Graph-based Algorithms in NLP

Regina Barzilay

MIT

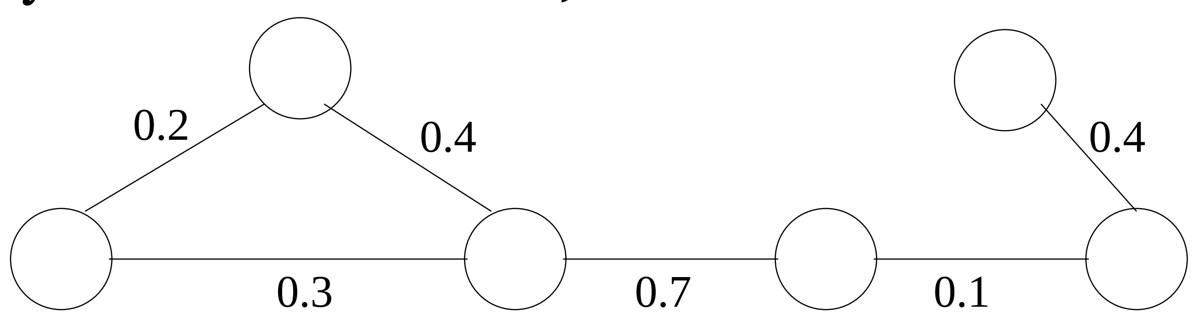
November, 2005

Graph-Based Algorithms in NLP

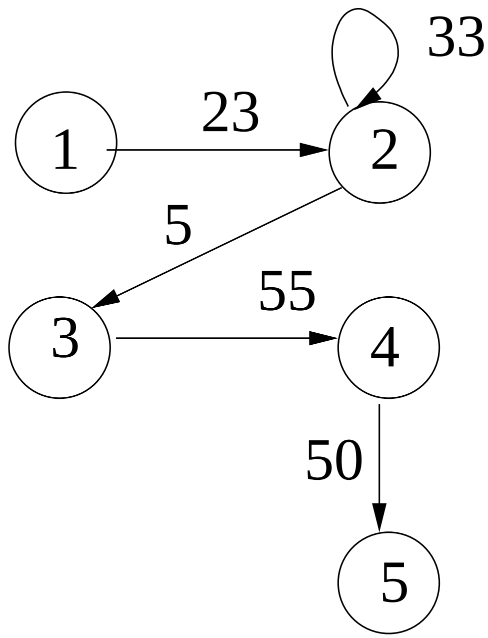
- In many NLP problems entities are connected by a range of relations
- Graph is a natural way to capture connections between entities
- Applications of graph-based algorithms in NLP:
  - Find entities that satisfy certain structural properties defined with respect to other entities
  - Find globally optimal solutions given relations between entities

Graph-based Representation

- Let $G(V, E)$ be a weighted undirected graph
  - V - set of nodes in the graph
  - E - set of weighted edges
- Edge weights $w(u, v)$ define a measure of pairwise similarity between nodes u,v



Graph-based Representation



	1	2	3	4	5
1		23			
2		33	5		
3				55	
4					50
5					

Examples of Graph-based Representations

Data	Directed?	Node	Edge
Web	yes	page	link
Citation Net	yes	citation	reference relation
Text	no	sent	semantic connectivity

Hubs and Authorities Algorithm (Kleinberg, 1998)

- **Application context:** information retrieval
- **Task:** retrieve documents relevant to a given query
- **Naive Solution:** text-based search
  - Some relevant pages omit query terms
  - Some irrelevant do include query terms

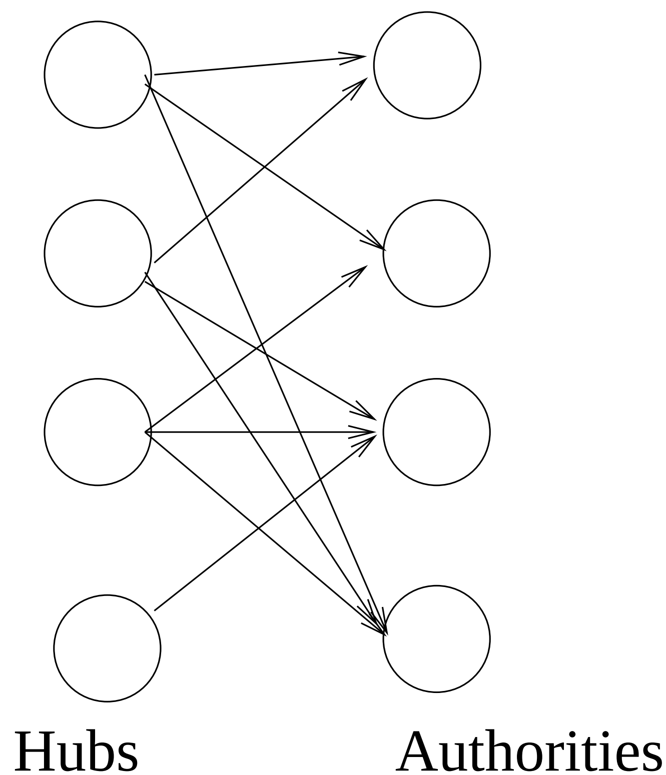
We need to take into account the authority of the page!

Analysis of the Link Structure

- **Assumption:** the creator of page p, by including a link to page q, has in some measure conferred authority in q
- **Issues to consider:**
  - some links are not indicative of authority (e.g., navigational links)
  - we need to find an appropriate balance between the criteria of relevance and popularity

Outline of the Algorithm

- Compute focused subgraphs given a query
- Iteratively compute hubs and authorities in the subgraph



Focused Subgraph

- Subgraph $G[W]$ over $W \subseteq V$, where edges correspond to all the links between pages in W
- How to construct G_σ for a string σ?
  - G_σ has to be relatively small
  - G_σ has to be rich in relevant pages
  - G_σ must contain most of the strongest authorities

Constructing a Focused Subgraph: Notations

Subgraph (σ, Eng, t, d)

σ: a query string

Eng: a text-based search engine

t, d: natural numbers

Let R_σ denote the top t results of Eng on σ

Constructing a Focused Subgraph: Algorithm

Set $S_c := R_\sigma$

For each page $p \in R_\sigma$

  Let $\Gamma^+(p)$ denote the set of all pages p points to

  Let $\Gamma^-(p)$ denote the set of all pages pointing to p

  Add all pages in $\Gamma^+(p)$ to S_σ

  If $|\Gamma^-(p)| \leq d$ then

    Add all pages in $|\Gamma^-(p)|$ to S_σ

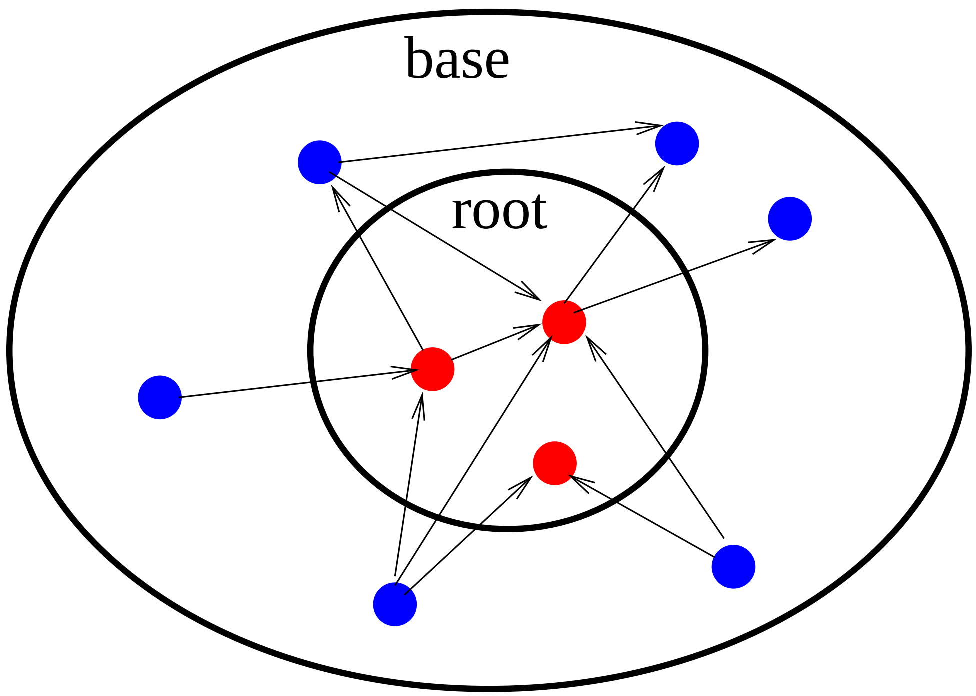
  Else

    Add an arbitrary set of d pages from $|\Gamma^-(p)|$ to S_σ

End

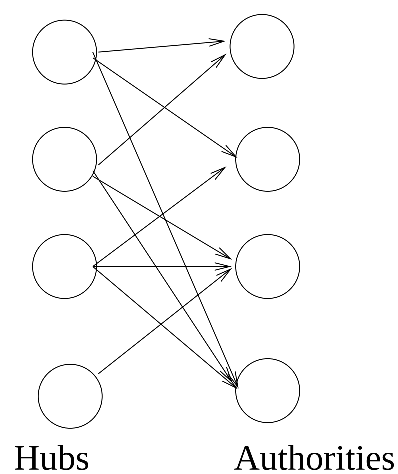
Return S_σ

Constructing a Focused Subgraph



Computing Hubs and Authorities

- Authorities should have considerable overlap in terms of pages pointing to them
- Hubs are pages that have links to multiple authoritative pages
- Hubs and authorities exhibit a mutually reinforcing relationship



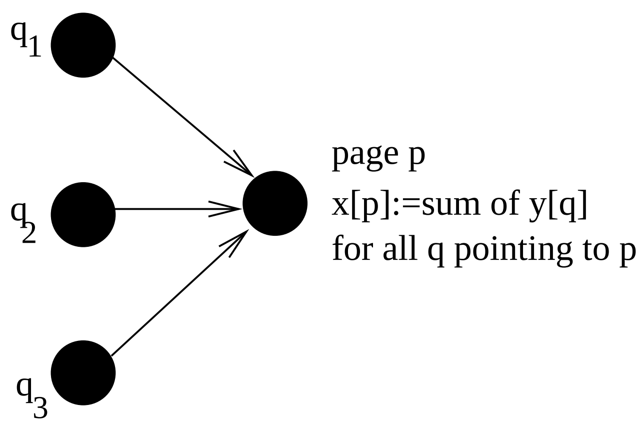
An Iterative Algorithm

- For each page p, compute authority weight $x^{(p)}$ and hub weight $y^{(p)}$
  - $x^{(p)} \geq 0$, $x^{(p)} \geq 0$
  - $\sum_{p \in s_\sigma} (x^{(p)})^2 = 1$, $\sum_{p \in s_\sigma} (y^{(p)})^2 = 1$
- Report top ranking hubs and authorities

I operation

Given $\{y^{(p)}\}$, compute:

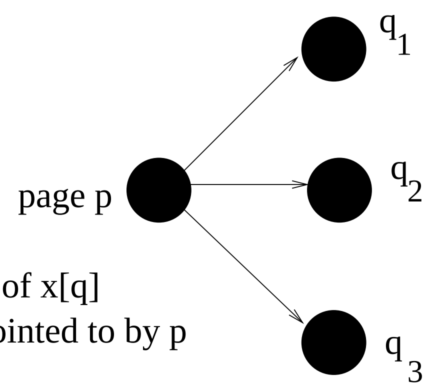
$$x^{(p)} \leftarrow \sum_{q:(q,p)\in E} y^{(p)}$$



O operation

Given $\{x^{(p)}\}$, compute:

$$y^{(p)} \leftarrow \sum_{q:(p,q)\in E} x^{(p)}$$



y[p]:=sum of x[q]
for all q pointed to by p

Algorithm:Iterate

Iterate (G,k)   G: a collection of n linked paged

k: a natural number

Let z denote the vector $(1, 1, 1, \ldots, 1) \in R^n$

Set $x_0 := z$

Set $y_0 := z$

For $i = 1, 2, \ldots, k$

Apply the I operation to (x_{i-1}, y_{i-1}), obtaining new x-weights x'_i

Apply the O operation to (x'_i, y_{i-1}), obtaining new y-weights y'_i

Normalize x'_i, obtaining x_i

Normalize y'_i, obtaining y_i

Return (x_k, y_k)

Algorithm: Filter

Filter (G,k,c)   G: a collection of n linked paged

k,c: natural numbers

$(x_k, y_k) := Iterate(G, k)$

Report the pages with the c largest coordinates in x_k as authorities

Report the pages with the c largest coordinates in y_k as hubs

Convergence

Theorem: The sequence x_1, x_2, x_3 and y_1, y_2, y_3 converge.

- Let A be the adjacency matrix of g_σ
- Authorities are computed as the principal eigenvector of $A^T A$
- Hubs are computed as the principal eigenvector of AA^T

Subgraph obtained from `www.honda.com`

`http://www.honda.com`	*Honda*
`http://www.ford.com`	*Ford Motor Company*
`http://www.eff.org/blueribbon.html`	*Campaign for Free Speech*
`http://www.mckinley.com`	*Welcome to Magellan!*
`http://www.netscape.com`	*Welcome to Netscape!*
`http://www.linkexchange.com`	*LinkExchange — Welcome*
`http://www.toyota.com`	*Welcome to Toyota*

Authorities obtained from `www.honda.com`

0.202	`http://www.toyota.com`	*Welcome to Toyota*
0.199	`http://www.honda.com`	*Honda*
0.192	`http://www.ford.com`	*Ford Motor Company*
0.173	`http://www.bmwusa.com`	*BMW of North America, Inc.*
0.162	`http://www.bmwusa.com`	*VOLVO*
0.158	`http://www.saturncars.com`	*Saturn Web Site*
0.155	`http://www.nissanmotors.com`	*NISSAN*

PageRank Algorithm (Brin&Page,1998)

Original Google ranking algorithm

- Similar idea to Hubs and Authorities
- Key differences:
  - Authority of each page is computed off-line
  - Query relevance is computed on-line
    - Anchor text
    - Text on the page
  - The prediction is based on the combination of authority and relevance

Intuitive Justification

From *The Anatomy of a Large-Scale Hypertextual Web Search Engine* (Brin&Page, 1998)

> PageRank can be thought of as a model of used behaviour. We assume there is a “random surfer” who is given a web page at random and keeps clicking on links never hitting “back” but eventually get bored and starts on another random page. The probability that the random surfer visists a page is its PageRank. And, the d damping factor is the probability at each page the “random surfer” will get bored and request another random page.

Brin, S., and L. Page. "The Anatomy of a Large-Scale Hypertextual Web Search Engine."
WWW7 / Computer Networks 30 no. 1-7 (1998): 107-117.
Paper available at http://dbpubs.stanford.edu:8090/pub/1998-8.

PageRank Computation

Iterate PR(p) computation:

pages $q_1, \ldots, q_n$ that point to page p

d is a damping factor (typically assigned to 0.85)

$C(p)$ is out-degree of p

$$PR(p) = (1-d) + d * (\frac{PR(q_1)}{C(q_1)} + \ldots + \frac{PR(q_n)}{C(q_n)})$$

Notes on PageRank

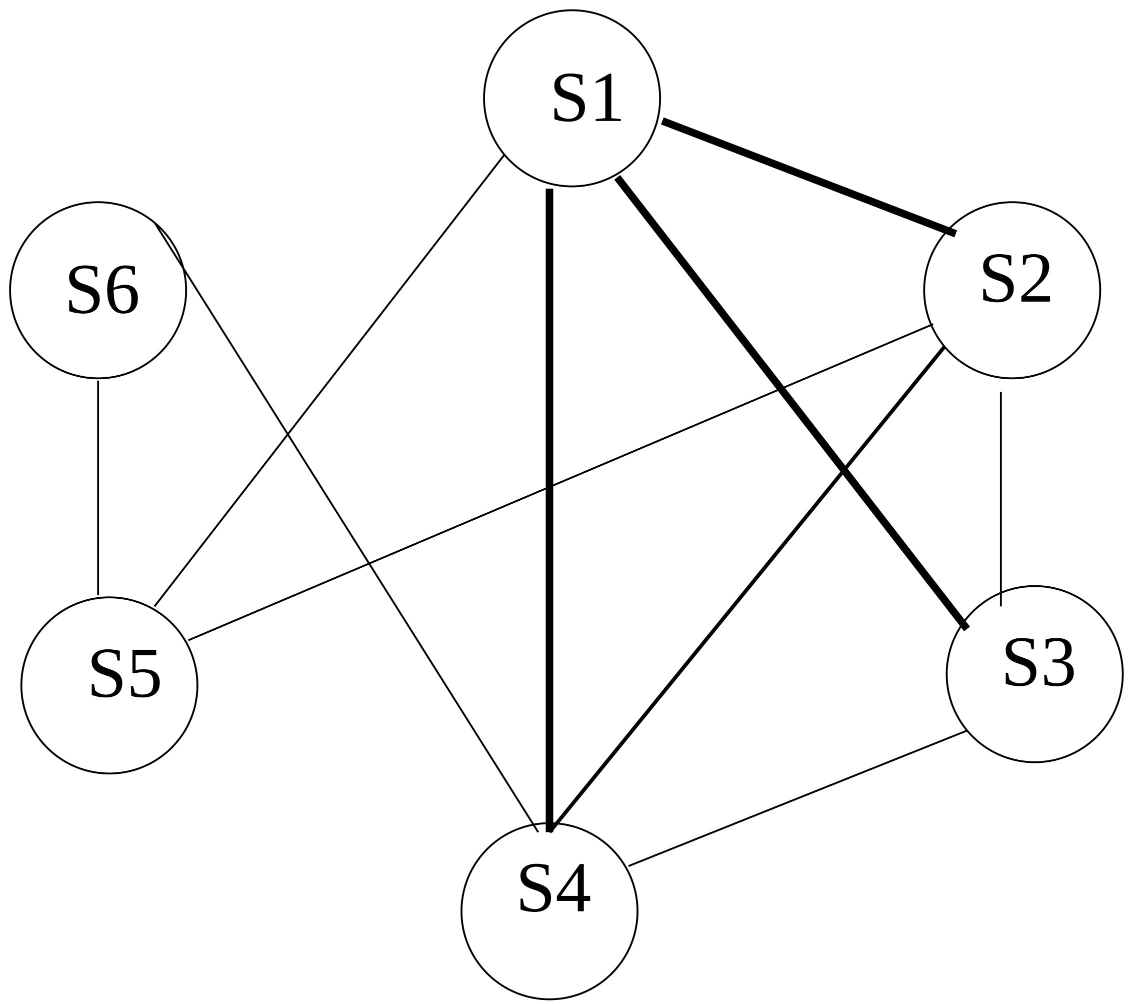
- PageRank forms a probability distribution over web pages
- PageRank corresponds to the principal eigenvector of the normalized link matrix of the web

Extractive Text Summarization

Task: Extract important information from a text

Figure removed for copyright reasons. Screenshots of several website text paragraphs.

Text as a Graph



Centrality-based Summarization(Radev)

- Assumption: The centrality of the node is an indication of its importance
- Representation: Connectivity matrix based on intra-sentence cosine similarity
- Extraction mechanism:
  - Compute PageRank score for every sentence u

    $$PageRank(u) = \frac{(1-d)}{N} + d \sum_{v \in adj[u]} \frac{PageRank(v)}{deg(v)}$$

    , where N is the number of nodes in the graph
  - Extract k sentences with the highest PageRanks score

Does it work?

- Evaluation: Comparison with human created summary
- Rouge Measure: Weighted n-gram overlap (similar to Bleu)

Method	Rouge score
Random	0.3261
Lead	0.3575
Degree	0.3595
PageRank	0.3666

Does it work?

- Evaluation: Comparison with human created summary
- Rouge Measure: Weighted n-gram overlap (similar to Bleu)

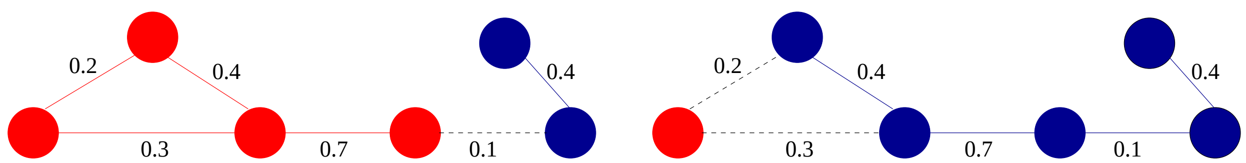
Method	**Rouge score**
Random	0.3261
Lead	0.3575
Degree	0.3595
PageRank	0.3666

Graph-Based Algorithms in NLP

- Applications of graph-based algorithms in NLP:
  - Find entities that satisfy certain structural properties defined with respect to other entities
  - Find globally optimal solutions given relations between entities

Min-Cut: Definitions

- Graph cut: partitioning of the graph into two disjoint sets of nodes A,B
- Graph cut weight: cut$(A, B) = \sum_{u \in A, v \in B} w(u, v)$
  - i.e. sum of crossing edge weights
- Minimum Cut: the cut that minimizes cross-partition similarity



Finding Min-Cut

- The problem is polynomial time solvable for 2-class min-cut when the weights are positive
  - Use max-flow algorithm
- In general case, $k - way$ cut is NP-complete.
  - Use approximation algrorithms (e.g., randomized algorithm by Karger)

MinCut first used for NLP applications by Pang&Lee'2004 (sentiment classification)

Min-Cut for Content Selection

Task: Determine a subset of database entries to be included in the generated document

TEAM STAT COMPARISON

	Oakland Raiders	New England Patriots
1st Downs	19	22
Total Yards	338	379
Passing	246	306
Rushing	92	73
Penalties	16-149	7-46
3rd Down Conversions	4-13	6-16
4th Down Conversions	0-0	0-1
Turnovers	2	0
Possession	27:40	32:20

INDIVIDUAL LEADERS

Oakland Passing				
	C/ATT	YDS	TD	INT
Collins	18/39	265	3	0

New England Passing				
	C/ATT	YDS	TD	INT
Brady	24/38	306	2	0

Oakland Rushing				
	CAR	YDS	TD	LG
Jordan	18	17	0	14
Crockett	3	20	8	19

New England Rushing				
	CAR	YDS	TD	LG
Dillon	23	63	2	10
Faulk	5	11	0	4

Oakland Receiving				
	REC	YDS	TD	LG
Moss	5	130	1	73
Porter	3	48	0	27

New England Receiving				
	REC	YDS	TD	LG
Branch	7	99	1	29
Watson	2	55	0	35

Figure by MIT OCW.

Parallel Corpus for Text Generation

Passing					
PLAYER	CP/AT	YDS	AVG	TD	INT
Brunell	17/38	192	6.0	0	0
Garcia	14/21	195	9.3	1	0
...	...	...	...	...	...

Rushing					
PLAYER	REC	YDS	AVG	LG	TD
Suggs	22	82	3.7	25	1
...	...	...	...	...	...

Fumbles				
PLAYER	FUM	LOST	REC	YDS
Coles	1	1	0	0
Portis	1	1	0	0
Davis	0	0	1	0
Little	0	0	1	0
...	...	...	...	...

Suggs rushed for 82 yards and scored a touchdown in the fourth quarter, leading the Browns to a 17-13 win over the Washington Redskins on Sunday. **Jeff Garcia went 14-of-21 for 195 yards and a TD** for the Browns, who didn't secure the win until **Coles fumbled** with 2:08 left. The Redskins (1-3) can pin their third straight loss on going just 1-for-11 on third downs, mental mistakes and **a costly fumble by Clinton Portis**. **"My fumble changed the momentum", Portis said.** **Brunell finished 17-of-38 for 192 yards**, but was unable to get into any rhythm because Cleveland's defense shut down Portis. The Browns faked a field goal, but holder Derrick Frost was stopped short of a first down. **Brunell then completed a 13-yard pass to Coles, who fumbled** as he was being taken down and Browns safety Earl Little recovered.

Content Selection: Problem Formulation

- Input format: a set of entries from a relational database
  - "entry"="raw in a database"
- Training: n sets of database entries with associated selection labels

Oakland Rushing				
	CAR	YDS	TD	LG
Jordan	18	17	0	14
Crockett	3	20	8	19

Figure by MIT OCW.

- Testing: predict selection labels for a new set of entries

Simple Solution

Formulate content selection as a classification task:

- **Prediction: {1,0}**
- **Representation of the problem:**

Player	**YDS**	**LG**	**TD**	**Selected**
Dillon	63	10	2	1
Faulk	11	4	0	0

Goal: Learn classification function $P(Y|X)$ that can classify unseen examples

$X = \langle Smith, 28, 9, 1 \rangle \quad Y_1 = ?$

Potential Shortcoming: Lack of Coherence

- Sentences are classified in isolation
- Generated sentences may not be connected in a meaningful way

Example: An output of a system that automatically generates scientific papers (Stribling et al., 2005):

> Active networks and virtual machines have a long history of collaborating in this manner. The basic tenet of this solution is the refinement of Scheme. The disadvantage of this type of approach, however, is that public-private key pair and red-black trees are rarely incompatible.

Enforcing Output Coherence

Sentences in a text are connected

The New England Patriots squandered a couple big leads. That was merely a setup for Tom Brady and Adam Vinatieri, who pulled out one of their typical last-minute wins.
Brady threw for 350 yards and three touchdowns before Vinatieri kicked a 29-yard field goal with 17 seconds left to lead injury-plagued New England past the Atlanta Falcons 31-28 on Sunday.

Simple classification approach cannot enforce coherence constraints

Constraints for Content Selection

Collective content selection: consider all the entries simultaneously

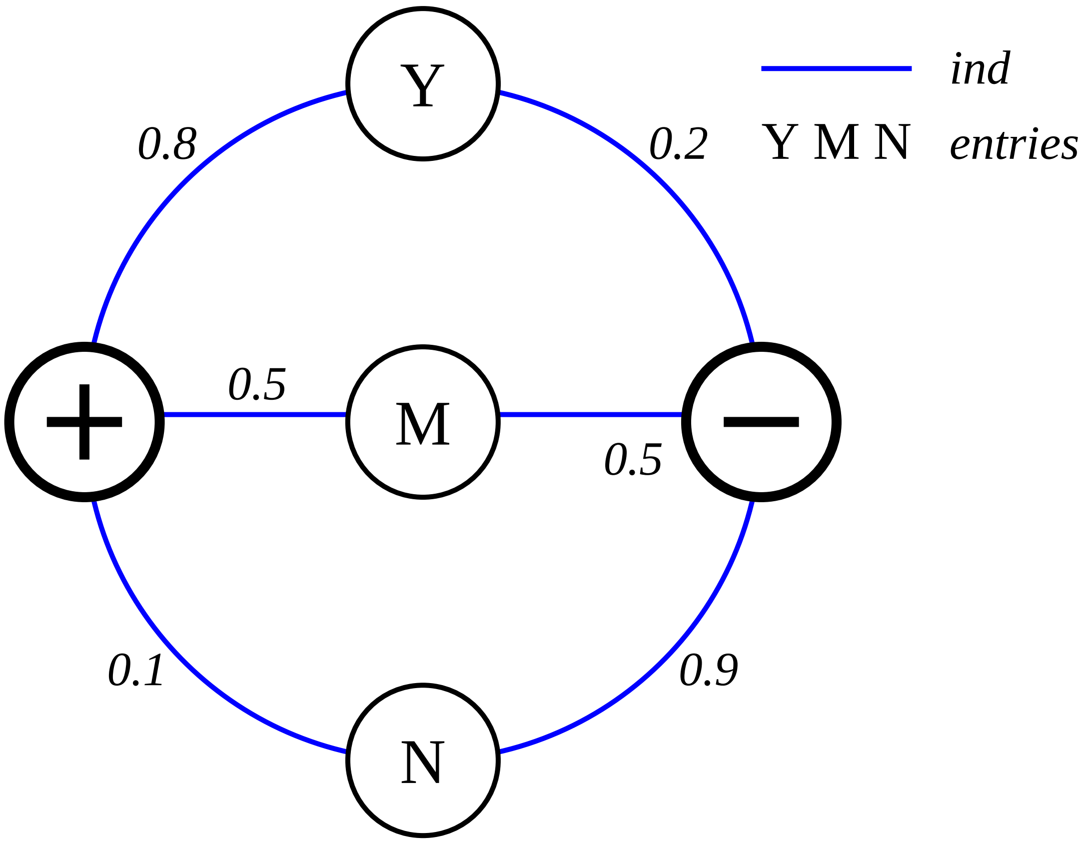
- Individual constraints:

3	Branch scores TD	7	10

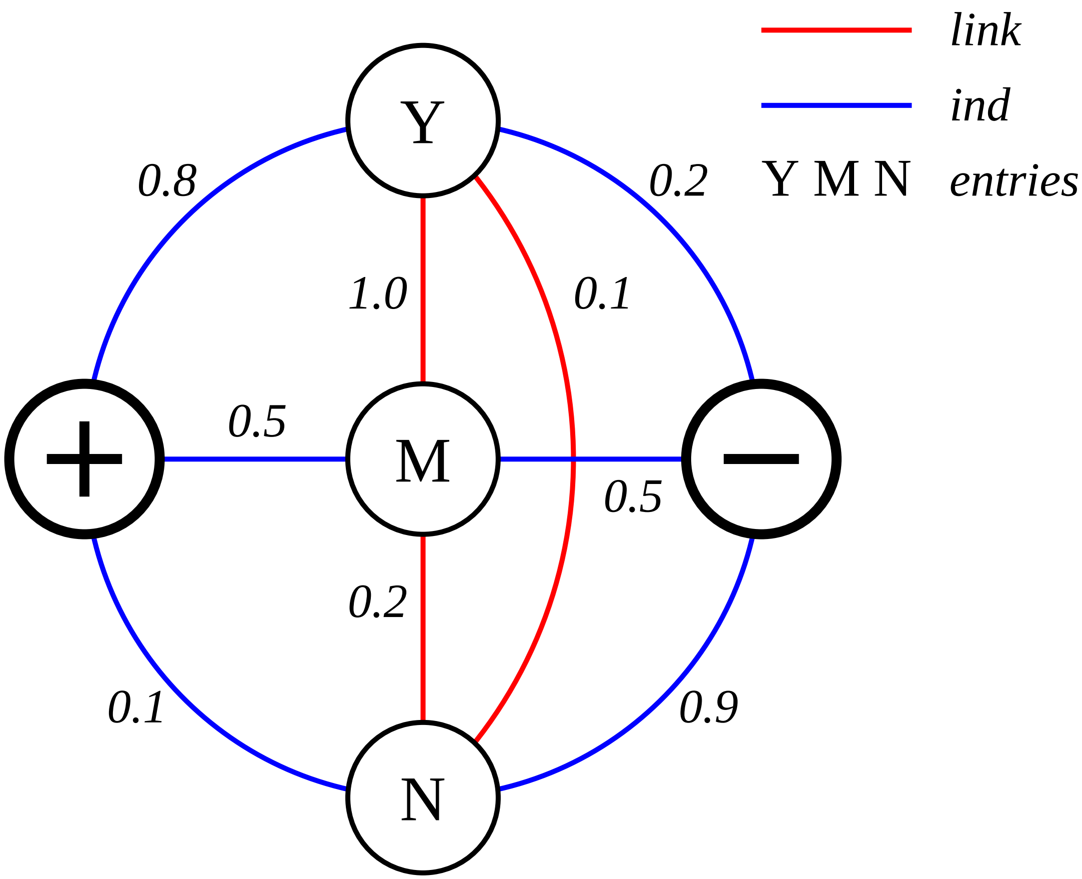
- Contextual constraints:

3	Brady passes to Branch	7	3
3	Branch scores TD	7	10

Individual Preferences



Combining Individual and Contextual Preferences



Collective Classification

$x \in C_+$	selected entities
$ind_+(x)$	preference to be selected
$link_L(x_i, x_j)$	x_i and x_j are connected by link of type L

Minimize penalty:

$$\sum_{x \in C_+} ind_-(x) + \sum_{x \in C_-} ind_+(x) + \sum_{L} \sum_{\substack{x_i \in C_+ \\ x_j \in C_-}} link_L(x_i, x_j)$$

Goal: Find globally optimal label assignment

Optimization Framework

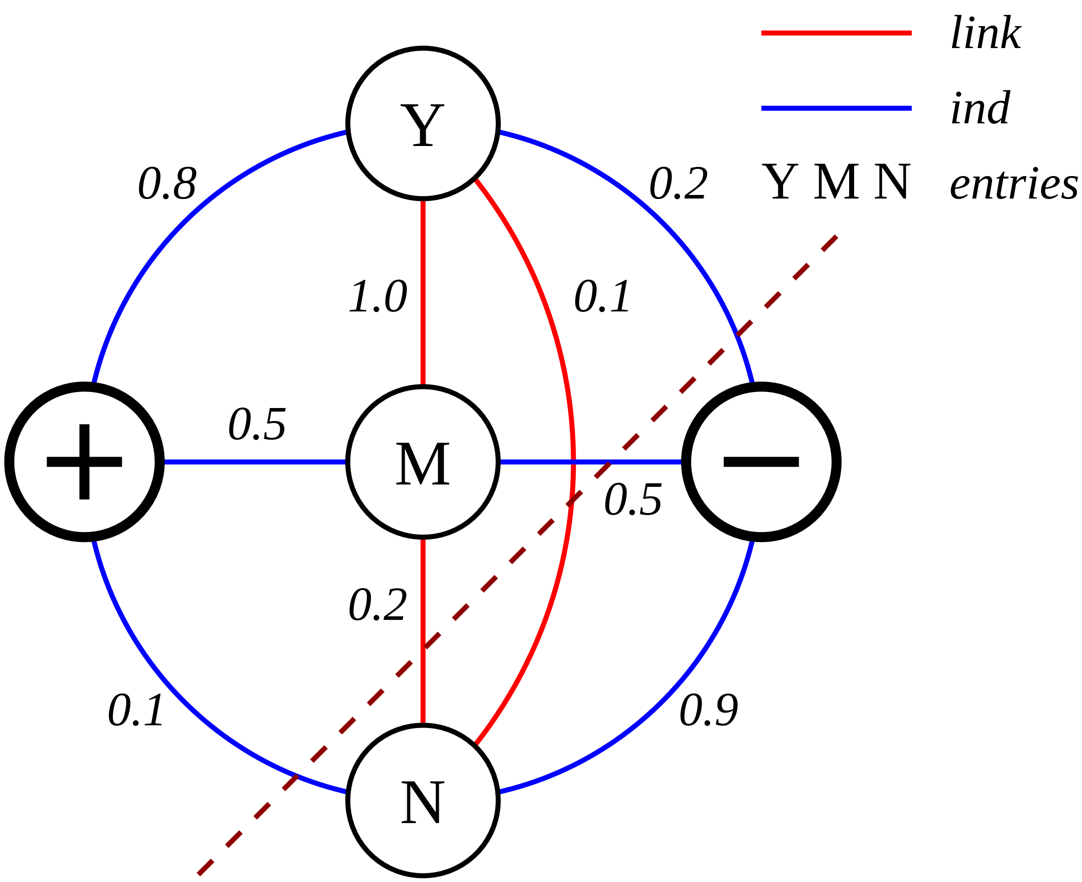
$$\sum_{x \in C_+} ind_-(x) + \sum_{x \in C_-} ind_+(x) + \sum_{L} \sum_{\substack{x_i \in C_+ \\ x_j \in C_-}} link_L(x_i, x_j)$$

Energy minimization framework (Besag, 1986, Pang&Lee, 2004)

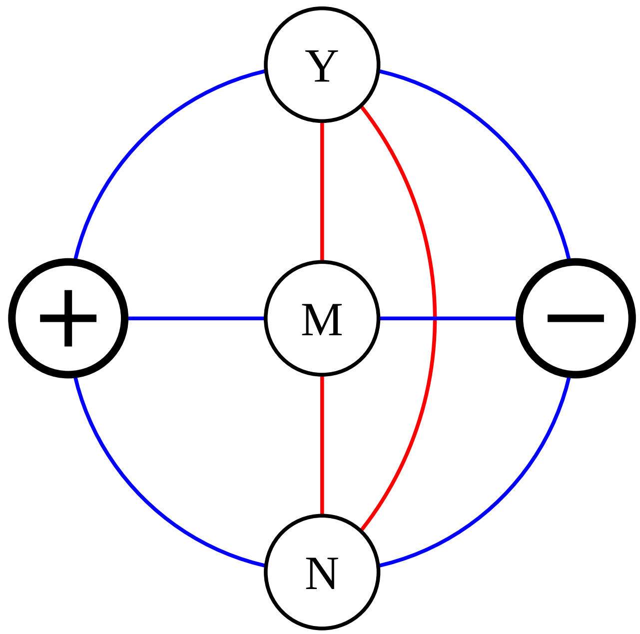
- Seemingly intractable
- Can be solved exactly in polynomial time (scores are positive) (Greig et al., 1989)

Graph-Based Formulation

Use max-flow to compute minimal cut partition



Learning Task



- Learning individual preferences
- Learning link structure

Learning Individual Preferences

- Map attributes of a database entry to a feature vector

Oakland Rushing				
	CAR	YDS	TD	LG
Jordan	18	17	0	14
Crockett	3	20	8	19

Figure by MIT OCW.

X=<Jordan, 18, 17, 0, 14>, Y=1
X=<Crockett, 3, 20, 8, 19>, Y=0

- Train a classifier to learn $D(Y|X)$

Contextual Constraints: Learning Link Structure

- Build on rich structural information available in database schema
  - Define entry links in terms of their database relatedness
    Players from the winning team that had touchdowns in the same quarter
- Discover links automatically
  - Generate-and-prune approach

Construction of Candidate Links

- Link space:
  - Links based on attribute sharing
- Link type template:
  create $L_{i,j,k}$ for every entry type E_i and E_j, and for every shared attribute k

  $E_i = \texttt{Rushing}$, $E_j = \texttt{Passing}$, and $k = \texttt{Name}$

  $E_i = \texttt{Rushing}$, $E_j = \texttt{Passing}$, and $k = \texttt{TD}$

Link Filtering

$E_i = \texttt{Rushing}$, $E_j = \texttt{Passing}$, and $k = \texttt{Name}$

$E_i = \texttt{Rushing}$, $E_j = \texttt{Passing}$, and $k = \texttt{TD}$

New England Passing					
	C/ATT	YDS	AVG	TD	INT
T. Brady	24/38	306	8.1	2	0

New England Rushing					
	CAR	YDS	AVG	TD	LG
C. Dillon	23	63	2.7	2	10
K. Faulk	5	11	2.2	0	4
T. Brady	3	-1	-0.3	0	0
Team	**31**	**73**	**2.4**	**2**	**10**

New England Passing					
	C/ATT	YDS	AVG	TD	INT
T. Brady	24/38	306	8.1	2	0

New England Rushing					
	CAR	YDS	AVG	TD	LG
C. Dillon	23	63	2.7	2	10
K. Faulk	5	11	2.2	0	4
T. Brady	3	-1	-0.3	0	0
Team	**31**	**73**	**2.4**	**2**	**10**

Figure by MIT OCW.

Link Filtering

$E_i = \texttt{Rushing}$, $E_j = \texttt{Passing}$, and $k = \texttt{Name}$

$E_i = \texttt{Rushing}$, $E_j = \texttt{Passing}$, and $k = \texttt{TD}$

New England Passing					
	C/ATT	YDS	AVG	TD	INT
T. Brady	24/38	306	8.1	2	0

New England Rushing					
	CAR	YDS	AVG	TD	LG
C. Dillon	23	63	2.7	2	10
K. Faulk	5	11	2.2	0	4
T. Brady	3	-1	-0.3	0	0
Team	**31**	**73**	**2.4**	**2**	**10**

New England Passing					
	C/ATT	YDS	AVG	TD	INT
T. Brady	24/38	306	8.1	2	0

New England Rushing					
	CAR	YDS	AVG	TD	LG
C. Dillon	23	63	2.7	2	10
K. Faulk	5	11	2.2	0	4
T. Brady	3	-1	-0.3	0	0
Team	**31**	**73**	**2.4**	**2**	**10**

Figure by MIT OCW.

Link Filtering

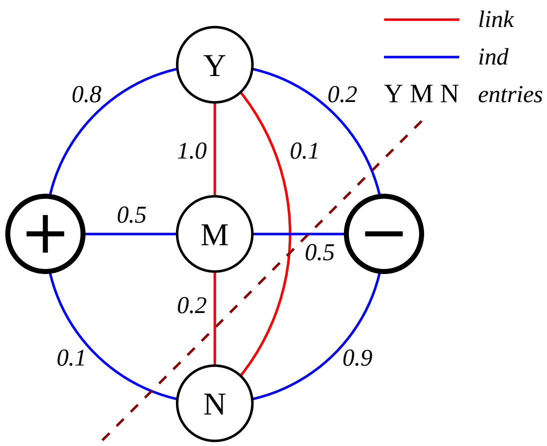
$E_i =$ `Rushing`, $E_j =$ `Passing`, and $k =$ `Name`

$E_i =$ `Rushing`, $E_j =$ `Passing`, and $k =$ `TD`

Measure similarity in label distribution using χ^2 test

- Assume H_0: labels of entries are independent
- Consider the joint label distribution of entry pairs from the training set
- H_0 is rejected if $\chi^2 > \tau$

Collective Content Selection



- Learning
  - Individual preferences
  - Link structure
- Inference
  - Minimal Cut Partitioning

Data

- Domain: American Football
- Data source: the official site of NFL
- Corpus: AP game recaps with corresponding databases for 2003 and 2004 seasons
  - Size: 468 recaps (436,580 words)
  - Average recap length: 46.8 sentences

Data: Preprocessing

- Anchor-based alignment (Duboue &McKeown, 2001, Sripada et al., 2001)
  - 7,513 aligned pairs
  - 7.1% database entries are verbalized
  - 31.7% sentences had a database entry
- Overall: 105, 792 entries
  - Training/Testing/Development: 83%, 15%, 2%

Results: Comparison with Human Extraction

- Precision (P): the percentage of extracted entries that appear in the text
- Recall (R): the percentage of entries appearing in the text that are extracted by the model
- F-measure: $F = 2\frac{PR}{(P+R)}$

Method	P	R	F
Previous Methods			
Class Majority Baseline	29.4	68.19	40.09
Standard Classifier	44.88	62.23	49.75
Collective Model	52.71	76.50	60.15

Summary

- Graph-based Algorithms: Hubs and Authorities, Min-Cut
- Applications: information Retrieval, Summarization, Generation