HITS Iterative Algorithm: Example

	Old Auth	Old Hub	New Auth	New Hub
A	3/15	1/15	4/15	
B	2/15	2/15	6/15	
C	5/15	1/15	12/15	
D	2/15	2/15	5/15	
E	1/15	4/15	2/15	
F	1/15	2/15	4/15	
G	0/15	2/15	0/15	
H	1/15	1/15	2/15	



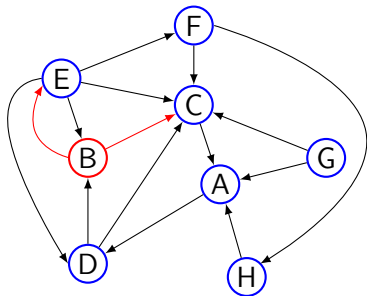
The HITS Iterative Algorithm: Example

	Old Auth	Old Hub	New Auth	New Hub
A	3/15	1/15	4/15	2/15
B	2/15	2/15	6/15	
C	5/15	1/15	12/15	
D	2/15	2/15	5/15	
E	1/15	4/15	2/15	
F	1/15	2/15	4/15	
G	0/15	2/15	0/15	
H	1/15	1/15	2/15	



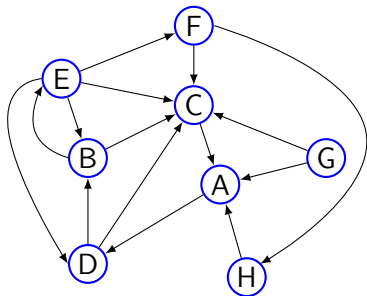
The HITS Iterative Algorithm: Example

	Old Auth	Old Hub	New Auth	New Hub
A	3/15	1/15	4/15	2/15
B	2/15	2/15	6/15	6/15
C	5/15	1/15	12/15	
D	2/15	2/15	5/15	
E	1/15	4/15	2/15	
F	1/15	2/15	4/15	
G	0/15	2/15	0/15	
H	1/15	1/15	2/15	



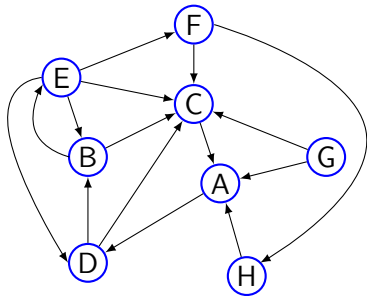
The HITS Iterative Algorithm: Example

	Old Auth	Old Hub	New Auth	New Hub
A	3/15	1/15	4/15	2/15
B	2/15	2/15	6/15	6/15
C	5/15	1/15	12/15	3/15
D	2/15	2/15	5/15	7/15
E	1/15	4/15	2/15	10/15
F	1/15	2/15	4/15	6/15
G	0/15	2/15	0/15	8/15
H	1/15	1/15	2/15	3/15



The HITS Iterative Algorithm: Example

	Old Auth	Old Hub	New Auth	New Hub
A	3/15	1/15	4/15	2/15
B	2/15	2/15	6/15	6/15
C	5/15	1/15	12/15	3/15
D	2/15	2/15	5/15	7/15
E	1/15	4/15	2/15	10/15
F	1/15	2/15	4/15	6/15
G	0/15	2/15	0/15	8/15
H	1/15	1/15	2/15	3/15

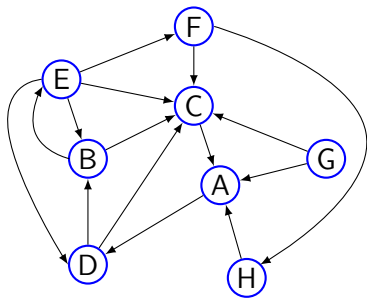


Normalize:

$$\sum_{i \in S} auth(i) = 35/15; \sum_{i \in S} hub(i) = 45/15 = 3$$

The HITS Iterative Algorithm: Example

	Old Auth	Old Hub	New Auth	New Hub
A	3/15	1/15	4/35	2/45
B	2/15	2/15	6/35	2/15
C	5/15	1/15	12/35	1/15
D	2/15	2/15	5/35	7/45
E	1/15	4/15	2/35	2/9
F	1/15	2/15	4/35	2/15
G	0/15	2/15	0/35	8/45
H	1/15	1/15	2/35	1/15



Normalize:

$$\sum_{i \in S} auth(i) = 35/15; \sum_{i \in S} hub(i) = 45/15 = 3$$

Convergence

What happens if we do this for larger and larger values of k ?

- Algorithm converges to a **fix-point** if iterated indefinitely
- In practice, 20 iterations produce fairly stable results
- Regardless of the **initial** hub and authority values (provided they are positive), we generally reach the same limiting values

Results

- Authorities for query: “Java”
 - java.sun.com
 - comp.lang.java FAQ

Finding Similar Pages Using Link Structure

- Given a page, p , let R (the root set) be t (e.g., 200) pages that point to p .
- Grow a base set S from R .
- Run HITS on S .
- Return the best authorities in S as the best similar-pages for p .
- Finds authorities in the “link neighborhood” of p .

Similar Page Results

- Given “honda.com”
 - toyota.com
 - ford.com
 - bmwusa.com
 - saturncars.com
 - nissanmotors.com
 - audi.com
 - volvocars.com