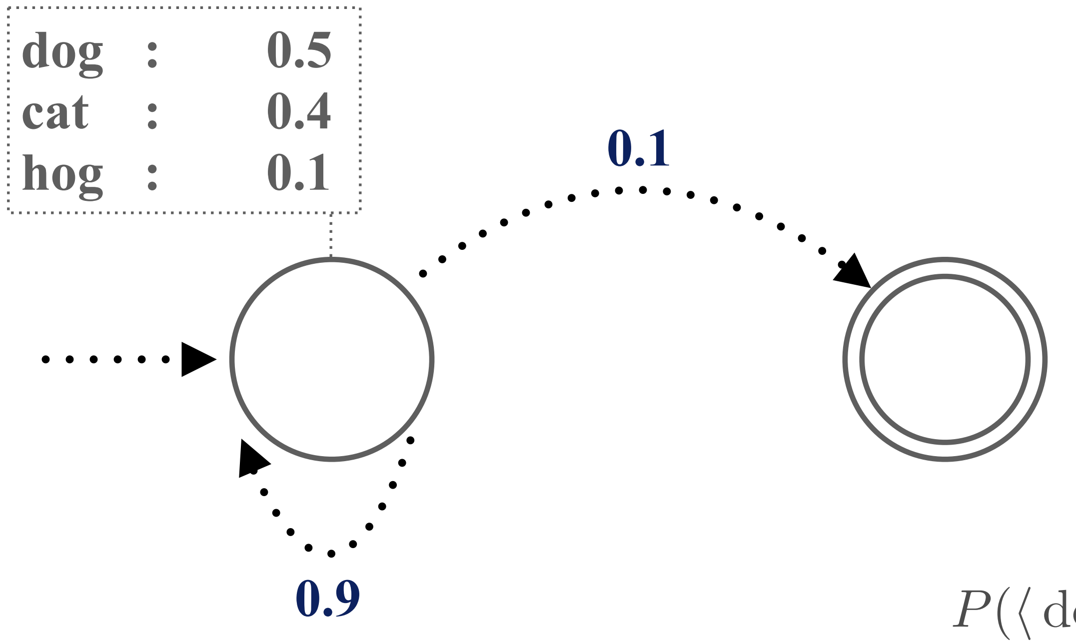
III.4 Statistical Language Models

1. **Basics of Statistical Language Models**
2. **Query-Likelihood Approaches**
3. **Smoothing Methods**
4. **Divergence Approaches**
5. **Extensions**

Based on **MRS Chapter 12** and [Zhai 2008]

1. Basics of Statistical Language Models

- Statistical language models (LMs) are **generative models of word sequences** (or, bags of words, sets of words, etc.)



$$P(\langle \text{hog} \rangle) = 0.1 \times 0.1$$

$$P(\langle \text{cat, dog} \rangle) = 0.4 \times 0.9 \times 0.5 \times 0.1$$

$$P(\langle \text{dog, dog, hog} \rangle) = 0.5 \times 0.9 \times 0.5 \times 0.9 \times 0.1 \times 0.1$$

- Application examples:
    - **Speech recognition**, e.g., to select among multiple phonetically similar sentences (“*get up at 8 o’clock*” vs. “*get a potato clock*”)
    - **Statistical machine translation**, e.g., to select among multiple candidate translations (“*logical closing*” vs. “*logical reasoning*”)
    - **Information retrieval**, e.g., to rank documents in response to a query

Types of Language Models

- **Unigram LM** based on only **single words** (unigrams), considers **no context**, and assumes **independent generation** of words

$$P(\langle t_1, \ldots, t_m \rangle) = \prod_{i=1}^{m} P(t_i)$$

- **Bigram LM** conditions on the preceding term

$$P(\langle t_1, \ldots, t_m \rangle) = P(t_1) \prod_{i=2}^{m} P(t_i | t_{i-1})$$

- ***n*-Gram LM** conditions on the preceding (*n*-1) terms

$$P(\langle t_1, \ldots, t_m \rangle) = P(t_1)\, P(t_2 | t_1) \ldots \prod_{i=n}^{m} P(t_i | t_{i-n+1} \ldots t_{i-1})$$

Parameter Estimation

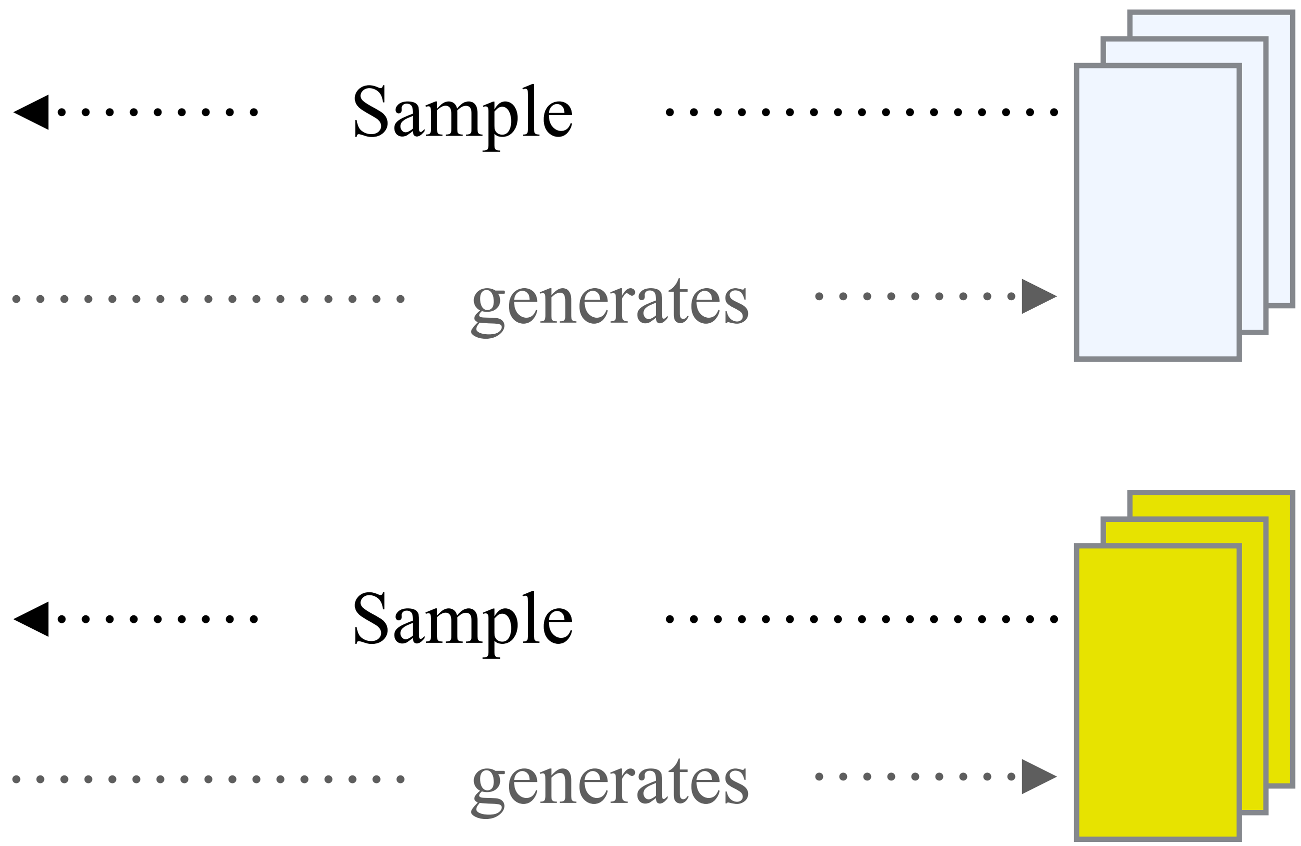
- **Parameters** (e.g., $P(t_i)$, $P(t_i \mid t_{i-1})$) of **language model** θ are estimated based on a **sample of documents**, which are assumed to have been **generated by** θ
- Example: Unigram language models θ_{Sports} and $\theta_{Politics}$ estimated from documents about sports and politics

θ_{Sports}

soccer	:	0.20
goal	:	0.15
tennis	:	0.10
player	:	0.05
	:	

$\theta_{Politics}$

party	:	0.20
debate	:	0.20
scandal	:	0.15
election	:	0.05
	:	



Probabilistic IR vs. Statistical Language Models

$$P[R|d,q]$$

"User finds document d relevant to query q"

$$\propto \frac{P[R|d,q]}{P[\bar{R}|d,q]}$$

$$\propto \frac{P[q,d|R]}{P[q,d|\bar{R}]}$$

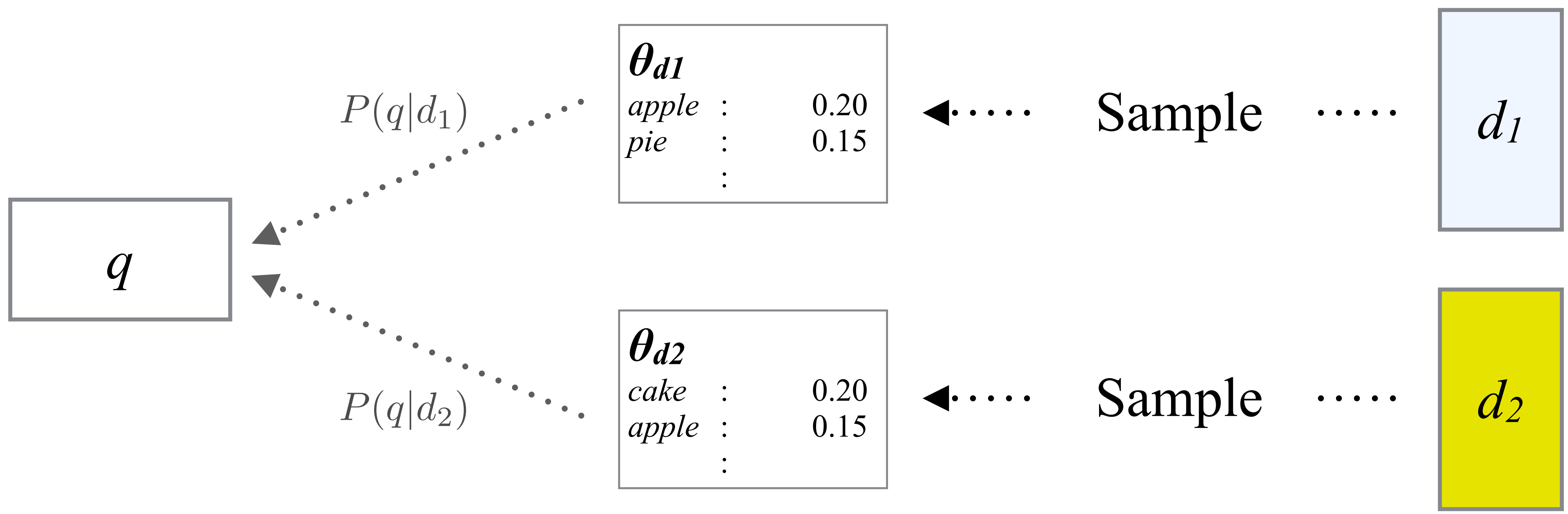
$$= \frac{P[q|d,R]}{P[q|d,\bar{R}]} \frac{P[R|d]}{P[\bar{R}|d]}$$

$$\propto P[q|d,R]$$

<u>Probabilistic IR</u>
ranks according to
relevance odds

<u>Statistical LMs</u>
rank according to
query likelihood

2. Query-Likelihood Approaches



- $P(q|d)$ is the **likelihood that the query was generated** by the language model θ_d estimated from document d
- Intuition:
  - User formulates query q by selecting words from a **prototype document**
  - Which document is "closest" to that prototype document

Multi-Bernoulli LM

- Query q is seen as a **set of terms** and generated from document d by **tossing a coin** for every word from the vocabulary V

$$P(q|d) = \prod_{t \in q} P(t|d) \times \prod_{t \in V \setminus q} (1 - P(t|d))$$

$$\approx \prod_{t \in q} P(t|d) \quad (\textbf{assuming } |q| << |V|)$$

- [Ponte and Croft '98] **pioneered** the use of LMs in IR

Multinomial LM

- Query q is seen as a **bag of terms** and generated from document d by **drawing terms** from the bag of terms corresponding to d

$$
\begin{aligned}
P(q|d) &= \binom{|q|}{tf(t_1, q) \ldots tf(t_{|q|}, q)} \prod_{t_i \in q} P(t_i|d)^{\,tf(t_i, q)} \\
&\propto \prod_{t_i \in q} P(t_i|d)^{\,tf(t_i, q)} \\
&\approx \prod_{t_i \in q} P(t_i|d) \quad (\textbf{assuming } \forall t_i \in q \, : \, tf(t_i, q) = 1)
\end{aligned}
$$

- **Multinomial LM** is more expressive than **Multi-Bernoulli LM** and therefore **usually preferred**

Multinomial LM (cont'd)

- **Maximum-likelihood estimate** for parameters $P(t_i|d)$

$$P(t_i|d) = \frac{tf(t_i, d)}{|d|}$$

is prone to **overfitting** and leads to

- bias in favor of **short documents** / against **long documents**
- **conjunctive query semantics**, i.e., query can not be generated from language models of documents that miss one of the query terms

3. Smoothing

- Smoothing methods avoid **overfitting** to the sample (often: one document) and are **essential** for LMs to work in practice
  - Laplace smoothing (cf. Chapter III.3)
  - Absolute discounting
  - **Jelinek-Mercer smoothing**
  - **Dirichlet smoothing**
  - Good-Turing smoothing
  - Katz's back-off model
  - …
- **Choice** of smoothing method and **parameter setting** still mostly **"black art"** (or empirical, i.e., based on training data)

Jelinek-Mercer Smoothing

- Uses a **linear combination** (mixture) of document language model θ_d and **document-collection language model** θ_D

$$P(t|d) = \lambda \frac{tf(t,d)}{|d|} + (1-\lambda) \frac{tf(t,D)}{|D|}$$

  with document *D* as concatenation of entire document collection

- Parameter λ can be tuned by **cross-validation** with held-out data
  - divide set of relevant (*q*, *d*) pairs into *n* partitions
  - build LM on the pairs from *n*-1 partitions
  - choose λ to maximize precision (or recall or F1) on held-out partition
  - iterate with different choice of n^{th} partition and average
- Parameter λ can be made **document- or term-dependent**

Jelinek-Mercer Smoothing vs. TF*IDF

$$\begin{aligned} P(q|d) &= \prod_{t \in q} P(t|d) \\ &= \prod_{t \in q} \left(\lambda \frac{tf(t,d)}{|d|} + (1-\lambda) \frac{tf(t,D)}{|D|} \right) \\ &\propto \sum_{t \in q} \log \left(\lambda \frac{tf(t,d)}{|d|} + (1-\lambda) \frac{tf(t,D)}{|D|} \right) \\ &\propto \sum_{t \in q} \log \left(1 + \frac{\lambda}{1-\lambda} \underbrace{\frac{tf(t,d)}{|d|}}_{\sim \text{tf}} \underbrace{\frac{|D|}{tf(t,D)}}_{\sim \text{idf}} \right) \end{aligned}$$

- (Jelinek-Mercer) smoothing has **effect similar to IDF weighting**
- Jelinek-Mercer smoothing leads to a **TF*IDF-style model**

Dirichlet-Prior Smoothing

- Uses **Bayesian estimation** with a conjugate Dirichlet prior instead of the Maximum-Likelihood Estimation

$$P(t|d) = \frac{tf(t,d) + \alpha \frac{tf(t,D)}{|D|}}{|d| + \alpha}$$

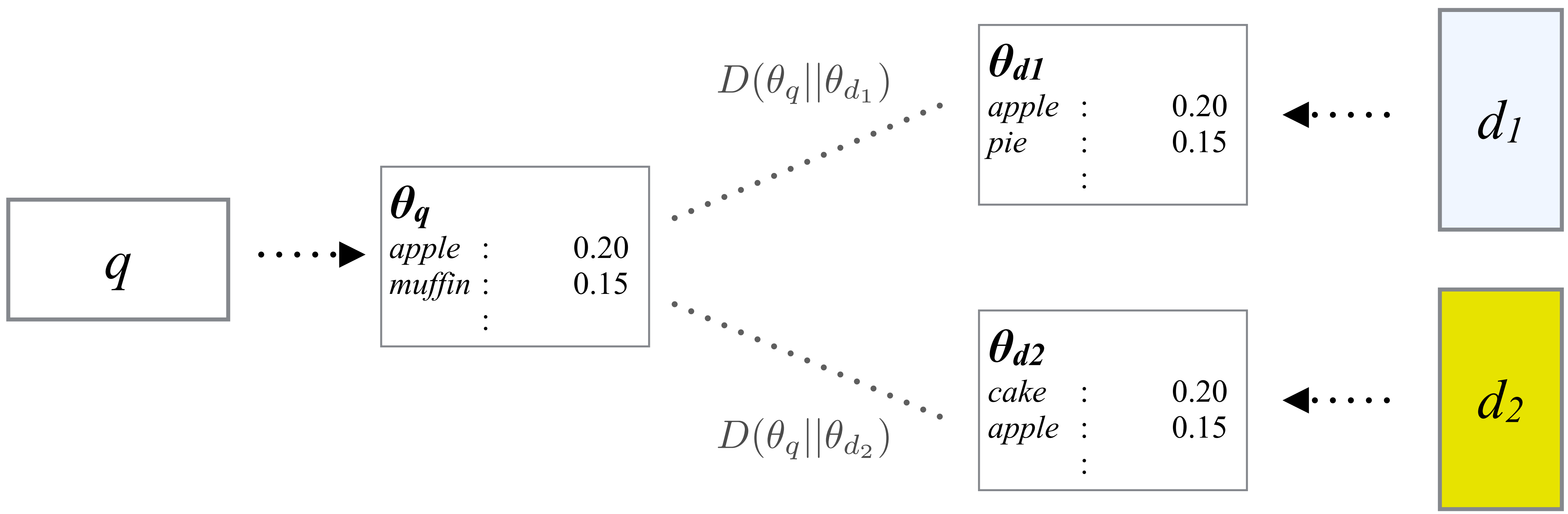
- <u>Intuition</u>: Document d is **extended by $\boldsymbol{\alpha}$ terms** generated by the document-collection language model
- Parameter $\boldsymbol{\alpha}$ usually set as multiple of **average document length**

Dirichlet Smoothing vs. Jelinek-Mercer Smoothing

$$
\begin{aligned}
P(t|d) &= \lambda \frac{tf(t,d)}{|d|} + (1-\lambda)\frac{tf(t,D)}{|D|} \\
&= \frac{|d|}{|d|+\alpha}\frac{tf(t,d)}{|d|} + \frac{\alpha}{|d|+\alpha}\frac{tf(t,D)}{|D|} \quad (\textbf{set } \lambda = \frac{|d|}{|d|+\alpha}) \\
&= \frac{tf(t,d)+\alpha\frac{tf(t,D)}{|D|}}{|d|+\alpha}
\end{aligned}
$$

- Jelinek-Mercer smoothing with **document-dependent λ** becomes a **special case** of Dirichlet smoothing

4. Divergence Approaches



- Query-likelihood approaches see **query as a sample from a LM**
- Query expansion, relevance feedback, etc. are **difficult to express as query-likelihood approaches**, since they would require tinkering with the sample (i.e., the query) and more fine-grained control than adding/removing terms

Kullback-Leibler Divergence

- Kullback-Leibler divergence (aka. information gain or relative entropy) is an **information-theoretic non-symmetric** measure of distance between **probability distributions**

$$D(\theta_q||\theta_d) \quad = \quad \sum_{t \in V} P(t|\theta_q) \, \log \, \frac{P(t|\theta_q)}{P(t|\theta_d)}$$

- <u>Example</u>:

$\boldsymbol{\theta_q}$	
apple :	0.50
muffin :	0.50

$\boldsymbol{\theta_d}$	
apple :	0.25
muffin :	0.25
recipe :	0.10
water :	0.10
sugar :	0.30

$$\begin{aligned} D(\theta_q \| \theta_d) &= P(\text{apple}|\theta_q) \, \log \, \frac{P(\text{apple}|\theta_q)}{P(\text{apple}|\theta_d)} + P(\text{muffin}|\theta_q) \, \log \, \frac{P(\text{muffin}|\theta_q)}{P(\text{muffin}|\theta_d)} \\ &= 0.50 \, \log \, \frac{0.50}{0.25} + 0.50 \, \log \, \frac{0.50}{0.25} \\ &= 1.00 \end{aligned}$$

Relevance Feedback LM

- [Zhai and Lafferty '01] re-estimate query language model as

$$P(t|\theta'_q) = (1-\alpha)\, P(t|\theta_q) + \alpha\, P(t|\theta_F)$$

  with F as the set of **documents with positive feedback** from user

- MLE of θ_F obtained by **maximizing log-likelihood function**

$$\log\ P(F|\theta_F) = \sum_{t \in V} tf(t,F)\ \log\left((1-\lambda)\, P(t|\theta_F) + \lambda\, P(t|\theta_D)\right)$$

  with *tf*(t, F) as the **total term frequency** of t in documents from F and θ_D as the **document-collection language model**

5. Extensions

- Statistical language models have been **one of the highly active areas in IR research** during the past decade and continue to be
- Extensions:
  - Term-specific and document-specific smoothing (JM-style smoothing with term-specific λt or document-specific λd)
  - **(Semantic) Translation LMs** (e.g., to consider synonyms or support cross-lingual IR)
  - **Time-based LMs** (e.g., with time-dependent document prior to favor recent documents)
  - **LMs for (semi-)structured XML and RDF data** (e.g., for entity search or question answering)
  - …

Translation LM for Cross-Lingual IR

- Cross-Lingual IR:
  - Users issue **queries in their native language** (e.g., German) (e.g., *spionage usa bundesregierung*)
  - System returns **documents in another known language** (e.g., English) (e.g., *reactions of the German government to U.S. eavesdropping on …*)

$$P(q|d) = \prod_{t \in q} \sum_{w} P(t|w)\, P(w|d)$$

- **Translation probabilities** $P(t|w)$ obtained from a **dictionary** or estimated based on a **parallel cross-lingual corpus**
- [Federico and Bertoldi ’01] as **more advanced approach** based on a Hidden-Markov Model that also considers **term contexts**

Time-Based LMs

- Intuition: For **news-related queries** (e.g., *german election*) documents **published more recently** are often preferable

- [Li and Croft '03] rank documents according to

$$P(q|d)\,P(d^{\,t}) = \left(\prod_{t \in q} P(t|d^{\,t}) \right) \left(\lambda\, e^{-\lambda\,(\mathrm{now}-t)} \right)$$

  with **document publication timestamp** *t* and **time-dependent exponentially decaying document prior** P(d^t)

- [Peetz and de Rijke '13] consider other document priors motivated by cognitive psychology research on human memory

LM for Entity Search

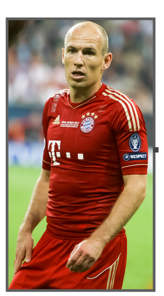
- Objective: Retrieve **entities** (e.g., people, locations, organizations) relevant to query q as opposed to only documents [Ni et al. '07]

Query q:
dutch soccer player munich

→

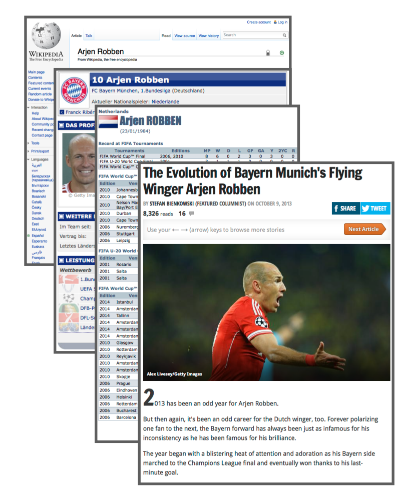
Candidate Entities:
1. Arjen Robben
2. Rafael van der Vaart
3. Louis van Gaal
4. Daniel van Buyten
5. Toni Kroos

- Language model θ_e for entity e can be **estimated from contexts** in which the **entity is mentioned** in the document collection, possibly taking into account extraction accuracy



rayand@flickr

…munich's flying dutchman…
…one of bayern's most valuable players…
…winning soccer's most prestigious champions league…
…with the dutch national team…



Summary of III.4

- **Statistical language models**
  widely used in natural language applications other than IR
- **Query-likelihood approaches**
  see the query as a sample from the document LM
- **Divergence approaches**
  are more expressive comparing query LM against document LM
- **Smoothing methods**
  are absolutely essential to make LMs work in practice
- **Various extensions**
  for advanced tasks such as cross-lingual IR or entity search

Additional Literature for III.4

- **D. Hiemstra**: *Using Language Models for Information Retrieval*, Ph.D. Thesis, University of Twente, 2001
- **M. Federico and N. Bertoldi**: *Statistical Cross-Language Information Retrieval using N-Best Query Translations*, SIGIR 2001
- **Z. Nie, Y. Ma, S. Shi, J.-R. Wen and W.-Y. Ma**: *Web Object Retrieval,* WWW 2007
- **H. M. Peetz and M. de Rijke**: *Cognitive Temporal Document Priors*, ECIR 2013
- **J. M. Ponte and B. Croft**: *A Language Modeling Approach to Information Retrieval*, SIGIR 1998
- **C. Zhai and J. Lafferty**: *Model-based Feedback in the Language Modeling Approach for Information Retrieval*, CIKM 2001
- **C. Zhai**: *Statistical Language Models for Information Retrieval A Critical Review*, Foundations and Trends in Information Retrieval 2(3):137-213, 2008

III.5 Latent Topic Models

1. **Latent Semantic Indexing**
2. **Probabilistic Latent Semantic Indexing**
3. **Latent Dirichlet Allocation**

Based on **MRS Chapter 18** and [Blei ‘12]

Latent Topic Models

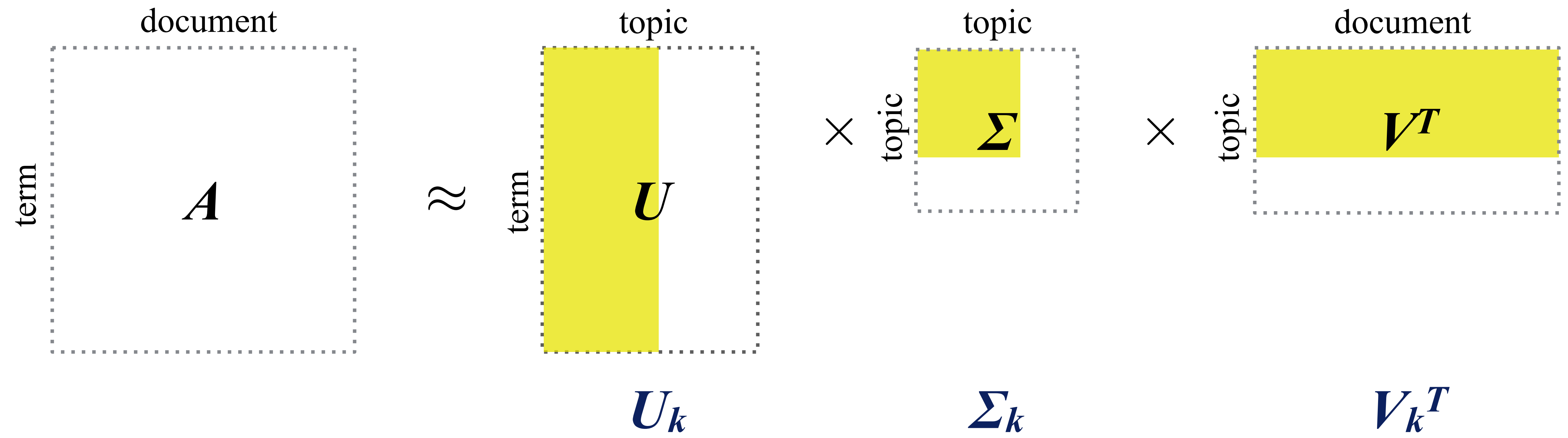
- Retrieval models seen so far (e.g., TF*IDF, LMs) do not handle **synonymy** (e.g., *car* and *automobile*), **polysemy** (e.g., *java*), etc.

- **Word co-occurrence** can help us, e.g.:
  - *car* and *automobile* both occur together with *garage*, *exhaust*, *fuel,…*
  - *java* occurs together with *class* and *method* but also with *grind* and *coffee*

- **Latent topic models** assume that documents are composed from a small number k of **latent (i.e., hidden, unknown) topics**
  - Latent Semantic Indexing (LSI) [Deerwester et al. '90]
  - Probabilistic Latent Semantic Indexing (pLSI) [Hofmann '99]
  - Latent Dirichlet Allocation (LDA) [Blei et al. '03]

1. Latent Semantic Indexing (LSI)

- Idea: Apply SVD to *m*-by-*n* term-document matrix A



- U_k, V_k^T, Σ_k contain the first k singular vectors and values
- U_k maps terms to topics
- V_k maps documents to topics

Operations in Latent Topic Space

- We can **map a query *q*** from *m*-dimensional term space into the *k*-dimensional topic space by $\boldsymbol{q} \rightarrow \boldsymbol{U_k^T}\, \boldsymbol{q} = \boldsymbol{q'}$
- **Ranking of documents** can then be determined by comparing $\boldsymbol{q'}$ against the columns of $\boldsymbol{V_k^T}$ using dot product or cosine similarity
- We can **fold in a new document** from *m*-dimensional term space by mapping it to *k*-dimensional topic space as $\boldsymbol{d} \rightarrow \boldsymbol{U_k^T}\, \boldsymbol{d} = \boldsymbol{d'}$ and appending it as a new column to $\boldsymbol{V_k^T}$ (with quality deteriorating over time)

LSI (Example)

m = 6 (terms)

t_1	:	*bak(e,ing)*
t_2	:	*recipe(s)*
t_3	:	*bread*
t_4	:	*cake*
t_5	:	*pastr(y,ies)*
t_6	:	*pie*

n = 5 (documents)

d_1	:	*how to bake bread without recipes*
d_2	:	*the classic art of viennese pastry*
d_3	:	*numerical recipes: the art of scientific computing*
d_4	:	*breads, pastries, pies and cakes: quantity baking recipes*
d_5	:	*pastry: a book of best french recipes*

$$A = \begin{pmatrix} 0.5774 & 0.0000 & 0.0000 & 0.4082 & 0.0000 \\ 0.5774 & 0.0000 & 1.0000 & 0.4082 & 0.7071 \\ 0.5774 & 0.0000 & 0.0000 & 0.4082 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & 0.4082 & 0.0000 \\ 0.0000 & 1.0000 & 0.0000 & 0.4082 & 0.7071 \\ 0.0000 & 0.0000 & 0.0000 & 0.4082 & 0.0000 \end{pmatrix}$$

LSI (Example)

$$A = \begin{pmatrix} 0.2670 & -0.2567 & 0.5308 & -0.2847 \\ 0.7479 & -0.3981 & -0.5249 & 0.0816 \\ 0.2670 & -0.2567 & 0.5308 & -0.2847 \\ 0.1182 & -0.0127 & 0.2774 & 0.6394 \\ 0.5198 & 0.8423 & 0.0838 & -0.1158 \\ 0.1182 & -0.0127 & 0.2774 & 0.6394 \end{pmatrix} \quad \boldsymbol{U}$$

$$\times \begin{pmatrix} 1.6950 & 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 1.1158 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.8403 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & 0.4195 \end{pmatrix} \quad \boldsymbol{\Sigma}$$

$$\times \begin{pmatrix} 0.4366 & 0.3067 & 0.4412 & 0.4909 & 0.5288 \\ -0.4717 & 0.7549 & -0.3568 & -0.0346 & 0.2815 \\ 0.3688 & 0.0998 & -0.6247 & 0.5711 & -0.3712 \\ -0.6715 & -0.2760 & 0.1945 & 0.6571 & -0.0577 \end{pmatrix} \quad \boldsymbol{V}^T$$

LSI (Example)

$$A_3 = \begin{pmatrix} 0.4971 & -0.0330 & 0.0232 & 0.4867 & -0.0069 \\ 0.6003 & 0.0094 & 0.9933 & 0.3858 & 0.7091 \\ 0.4971 & -0.0330 & 0.0232 & 0.4867 & -0.0069 \\ 0.1801 & 0.0740 & -0.0522 & 0.2320 & 0.0155 \\ -0.0326 & 0.9866 & 0.0094 & 0.4402 & 0.7043 \\ 0.1801 & 0.0740 & -0.0522 & 0.2320 & 0.0155 \end{pmatrix} = \boldsymbol{U}_3 \boldsymbol{\Sigma}_3 \boldsymbol{V}_3^T$$

LSI (Example)

- Query: *baking bread*
  - $\boldsymbol{q} = (1\ 0\ 1\ 0\ 0\ 0)^T$
  - $\boldsymbol{q'} = \boldsymbol{U_3^T q} = (0.5340\ {-0.5134}\ 1.0616)^T$
- Dot-product similarity in topic space
  - $\text{sim}(\boldsymbol{q}, \boldsymbol{d_1}) \approx 0.86$ / $\text{sim}(\boldsymbol{q}, \boldsymbol{d_2}) \approx -0.12$ / $\text{sim}(\boldsymbol{q}, \boldsymbol{d_3}) \approx -0.24$
- Adding $\boldsymbol{d_6}$ = "*algorithmic recipes for the computation of pie*"
  - $\boldsymbol{d} = (0\ 0.07071\ 0\ 0\ 0\ 0.07071)^T$
  - $\boldsymbol{d'} = \boldsymbol{U_3^T d} = (0.5\ {-0.28}\ {-0.15})^T$
  - $\boldsymbol{d'}$ becomes a new column of $\boldsymbol{V_k^T}$

Issues with LSI

- **Parameter tuning**
  - How to select proper number of latent topics k?
- **Memory consumption**
  - Term-by-document matrix $\boldsymbol{A}$ is usually sparse
  - SVD factors $\boldsymbol{U}$ and $\boldsymbol{V}$ are almost never sparse
- **Computational cost**
  - SVD still expensive to compute when m and n at the order of millions
- **Retrieval effectiveness**
  - LSI achieved only mediocre performance on TREC datasets with good gains for some queries but losses for others

2. Probabilistic Latent Semantic Indexing (pLSI)

- Idea: Model documents as (probabilistic) mixtures of topics
- Each topic generates terms with topic-specific probabilities
- Assume **conditional independence** of word *w* and document *d* given topic *t*:

$$P[w, d, t] = P[w, d|t]\, P[t] = P[w|t]\, P[d|t]\, P[t]$$

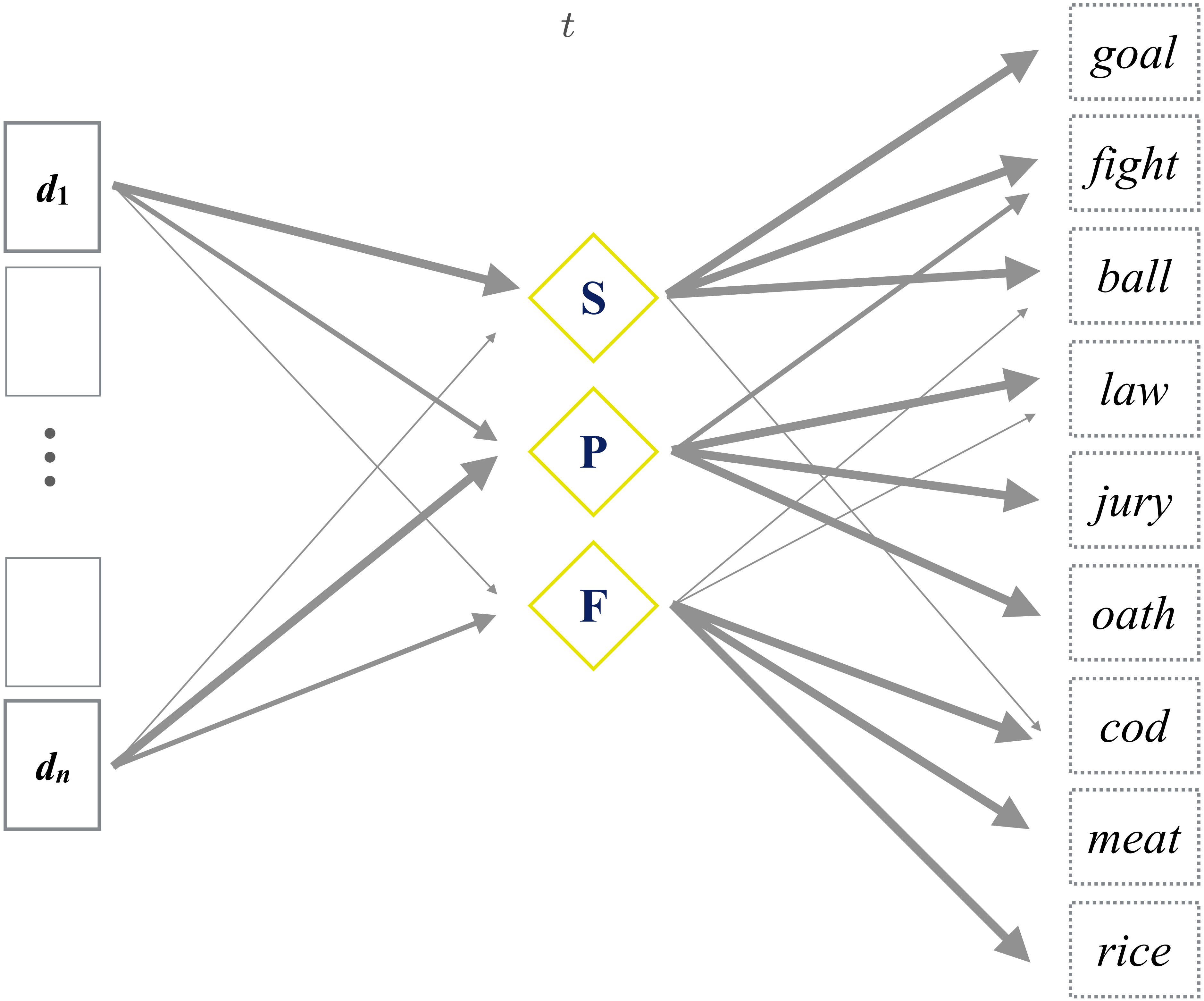
$$P[w, d] = \sum_t P[w|t] P[d|t] P[t]$$

- **Generative model**

$$P[w|d] = \sum_t P[w|t]\, P[t|d]$$

pLSI Generative Model

$$P[w|d] = \sum_t P[w|t]\, P[t|d]$$

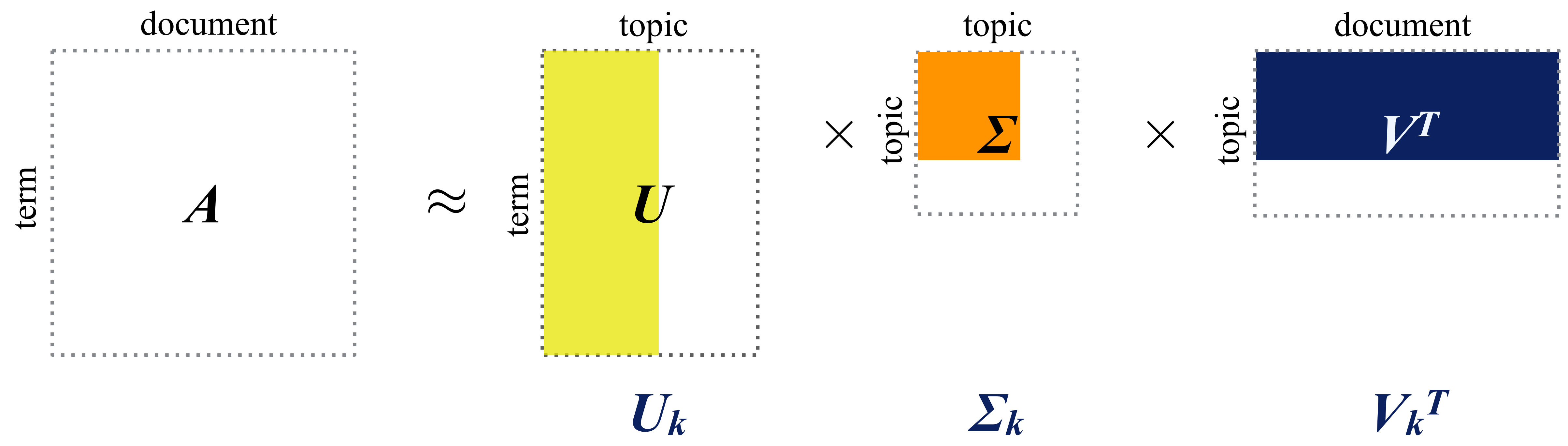


Computing pLSI

- **Parameters** $P[t|d]$ and $P[w|t]$ can be determined using the iterative method **Expectation Maximization** (EM)
- **Query** q is folded in by estimating the topic distribution $P[t|q]$ that provides the best explanation of the query terms
- **Ranking of documents** can then be determined by comparing the topic distributions $P[t|q]$ and $P[t|d]$, e.g., using KL divergence

pLSI vs. LSI

$$P[w, d] = \sum_t P[w|t] P[d|t] P[t]$$



- Differences to SVD:
    - probabilities $P[w|t]$, $P[d|t]$, and $P[t]$ are non-negative and normalized
    - loss function is Kullback-Leibler divergence instead of squared loss

pLSI (Example)

- Topics (10 of 128) extracted from 12K Science Magazine articles

$P[w|t]$

universe	0.0439	drug	0.0672	cells	0.0675	sequence	0.0818	years	0.156
galaxies	0.0375	patients	0.0493	stem	0.0478	sequences	0.0493	million	0.0556
clusters	0.0279	drugs	0.0444	human	0.0421	genome	0.033	ago	0.045
matter	0.0233	clinical	0.0346	cell	0.0309	dna	0.0257	time	0.0317
galaxy	0.0232	treatment	0.028	gene	0.025	sequencing	0.0172	age	0.0243
cluster	0.0214	trials	0.0277	tissue	0.0185	map	0.0123	year	0.024
cosmic	0.0137	therapy	0.0213	cloning	0.0169	genes	0.0122	record	0.0238
dark	0.0131	trial	0.0164	transfer	0.0155	chromosome	0.0119	early	0.0233
light	0.0109	disease	0.0157	blood	0.0113	regions	0.0119	billion	0.0177
density	0.01	medical	0.00997	embryos	0.0111	human	0.0111	history	0.0148

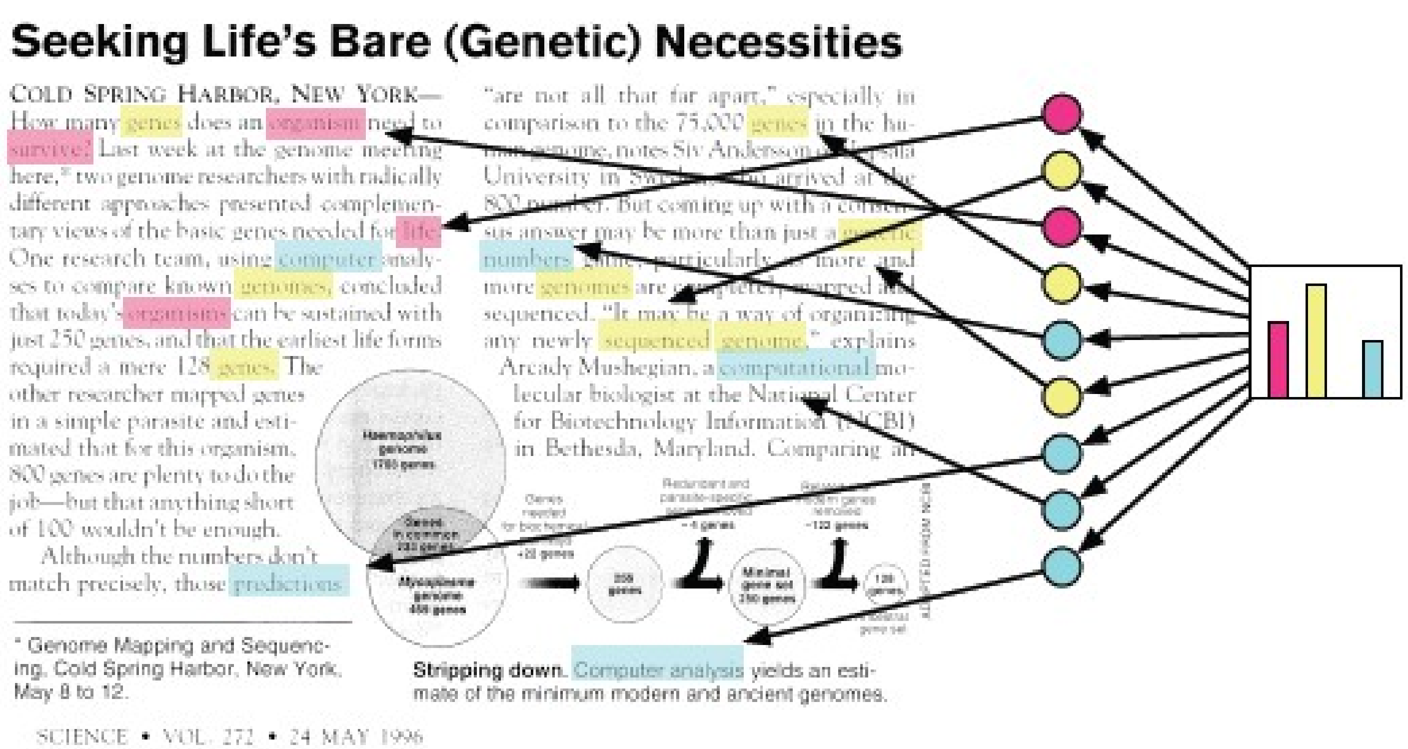
$P[w|t]$

bacteria	0.0983	male	0.0558	theory	0.0811	immune	0.0909	stars	0.0524
bacterial	0.0561	females	0.0541	physics	0.0782	response	0.0375	star	0.0458
resistance	0.0431	female	0.0529	physicists	0.0146	system	0.0358	astrophys	0.0237
coli	0.0381	males	0.0477	einstein	0.0142	responses	0.0322	mass	0.021
strains	0.025	sex	0.0339	university	0.013	antigen	0.0263	disk	0.0173
microbiol	0.0214	reproductive	0.0172	gravity	0.013	antigens	0.0184	black	0.0161
microbial	0.0196	offspring	0.0168	black	0.0127	immunity	0.0176	gas	0.0149
strain	0.0165	sexual	0.0166	theories	0.01	immunology	0.0145	stellar	0.0127
salmonella	0.0163	reproduction	0.0143	aps	0.00987	antibody	0.014	astron	0.0125
resistant	0.0145	eggs	0.0138	matter	0.00954	autoimmune	0.0128	hole	0.00824

Source: Thomas Hofmann, Tutorial at ADFOCS 2004

3. Latent Dirichlet Allocation (LDA)

- Multiple-cause mixture model (MCMM)
- Documents contain **multiple topics**
- Topics are expressed by specific **word distributions**
- LDA provides a **generative model** for this



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

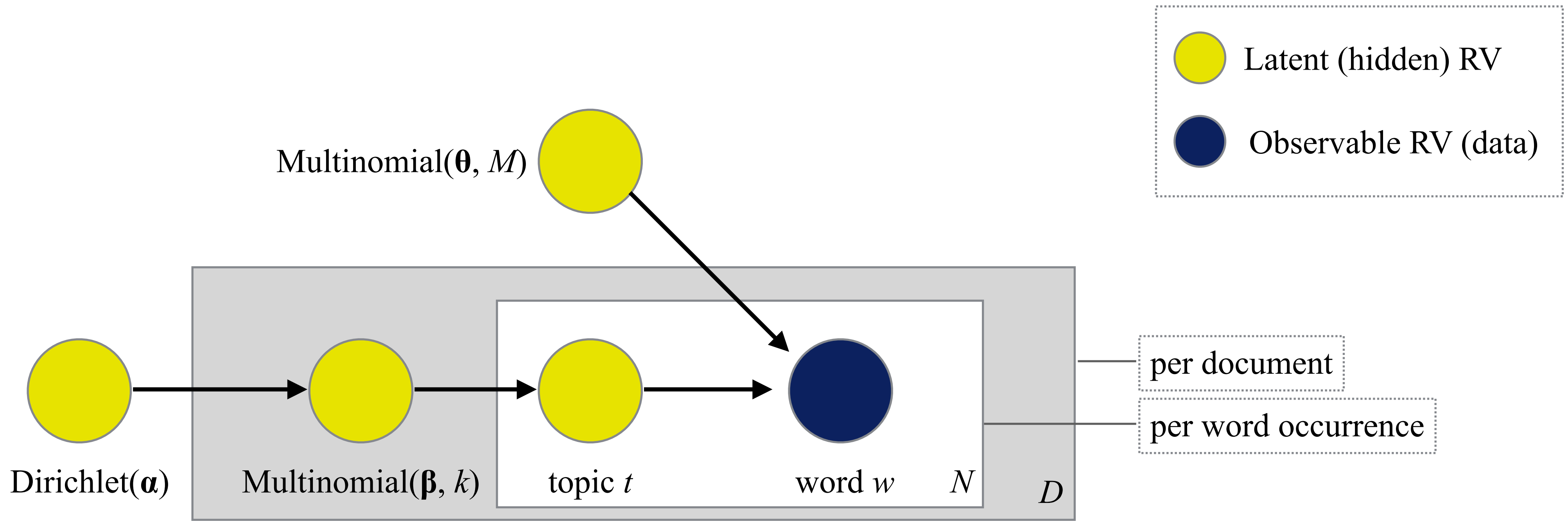
* Genome Mapping and Sequencing, Cold Spring Harbor, New York, May 8 to 12.

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

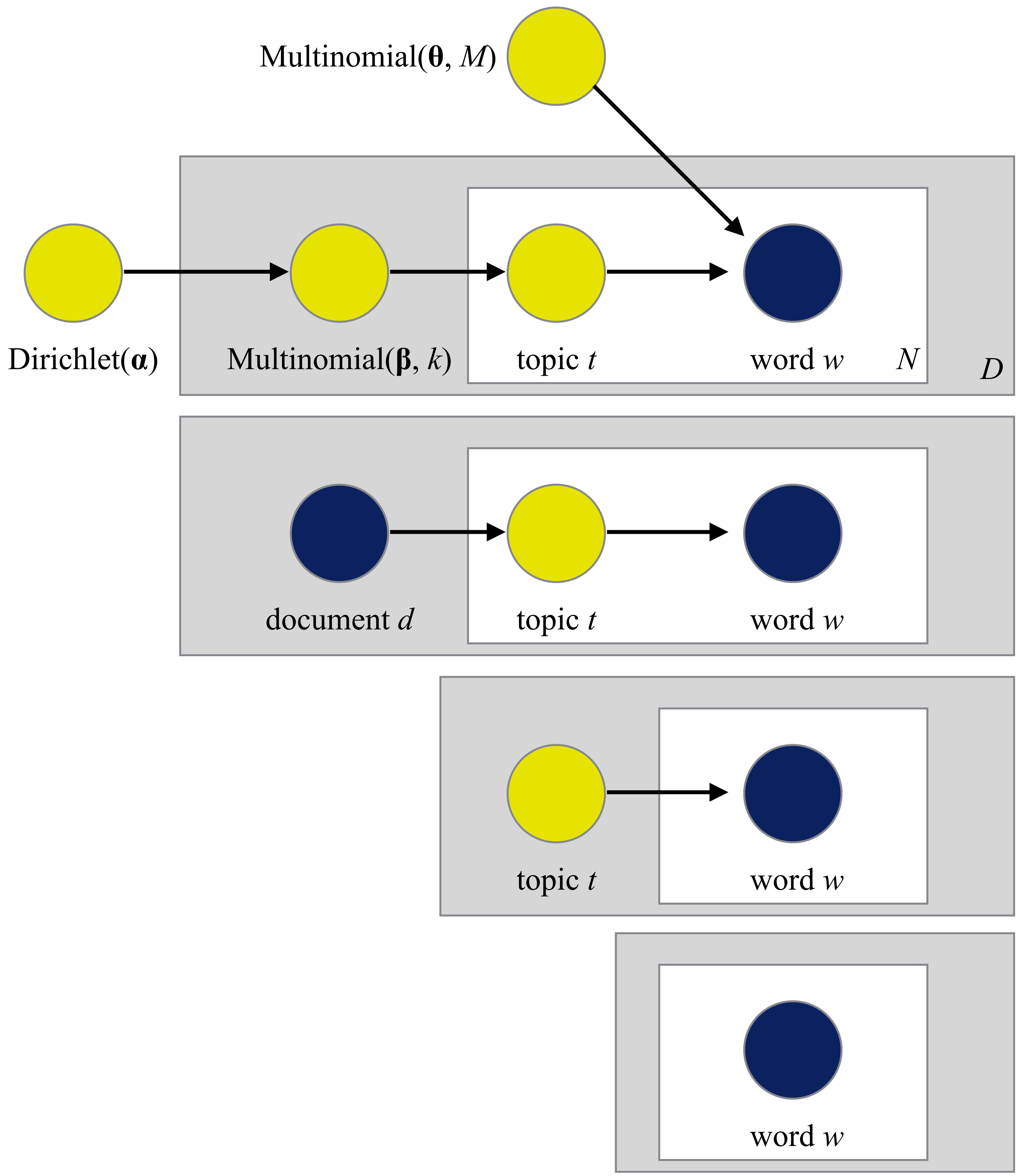
SCIENCE • VOL. 272 • 24 MAY 1996

LDA Generative Model

- For each of the D documents d
  - Choose document length N (# word occurrences) ~ Poisson(λ)
  - Choose topic-probability distribution parameters $\boldsymbol{\beta}$ ~ Dirichlet($\boldsymbol{\alpha}$)
  - For each of the N word occurrences in d (at position n)
    - Choose one of k topics t_n ~Multinomial($\boldsymbol{\beta}$, k)
    - Choose one of M words w_n from per-topic distribution ~ Multinomial($\boldsymbol{\theta}$, M)



Comparison to Other Generative Models



LDA

pLSI

Single-Cause
Mixture of Unigrams

Simple
Unigram Model

Computing LDA

- Dirichlet($\boldsymbol{\alpha}$) probability density function

$$f(\boldsymbol{\beta}|\boldsymbol{\alpha}) = \frac{\Gamma(\sum_{i=1}^{k} \boldsymbol{\alpha}_i)}{\prod_{i=1}^{k} \Gamma(\boldsymbol{\alpha}_i)} \prod_{i=1}^{k} \boldsymbol{\beta}_i^{\boldsymbol{\alpha}_i - 1}$$

  with $\boldsymbol{\alpha}_i \geq 0$, $\boldsymbol{\beta}_\mathrm{i} \geq 0$ and $\Sigma\, \boldsymbol{\beta}_i = 1$

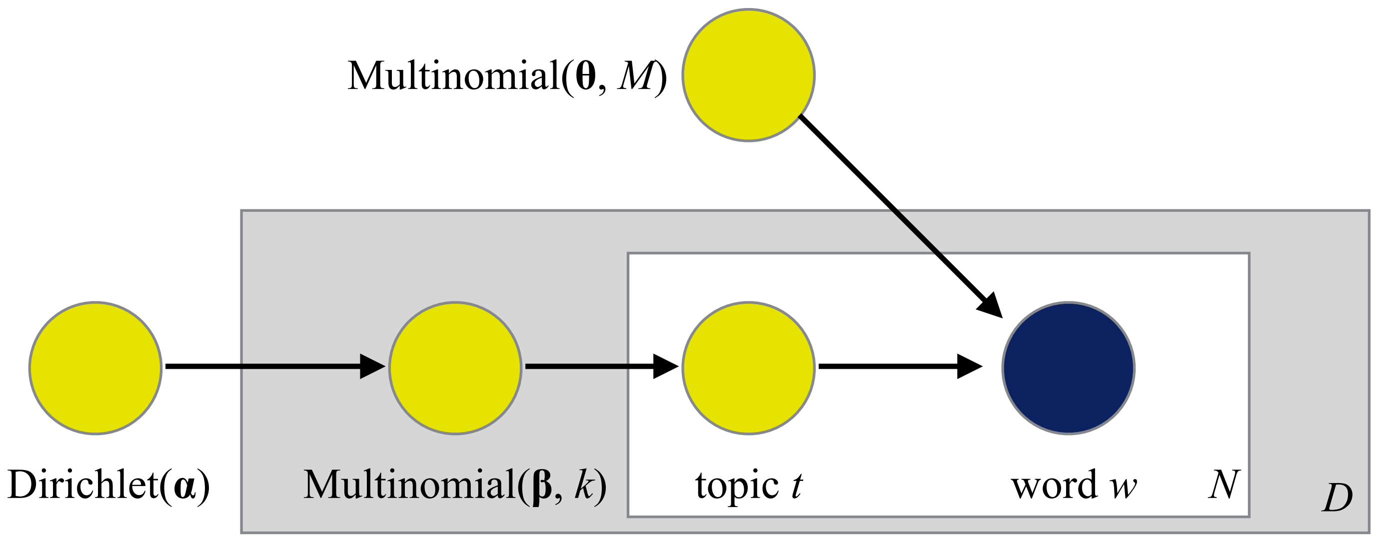
- Probability of document d given $\boldsymbol{\alpha}$ and $\boldsymbol{\theta}$

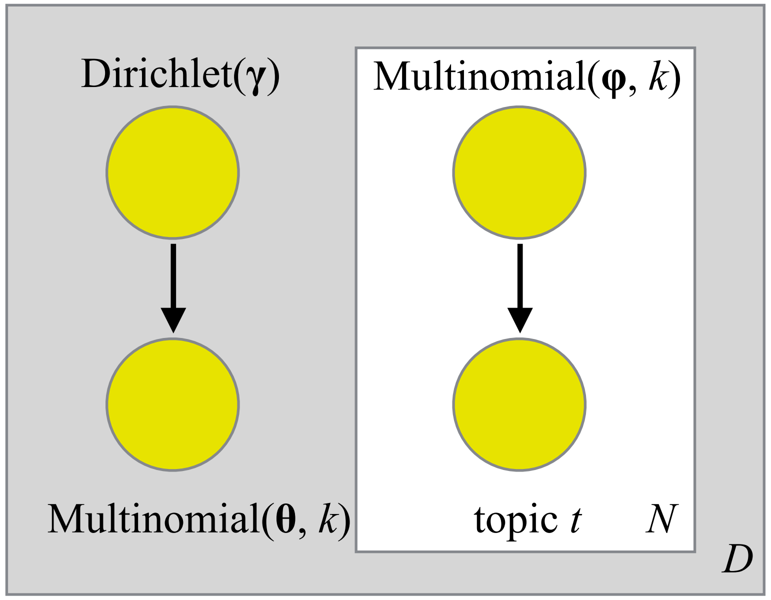
$$P[d|\boldsymbol{\alpha}, \boldsymbol{\theta}] = \int f(\beta|\alpha) \left(\prod_{n=1}^{N} \sum_{t_n=1}^{k} \beta_{t_n} \theta_{t_n, w_n} \right) d\beta$$

- Log-likelihood function (for corpus of D documents) is analytically intractable

Computing LDA (cont'd)

- Parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\theta}$ can be estimated using Expectation Maximization (EM) with lower-bound distributions





- **E-Step**: Determine optimal parameters γ* and φ* of lower-bound distributions given $\boldsymbol{\alpha}^{(i-1)}$ and $\boldsymbol{\theta}^{(i-1)}$
- **M-Step**: Given fixed lower-bound distributions determine parameters $\boldsymbol{\alpha}^{(i)}$ and $\boldsymbol{\theta}^{(i)}$ that maximize log-likelihood
- <u>Full details</u>: [Blei et al '03]

LDA (Example)

- Topics from 5K scientific articles and 16K newswire articles

"Arts"	"Budgets"	"Children"	"Education"
NEW	MILLION	CHILDREN	SCHOOL
FILM	TAX	WOMEN	STUDENTS
SHOW	PROGRAM	PEOPLE	SCHOOLS
MUSIC	BUDGET	CHILD	EDUCATION
MOVIE	BILLION	YEARS	TEACHERS
PLAY	FEDERAL	FAMILIES	HIGH
MUSICAL	YEAR	WORK	PUBLIC
BEST	SPENDING	PARENTS	TEACHER
ACTOR	NEW	SAYS	BENNETT
FIRST	STATE	FAMILY	MANIGAT
YORK	PLAN	WELFARE	NAMPHY
OPERA	MONEY	MEN	STATE
THEATER	PROGRAMS	PERCENT	PRESIDENT
ACTRESS	GOVERNMENT	CARE	ELEMENTARY
LOVE	CONGRESS	LIFE	HAITI

The William Randolph Hearst Foundation will give $1.25 million to Lincoln Center, Metropolitan Opera Co., New York Philharmonic and Juilliard School. "Our board felt that we had a real opportunity to make a mark on the future of the performing arts with these grants an act every bit as important as our traditional areas of support in health, medical research, education and the social services," Hearst Foundation President Randolph A. Hearst said Monday in announcing the grants. Lincoln Center's share will be $200,000 for its new building, which will house young artists and provide new public facilities. The Metropolitan Opera Co. and New York Philharmonic will receive $400,000 each. The Juilliard School, where music and the performing arts are taught, will get $250,000. The Hearst Foundation, a leading supporter of the Lincoln Center Consolidated Corporate Fund, will make its usual annual $100,000 donation, too.

<u>Source</u>: [Blei et al. '03]

Summary of III.5

- **Latent topic models**
  consider word co-occurrence and implicitly handle synonymy etc.
- **Latent Semantic Indexing (LSI)**
  applies SVD to term-document matrix $\boldsymbol{A}$
- **Probabilistic Latent Semantic Indexing (pLSI)**
  uses a non-negative probabilistic decomposition of $\boldsymbol{A}$
- **Latent Dirichlet Allocation (LDA)**
  uses a probabilistic generative model

Additional Literature for III.5

- **D. M. Blei, A. Y. Ng, M. I. Jordan**: *Latent Dirichlet Allocation*, Journal of Machine Learning Research 3:993-1022, 2003
- **D. M. Blei**: *Probabilistic Topic Models*, CACM 55(4):77-84, 2012
- **S. Deerwester, S. Dumais, G. W. Furnas, T. K. Landauer, R. Hashman**: *Indexing by Latent Semantic Analysis*, 1990
- **T. Hofmann**: *Probabilistic Latent Semantic Indexing*, SIGIR 1999