Clustering using Topic Models

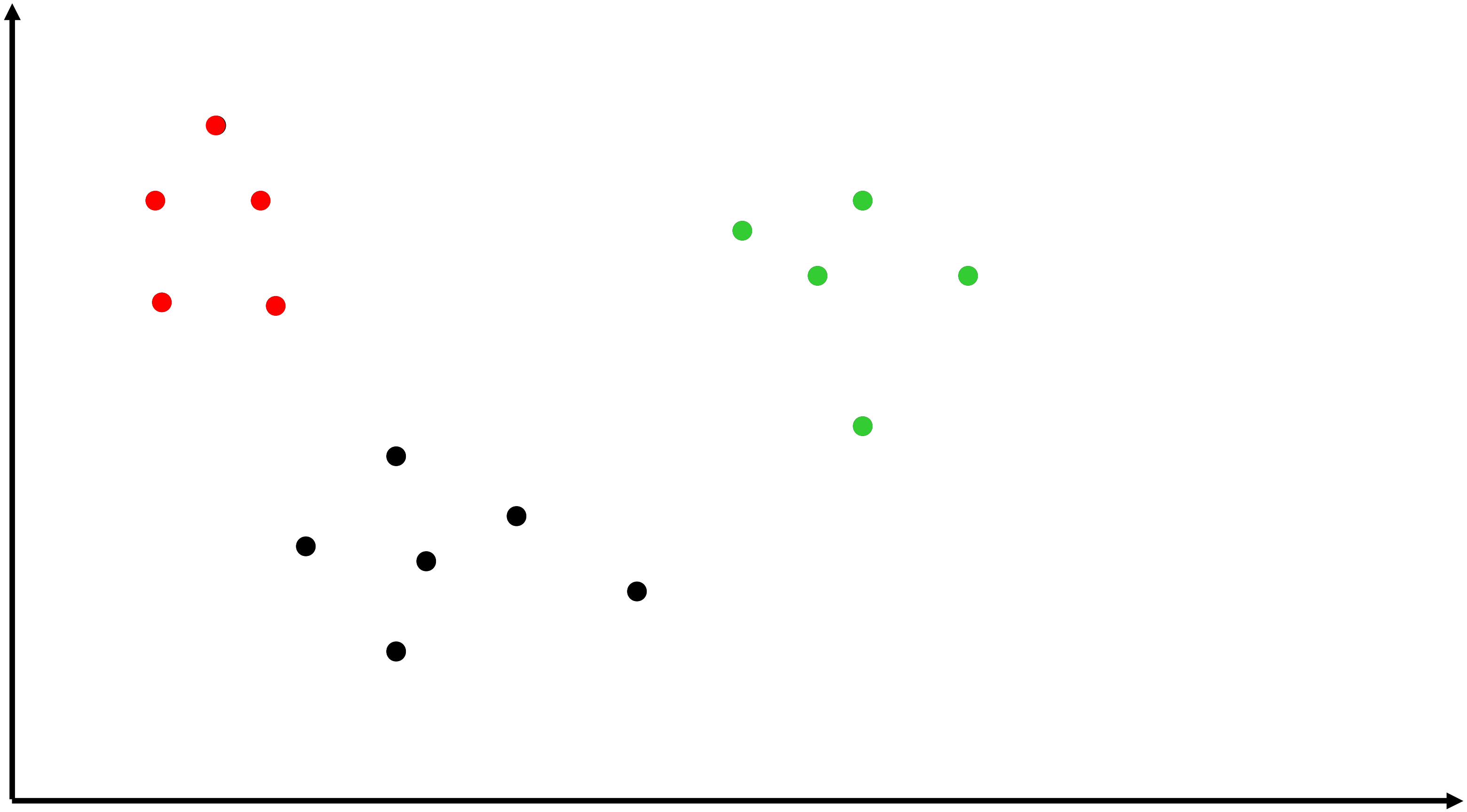
Compiled by Sujatha Das, Cornelia Caragea

Credits for slides: Blei, Allan, Arms, Manning, Rai, Lund, Noble, Page.

Clustering

- Partition unlabeled examples into disjoint subsets of *clusters*, such that:
  - Examples within a cluster are very similar
  - Examples in different clusters are very different
- Discover new categories in an *unsupervised* manner
- We don't know the categories apriori, the goal is to identify if there are any underlying “similar” groups

“Hard” Clustering Example



Soft clustering for document collections

- Suppose the clusters correspond to topics in a given document collection
- *Soft clustering* gives probabilities that an instance belongs to each of a set of clusters rather than assign a specific cluster to it
- Each instance is assigned a probability distribution across a set of discovered categories
  - A document is 80% politics and 20% religion

K-means vs. Topic Models

- K-means
  - Works on “points” in space
  - Performance depends on a similarity function between the points
  - Does not automatically give a generative view of documents. Probabilistically explain
    - What an “average” document on a “topic” look like?
    - What are the most likely words for a topic?
  - What if when multiple “modes” are available? For example, how to fold in citation links into clustering?

Topic Modeling

- From David Blei' page (Latent Dirichlet Allocation, a popularly used topic modeling algorithm)

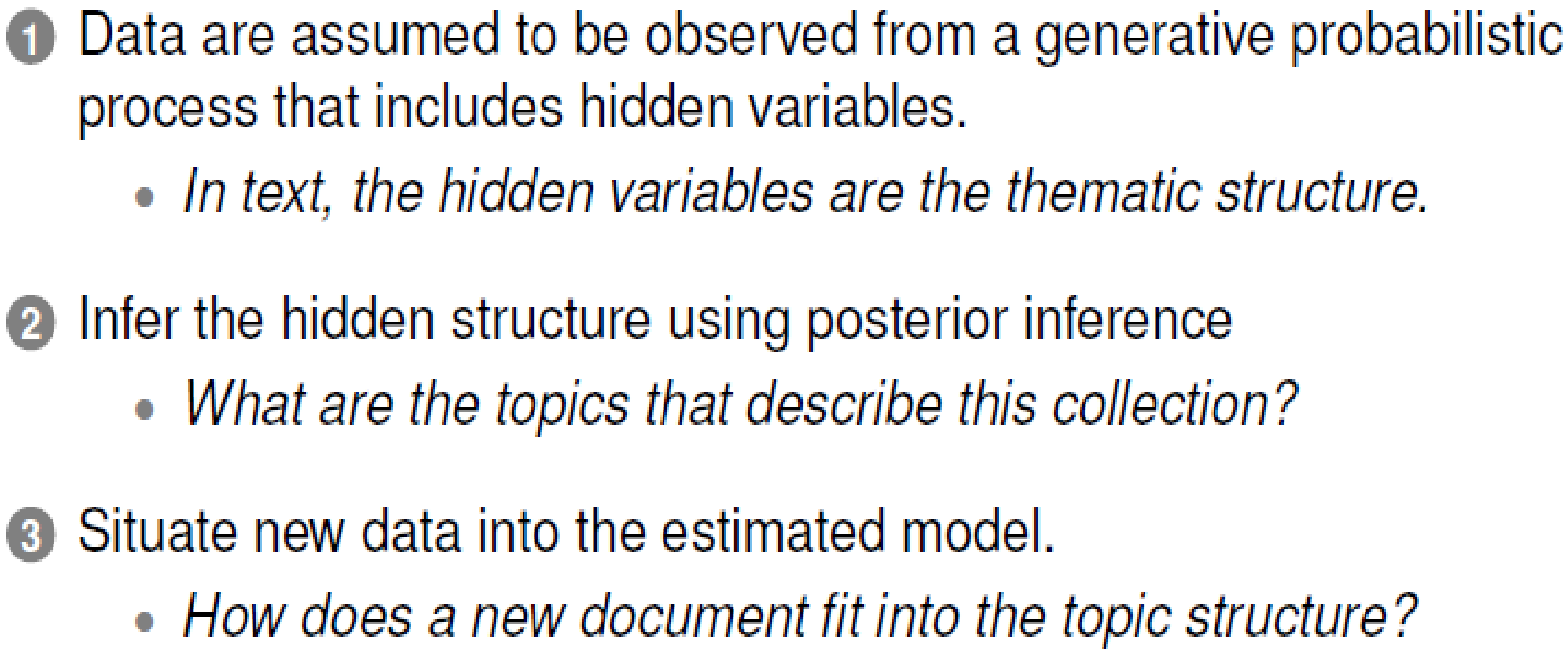
  "Topic models are a suite of algorithms that uncover the hidden thematic structure in document collections. These algorithms help us develop new ways to search, browse and summarize large archives of texts"

  - http://www.cs.princeton/~blei/topicmodeling.html
  - http://www.cs.princeton.edu/~blei/kdd-tutorial.pdf

Topic Modeling

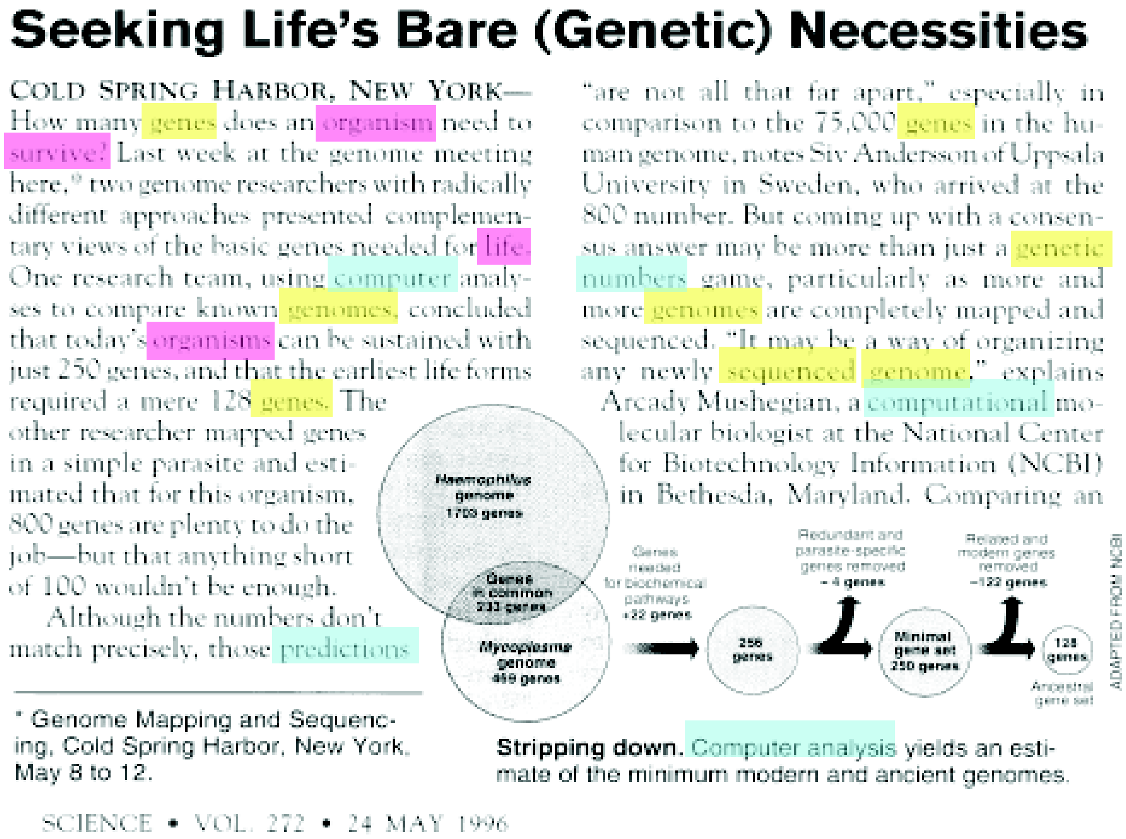
- Over the last decade, topic models (pLSA, PCA, LDA) were extensively studied for various applications
  - Discovering topics from a corpus
  - Evolution of topics with time
  - Model connections between topics
  - Find hierarchies of topics
  - Model influential articles/authors
  - Predict links between articles
  - Organize and browse large corpora

Probabilistic Modeling



1. Data are assumed to be observed from a generative probabilistic process that includes hidden variables.
   - *In text, the hidden variables are the thematic structure.*
2. Infer the hidden structure using posterior inference
   - *What are the topics that describe this collection?*
3. Situate new data into the estimated model.
   - *How does a new document fit into the topic structure?*

Latent Dirichlet allocation (LDA)



Seeking Life's Bare (Genetic) Necessities

COLD SPRING HARBOR, NEW YORK—How many genes does an organism need to survive? Last week at the genome meeting here,[*] two genome researchers with radically different approaches presented complementary views of the basic genes needed for life. One research team, using computer analyses to compare known genomes, concluded that today's organisms can be sustained with just 250 genes, and that the earliest life forms required a mere 128 genes. The other researcher mapped genes in a simple parasite and estimated that for this organism, 800 genes are plenty to do the job—but that anything short of 100 wouldn't be enough.

Although the numbers don't match precisely, those predictions "are not all that far apart," especially in comparison to the 75,000 genes in the human genome, notes Siv Andersson of Uppsala University in Sweden, who arrived at the 800 number. But coming up with a consensus answer may be more than just a genetic numbers game, particularly as more and more genomes are completely mapped and sequenced. "It may be a way of organizing any newly sequenced genome," explains Arcady Mushegian, a computational molecular biologist at the National Center for Biotechnology Information (NCBI) in Bethesda, Maryland. Comparing an

Haemophilus genome 1703 genes
Genes in common 233 genes
Mycoplasma genome 469 genes
Genes needed for biochemical pathways +22 genes
256 genes
Redundant and parasite-specific genes removed −4 genes
Minimal gene set 250 genes
Related and modern genes removed −122 genes
128 genes
Ancestral gene set

ADAPTED FROM NCBI

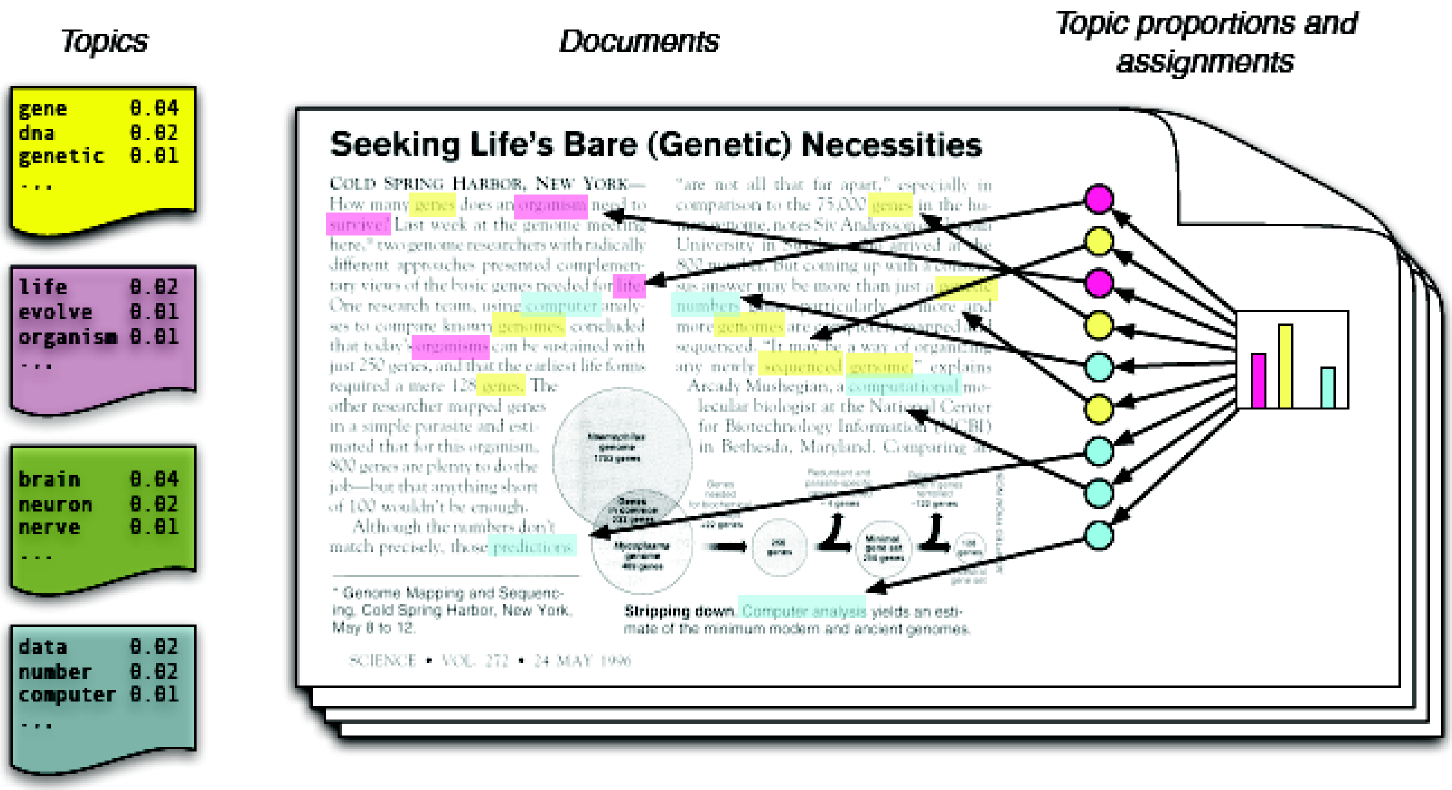
Stripping down. Computer analysis yields an estimate of the minimum modern and ancient genomes.

[*] Genome Mapping and Sequencing, Cold Spring Harbor, New York, May 8 to 12.

SCIENCE • VOL. 272 • 24 MAY 1996

Simple intuition: Documents exhibit multiple topics.

Generative model for LDA



Seeking Life's Bare (Genetic) Necessities

COLD SPRING HARBOR, NEW YORK—How many genes does an organism need to survive? Last week at the genome meeting here,* two genome researchers with radically different approaches presented complementary views of the basic genes needed for life. One research team, using computer analyses to compare known genomes, concluded that today's organisms can be sustained with just 250 genes, and that the earliest life forms required a mere 128 genes. The other researcher mapped genes in a simple parasite and estimated that for this organism, 800 genes are plenty to do the job—but that anything short of 100 wouldn't be enough.

Although the numbers don't match precisely, those predictions "are not all that far apart," especially in comparison to the 75,000 genes in the human genome, notes Siv Andersson of Uppsala University in Sweden, who arrived at the 800 number. But coming up with a consensus answer may be more than just a genetic numbers game, particularly as more and more genomes are completely mapped and sequenced. "It may be a way of organizing any newly sequenced genome," explains Arcady Mushegian, a computational molecular biologist at the National Center for Biotechnology Information (NCBI) in Bethesda, Maryland. Comparing an

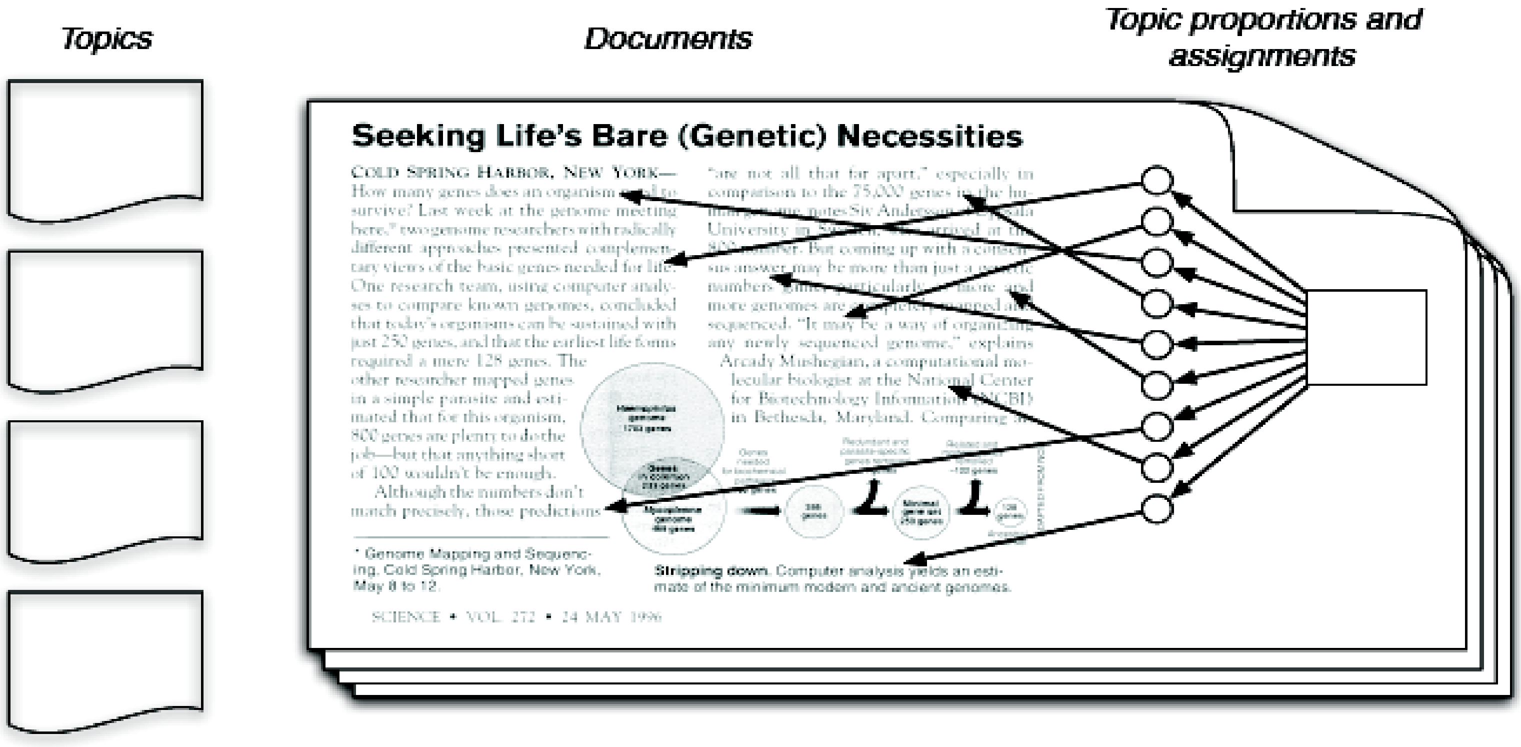
Stripping down. Computer analysis yields an estimate of the minimum modern and ancient genomes.

* Genome Mapping and Sequencing, Cold Spring Harbor, New York, May 8 to 12.

SCIENCE • VOL. 272 • 24 MAY 1996

- Each **topic** is a distribution over words
- Each **document** is a mixture of corpus-wide topics
- Each **word** is drawn from one of those topics

The posterior distribution


Topics
Documents
Topic proportions and assignments
Haemophilus genome 1703 genes
Genes in common 233 genes
Mycoplasma genome 468 genes
256 genes
Minimal gene set 250 genes
128 genes

Seeking Life's Bare (Genetic) Necessities

COLD SPRING HARBOR, NEW YORK— How many genes does an organism need to survive? Last week at the genome meeting here,* two genome researchers with radically different approaches presented complementary views of the basic genes needed for life. One research team, using computer analyses to compare known genomes, concluded that today's organisms can be sustained with just 250 genes, and that the earliest life forms required a mere 128 genes. The other researcher mapped genes in a simple parasite and estimated that for this organism, 800 genes are plenty to do the job—but that anything short of 100 wouldn't be enough.

Although the numbers don't match precisely, those predictions "are not all that far apart," especially in comparison to the 75,000 genes in the human genome, notes Siv Andersson of Uppsala University in Sweden, who arrived at the 800 number. But coming up with a consensus answer may be more than just a genetic numbers game, particularly as more and more genomes are completely mapped and sequenced. "It may be a way of organizing any newly sequenced genome," explains Arcady Mushegian, a computational molecular biologist at the National Center for Biotechnology Information (NCBI) in Bethesda, Maryland. Comparing an

Stripping down. Computer analysis yields an estimate of the minimum modern and ancient genomes.

* Genome Mapping and Sequencing, Cold Spring Harbor, New York, May 8 to 12.

SCIENCE • VOL. 272 • 24 MAY 1996

- In reality, we only observe the documents
- The other structure are **hidden variables**
  - Our goal is to **infer** the hidden variables
  - I.e., compute their distribution conditioned on the documents

$$p(\text{topics, proportions, assignments} \mid \text{documents})$$

Implementations of LDA

There are many available implementations of topic modeling—

LDA-C*	A C implementation of LDA
HDP*	A C implementation of the HDP (“infinite LDA”)
Online LDA*	A python package for LDA on massive data
LDA in R*	Package in R for many topic models
LingPipe	Java toolkit for NLP and computational linguistics
Mallet	Java toolkit for statistical NLP
TMVE*	A python package to build browsers from topic models

* available at www.cs.princeton.edu/~blei/

LDA is extendible

- LDA can be **embedded in more complicated models**, embodying further intuitions about the structure of the texts.

- LDA models can include syntax, authorship, word sense, dynamics, correlation, hierarchies, ...

- The **data generating distribution** can be changed.

- LDA models can be built for images, social networks, music, purchase histories, computer code, genetic data, click-through-data, neural spike trains, ...

- The **LDA posterior** can be used in creative ways

- It can be used for information retrieval, collaborative filtering, document similarity, visualization, ...

Understanding the output from basic LDA

Given a collection of documents, the basic LDA model uses the latent topic and term co-occurrences to estimate two quantities

- Topic-term matrix capturing term distributions for a topic

w1 w2 w3 ….

T1: 0.001 0.02 0.4....

- Top words for a topic "reveal" the theme captured by that topic for human consumption

- Topic proportions for each document in the corpus

t1 t2 t3 t4

D1: <0.8, 0.2, 0, 0 >

Understanding the output from basic LDA

Given a collection of documents, the basic LDA model uses the latent topic and term co-occurrences to estimate two quantities

- Topic-term matrix capturing term distributions for a topic

w1 w2 w3 ….

T1: 0.001 0.02 0.4....

- Top words for a topic "reveal" the theme captured by that topic for human consumption (low-dimensional projection)

- Topic proportions for each document in the corpus

t1 t2 t3 t4

D1: <0.8, 0.2, 0, 0 >

Unsupervised learning in CiteSeer

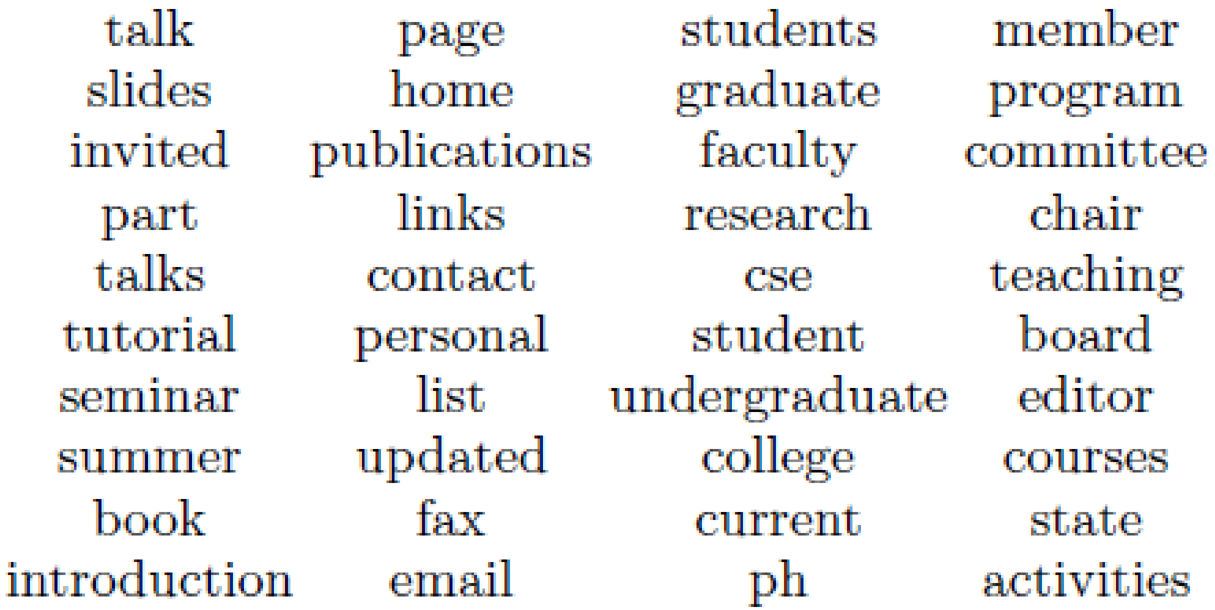
1. Clustering authors for disambiguation
2. Clustering document collections into subject areas
3. Citation recommendation
4. Predicting Influential Authors
5. Analyzing topic trends over time
6. Improving ranking tasks (Homepage retrieval, Expertise search)

What does a researcher homepage look like?

A CS researcher homepage is a mixture of types of information (topics)

Output from LDA on homepages from DBLP

Table 4. Top words of topics related to homepages



talk	page	students	member
slides	home	graduate	program
invited	publications	faculty	committee
part	links	research	chair
talks	contact	cse	teaching
tutorial	personal	student	board
seminar	list	undergraduate	editor
summer	updated	college	courses
book	fax	current	state
introduction	email	ph	activities

Researcher homepages often contain their research information

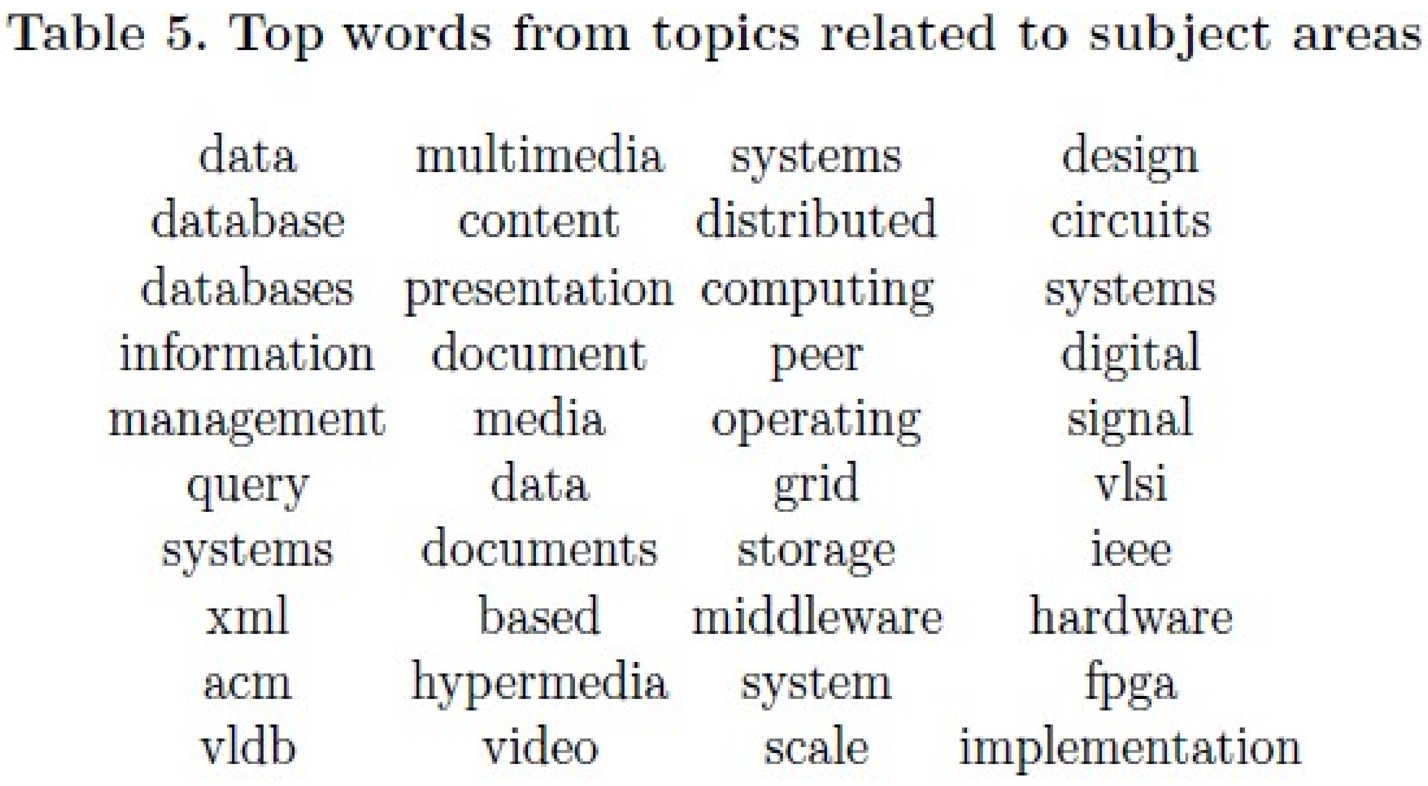


Table 5. Top words from topics related to subject areas

data	multimedia	systems	design
database	content	distributed	circuits
databases	presentation	computing	systems
information	document	peer	digital
management	media	operating	signal
query	data	grid	vlsi
systems	documents	storage	ieee
xml	based	middleware	hardware
acm	hypermedia	system	fpga
vldb	video	scale	implementation

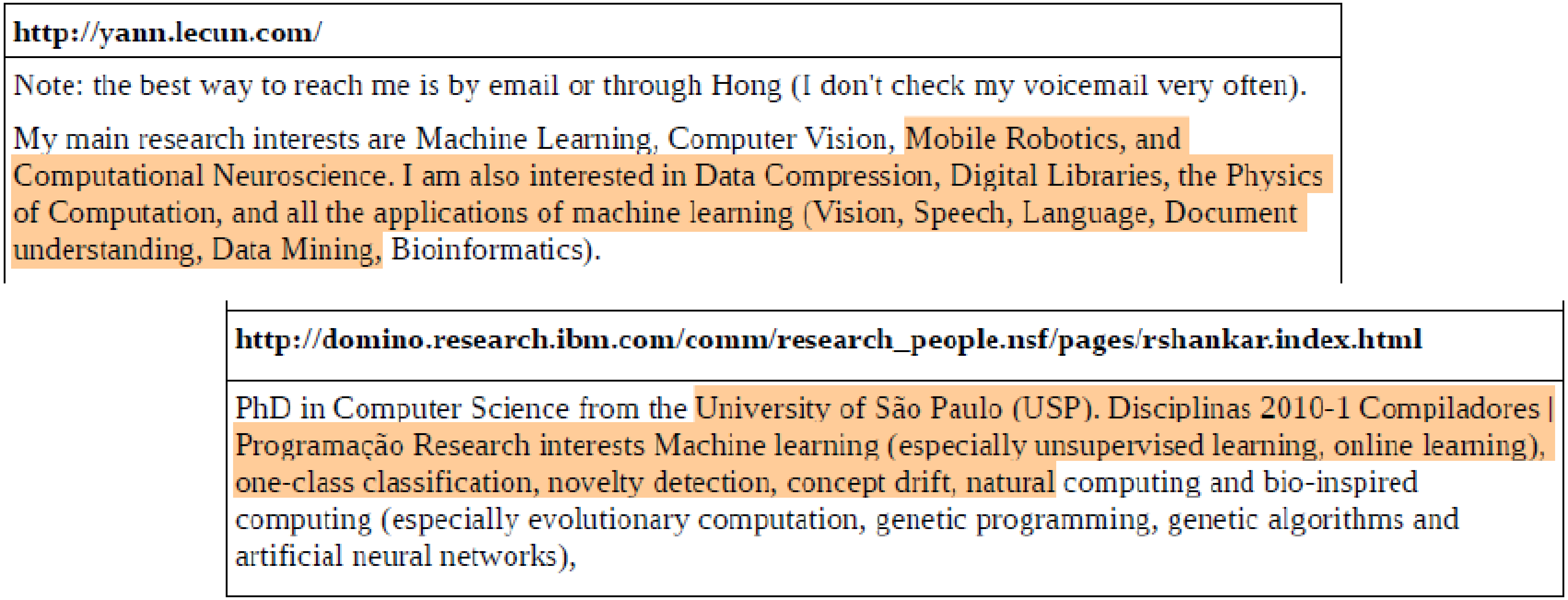
How can we use the output from LDA?

$$\mathtt{score}(s,t) = \sum_{w\in s} \phi_{w,t}$$

The research description segment is extracted using

$$p = \underset{t\in ST, s\in S}{argmax}\mathtt{score}(s,t)$$

Fig. 3. Sample research description segments extracted from homepages



http://yann.lecun.com/

Note: the best way to reach me is by email or through Hong (I don't check my voicemail very often).

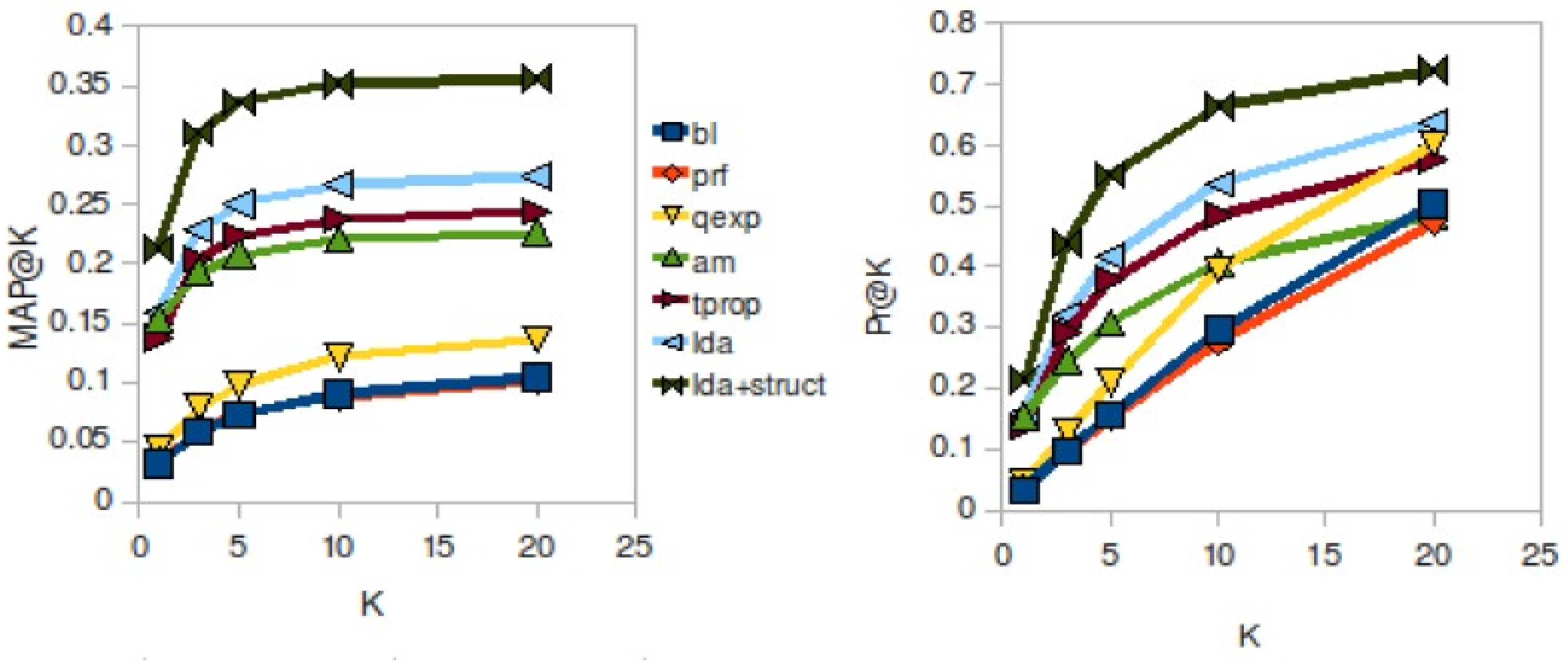
My main research interests are Machine Learning, Computer Vision, Mobile Robotics, and Computational Neuroscience. I am also interested in Data Compression, Digital Libraries, the Physics of Computation, and all the applications of machine learning (Vision, Speech, Language, Document understanding, Data Mining, Bioinformatics).

http://domino.research.ibm.com/comm/research_people.nsf/pages/rshankar.index.html

PhD in Computer Science from the University of São Paulo (USP). Disciplinas 2010-1 Compiladores | Programação Research interests Machine learning (especially unsupervised learning, online learning), one-class classification, novelty detection, concept drift, natural computing and bio-inspired computing (especially evolutionary computation, genetic programming, genetic algorithms and artificial neural networks), How can we use the output from LDA?

Improving homepage retrieval

- Train a re-ranking function on top of results from a search engine



References

1. Sujatha Das Gollapalli, C. Lee Giles, Prasenjit Mitra, Cornelia Caragea: On identifying academic homepages for digital libraries. JCDL 2011

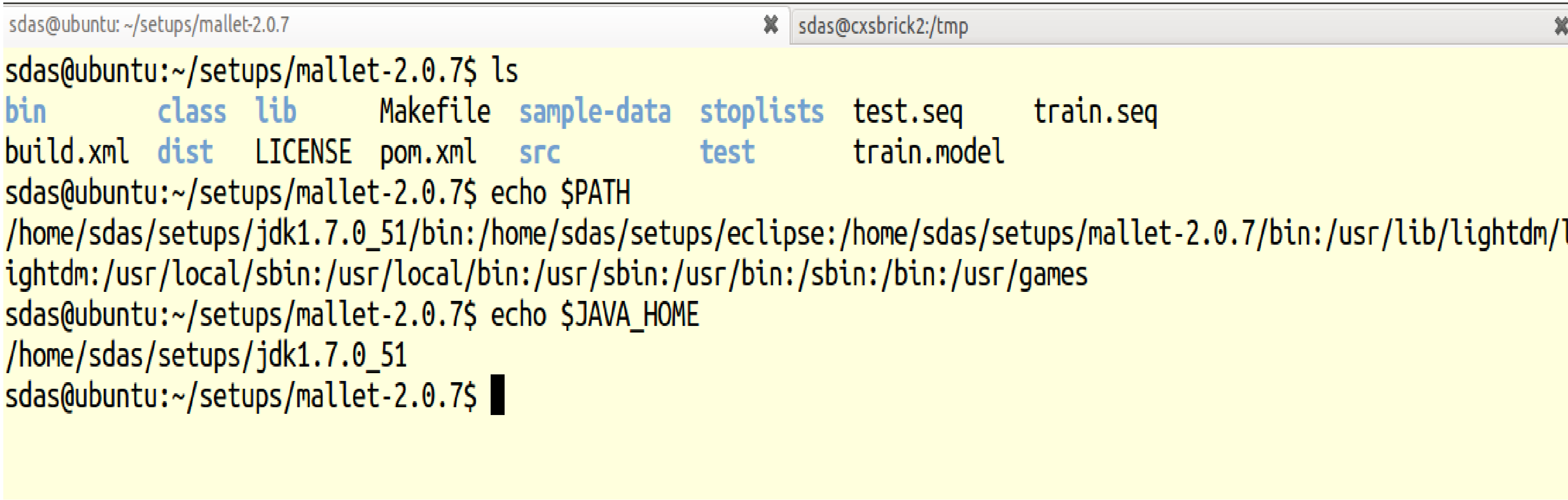
2. Sujatha Das Gollapalli, Prasenjit Mitra, C. Lee Giles: Ranking experts using author-document-topic graphs. JCDL 2013

3. Saurabh Kataria, Prasenjit Mitra, Cornelia Caragea, C. Lee Giles: Context Sensitive Topic Models for Author Influence in Document Networks. IJCAI 2011

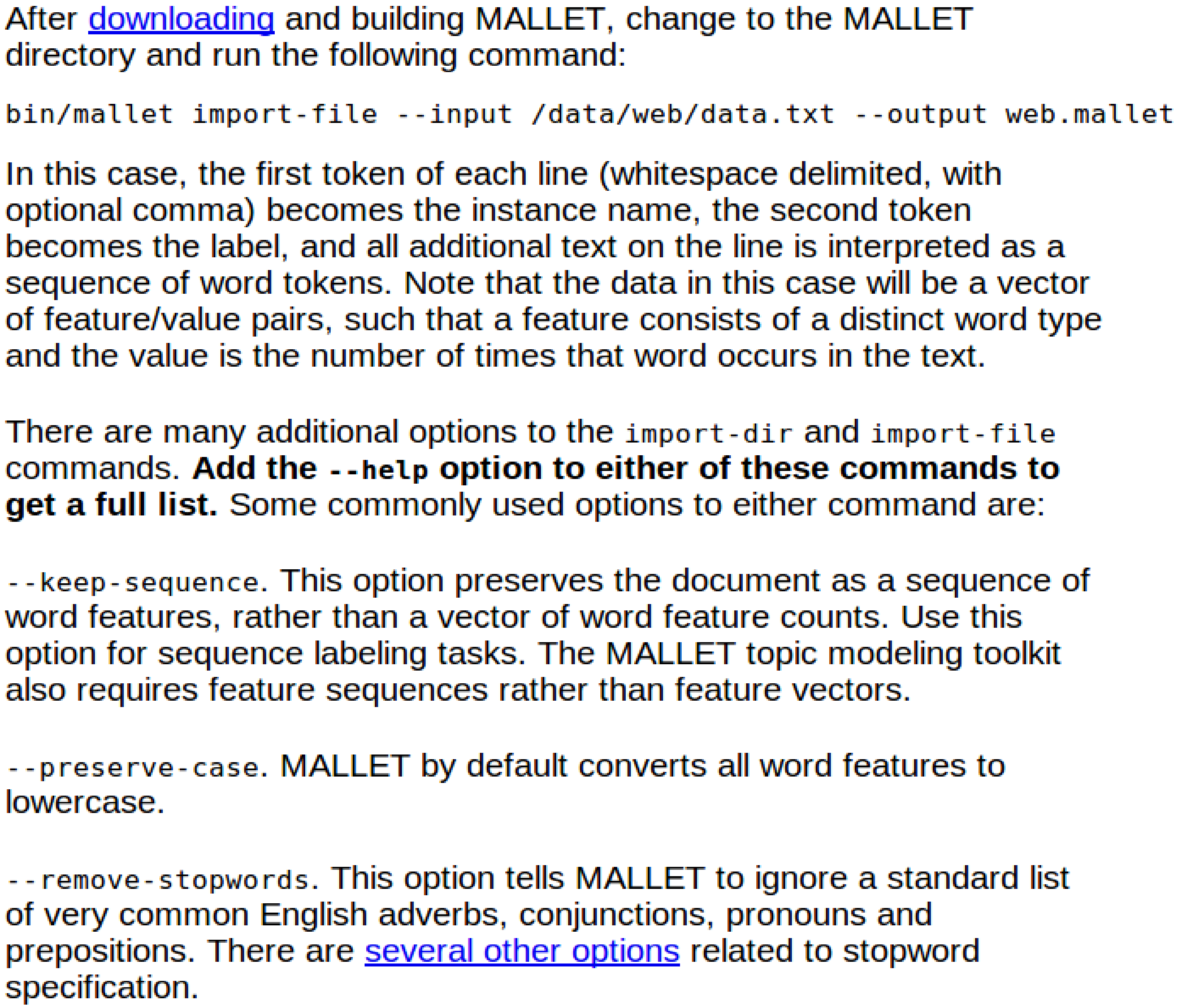
4. Qi He, Bi Chen, Jian Pei, Baojun Qiu, Prasenjit Mitra, C. Lee Giles: Detecting topic evolution in scientific literature: how can citations help? CIKM 2009

Thank you!

Download Mallet from
http://mallet.cs.umass.edu/download.php



```
sdas@ubuntu:~/setups/mallet-2.0.7$ ls
bin        class  lib      Makefile  sample-data  stoplists  test.seq     train.seq
build.xml  dist   LICENSE  pom.xml   src          test       train.model
sdas@ubuntu:~/setups/mallet-2.0.7$ echo $PATH
/home/sdas/setups/jdk1.7.0_51/bin:/home/sdas/setups/eclipse:/home/sdas/setups/mallet-2.0.7/bin:/usr/lib/lightdm/l
ightdm:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games
sdas@ubuntu:~/setups/mallet-2.0.7$ echo $JAVA_HOME
/home/sdas/setups/jdk1.7.0_51
sdas@ubuntu:~/setups/mallet-2.0.7$ █
```



After downloading and building MALLET, change to the MALLET directory and run the following command:

```
bin/mallet import-file --input /data/web/data.txt --output web.mallet
```

In this case, the first token of each line (whitespace delimited, with optional comma) becomes the instance name, the second token becomes the label, and all additional text on the line is interpreted as a sequence of word tokens. Note that the data in this case will be a vector of feature/value pairs, such that a feature consists of a distinct word type and the value is the number of times that word occurs in the text.

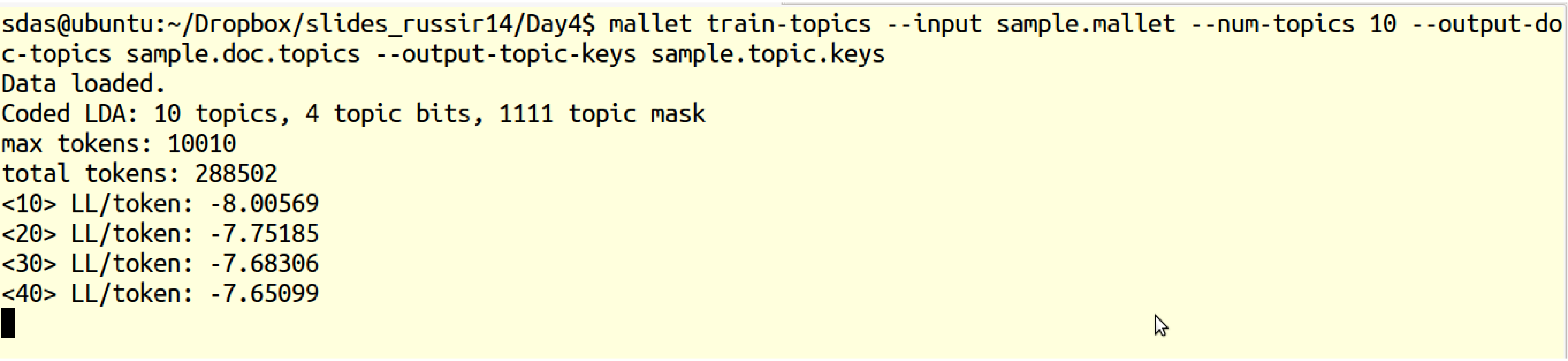
There are many additional options to the `import-dir` and `import-file` commands. **Add the `--help` option to either of these commands to get a full list.** Some commonly used options to either command are:

`--keep-sequence`. This option preserves the document as a sequence of word features, rather than a vector of word feature counts. Use this option for sequence labeling tasks. The MALLET topic modeling toolkit also requires feature sequences rather than feature vectors.

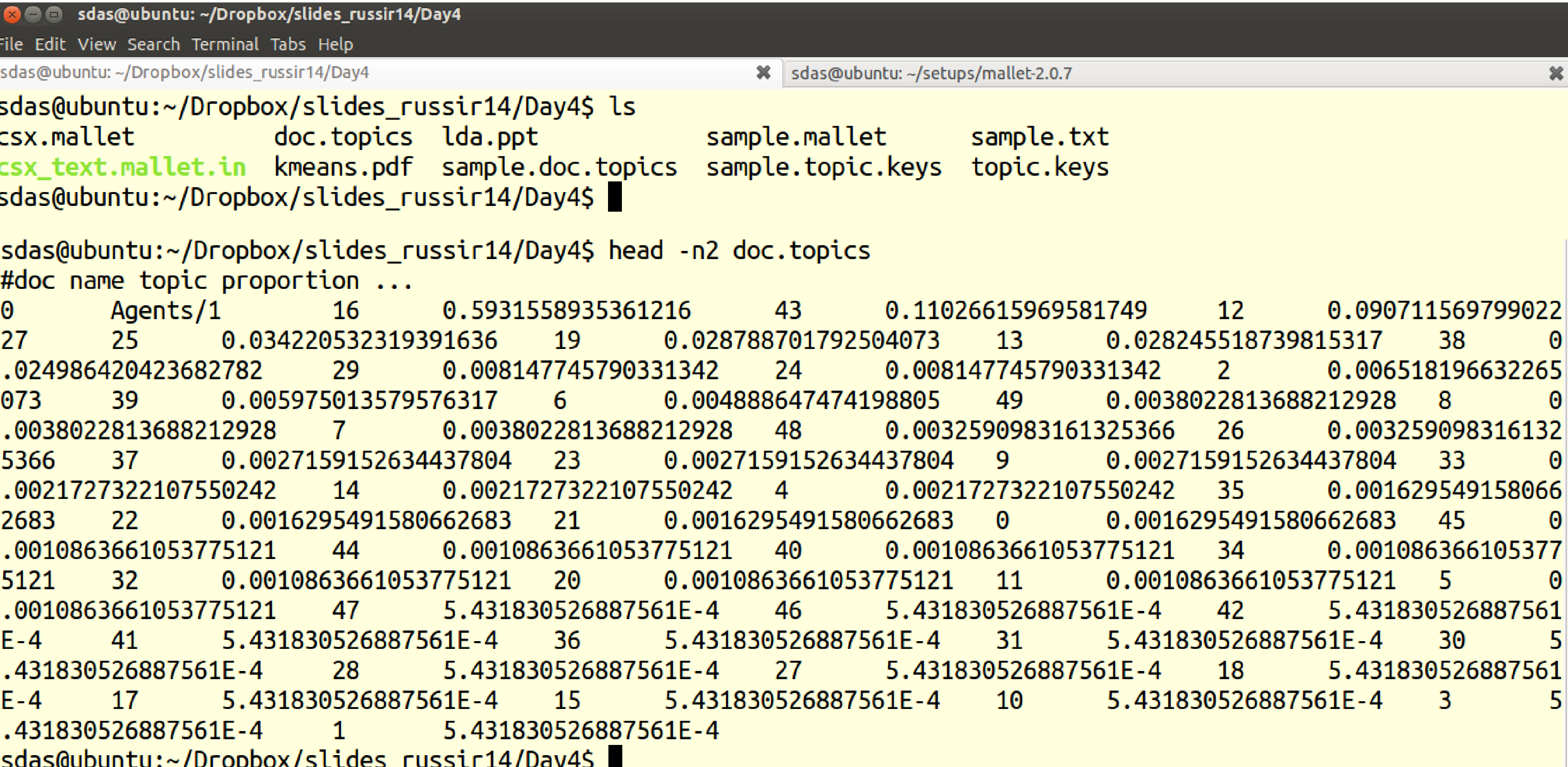
`--preserve-case`. MALLET by default converts all word features to lowercase.

`--remove-stopwords`. This option tells MALLET to ignore a standard list of very common English adverbs, conjunctions, pronouns and prepositions. There are several other options related to stopword specification.

```
mallet import-file --input sample.txt --output sample.mallet
--keep-sequence
```

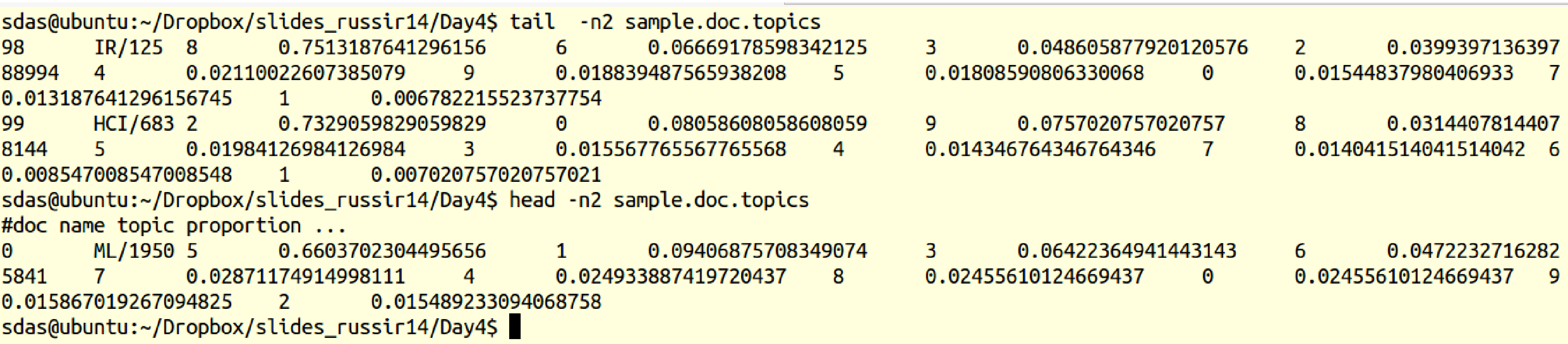


```
sdas@ubuntu:~/Dropbox/slides_russir14/Day4$ mallet train-topics --input sample.mallet --num-topics 10 --output-do
c-topics sample.doc.topics --output-topic-keys sample.topic.keys
Data loaded.
Coded LDA: 10 topics, 4 topic bits, 1111 topic mask
max tokens: 10010
total tokens: 288502
<10> LL/token: -8.00569
<20> LL/token: -7.75185
<30> LL/token: -7.68306
<40> LL/token: -7.65099
```

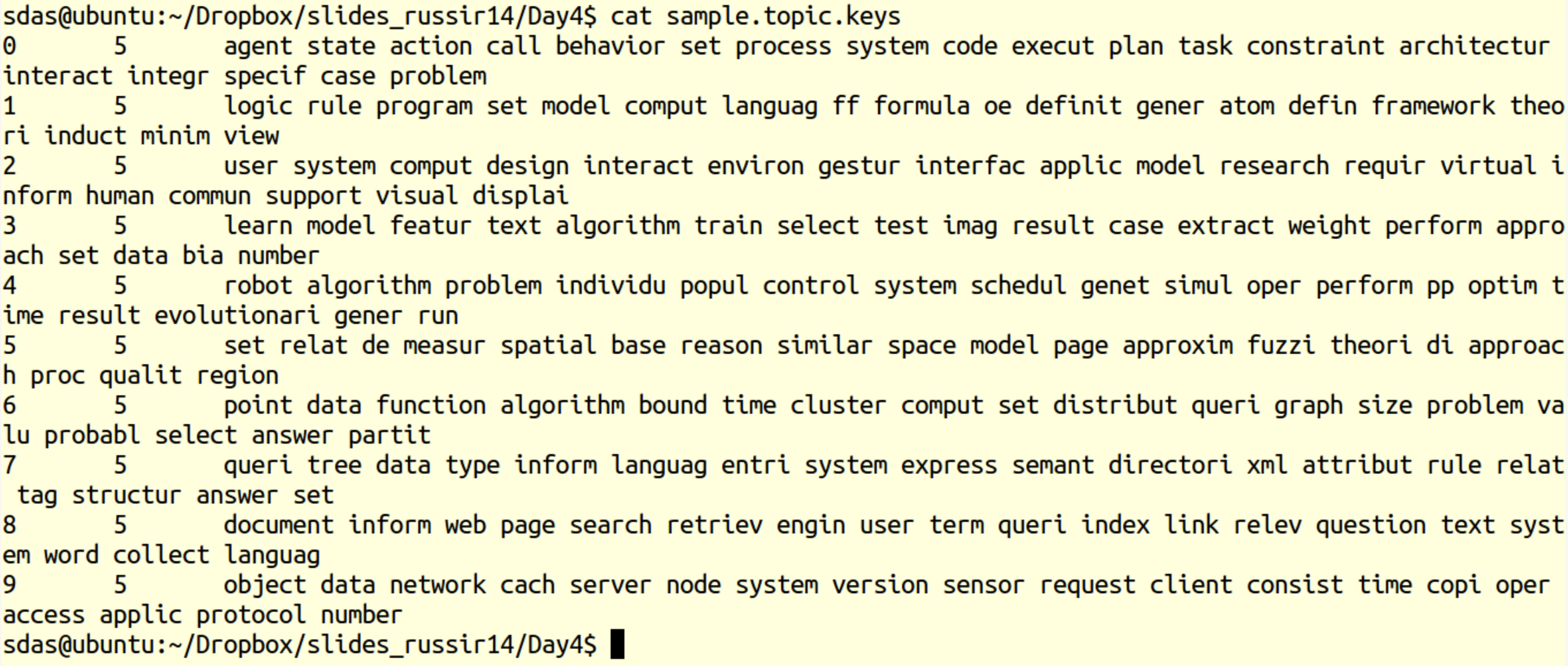

sdas@ubuntu: ~/Dropbox/slides_russir14/Day4
File Edit View Search Terminal Tabs Help
sdas@ubuntu: ~/Dropbox/slides_russir14/Day4
sdas@ubuntu: ~/setups/mallet-2.0.7

```
sdas@ubuntu:~/Dropbox/slides_russir14/Day4$ ls
csx.mallet          doc.topics  lda.ppt            sample.mallet      sample.txt
csx_text.mallet.in  kmeans.pdf  sample.doc.topics  sample.topic.keys  topic.keys
sdas@ubuntu:~/Dropbox/slides_russir14/Day4$
```

```
sdas@ubuntu:~/Dropbox/slides_russir14/Day4$ head -n2 doc.topics
#doc name topic proportion ...
0       Agents/1        16      0.5931558935361216      43      0.11026615969581749     12      0.090711569799022
27      25      0.034220532319391636    19      0.028787017925040073    13      0.028245518739815317    38      0
.024986420423682782     29      0.008147745790331342    24      0.008147745790331342    2       0.006518196632265
073     39      0.005975013579576317    6       0.004888647474198805    49      0.003802281368821292    8       0
.003802281368821292     7       0.003802281368821292    48      0.0032590983161325366   26      0.003259098316132
5366    37      0.0027159152634437804   23      0.0027159152634437804   9       0.0027159152634437804   33      0
.0021727322107550242    14      0.0021727322107550242   4       0.0021727322107550242   35      0.001629549158066
2683    22      0.0016295491580662683   21      0.0016295491580662683   0       0.0016295491580662683   45      0
.0010863661053775121    44      0.0010863661053775121   40      0.0010863661053775121   34      0.001086366105377
5121    32      0.0010863661053775121   20      0.0010863661053775121   11      0.0010863661053775121   5       0
.0010863661053775121    47      5.431830526887561E-4    46      5.431830526887561E-4    42      5.431830526887561
E-4     41      5.431830526887561E-4    36      5.431830526887561E-4    31      5.431830526887561E-4    30      5
.431830526887561E-4     28      5.431830526887561E-4    27      5.431830526887561E-4    18      5.431830526887561
E-4     17      5.431830526887561E-4    15      5.431830526887561E-4    10      5.431830526887561E-4    3       5
.431830526887561E-4     1       5.431830526887561E-4
sdas@ubuntu:~/Dropbox/slides_russir14/Day4$
```



```
sdas@ubuntu:~/Dropbox/slides_russir14/Day4$ tail  -n2 sample.doc.topics
98      IR/125  8       0.7513187641296156      6       0.06669178598342125     3       0.048605877920120576    2       0.0399397136397
88994   4       0.0211002260738507 9    9       0.018839487565938208    5       0.01808590806330068     0       0.015448379804 06933    7
0.013187641296156745    1       0.006782215523737754
99      HCI/683 2       0.7329059829059829      0       0.08058608058608059     9       0.0757020757020757      8       0.0314407814407
8144    5       0.0198412698412698 4    3       0.015567765567765568    4       0.014346764346764346    7       0.014041514041514042    6
0.008547008547008548    1       0.007020757020757021
sdas@ubuntu:~/Dropbox/slides_russir14/Day4$ head -n2 sample.doc.topics
#doc name topic proportion ...
0       ML/1950 5       0.6603702304495656      1       0.09406875708349074     3       0.06422364941443143     6       0.0472232716282
5841    7       0.02871174914998111     4       0.024933887419720437    8       0.02455610124669437     0       0.02455610124669437     9
0.015867019267094825    2       0.015489233094068758
sdas@ubuntu:~/Dropbox/slides_russir14/Day4$ 
```



```
sdas@ubuntu:~/Dropbox/slides_russir14/Day4$ cat sample.topic.keys
0       5       agent state action call behavior set process system code execut plan task constraint architectur
interact integr specif case problem
1       5       logic rule program set model comput languag ff formula oe definit gener atom defin framework theo
ri induct minim view
2       5       user system comput design interact environ gestur interfac applic model research requir virtual i
nform human commun support visual displai
3       5       learn model featur text algorithm train select test imag result case extract weight perform appro
ach set data bia number
4       5       robot algorithm problem individu popul control system schedul genet simul oper perform pp optim t
ime result evolutionari gener run
5       5       set relat de measur spatial base reason similar space model page approxim fuzzi theori di approac
h proc qualit region
6       5       point data function algorithm bound time cluster comput set distribut queri graph size problem va
lu probabl select answer partit
7       5       queri tree data type inform languag entri system express semant directori xml attribut rule relat
 tag structur answer set
8       5       document inform web page search retriev engin user term queri index link relev question text syst
em word collect languag
9       5       object data network cach server node system version sensor request client consist time copi oper
access applic protocol number
sdas@ubuntu:~/Dropbox/slides_russir14/Day4$ 
```