



UNSTRUCTURED TOPIC MODELLING

Using 'tm' and 'topicsmodels'
package in R

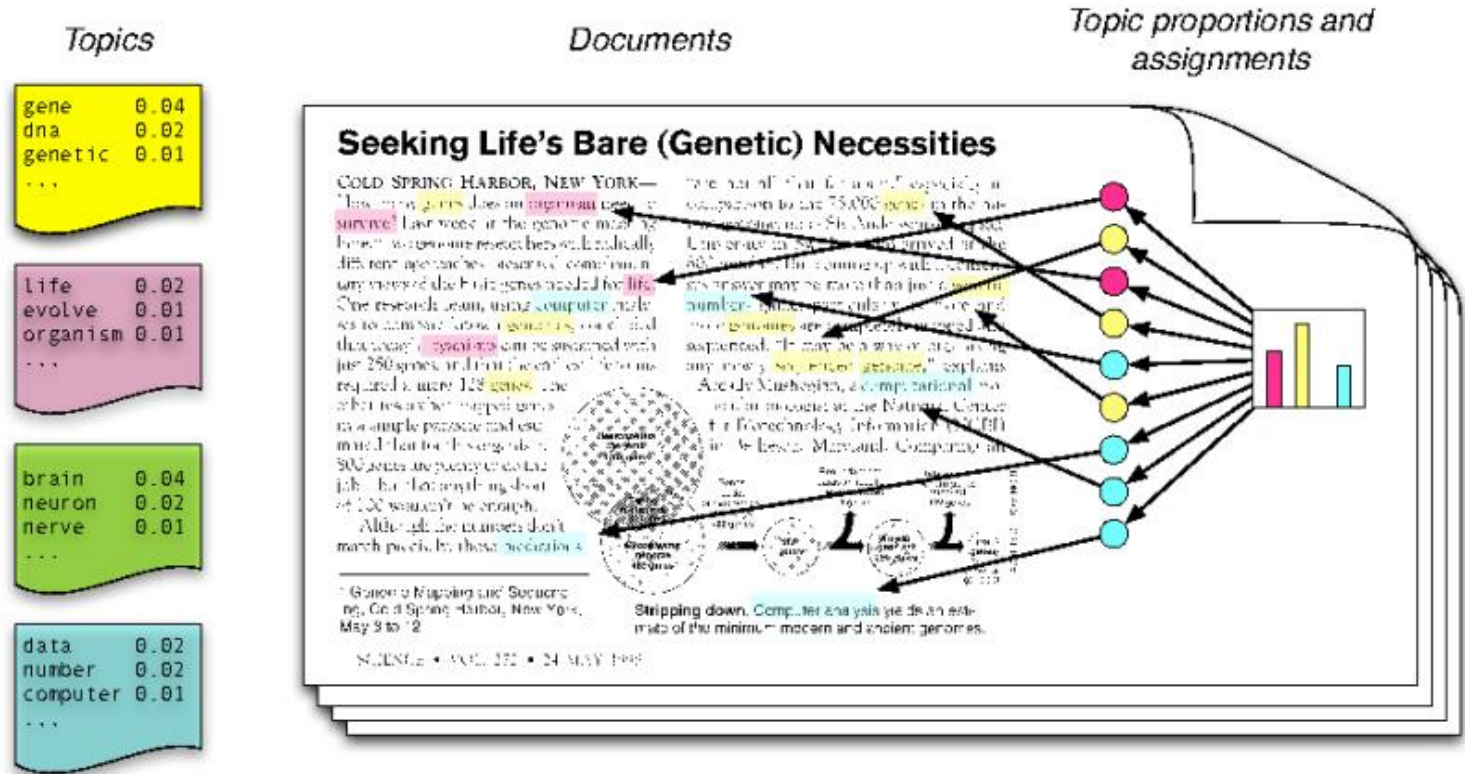
7 LEARNING OUTCOMES (LO)

1. Understand the concept of topic modelling
2. Overview of the process
3. Awareness of the R 'tm' & 'topicmodels' package
4. Grasp of the data structures and processing used
5. Understanding of the concept, parameters and application of Latent Dirichlet Allocation
6. Broad understanding of Latent Semantic Indexing
7. Translating analysis into insight



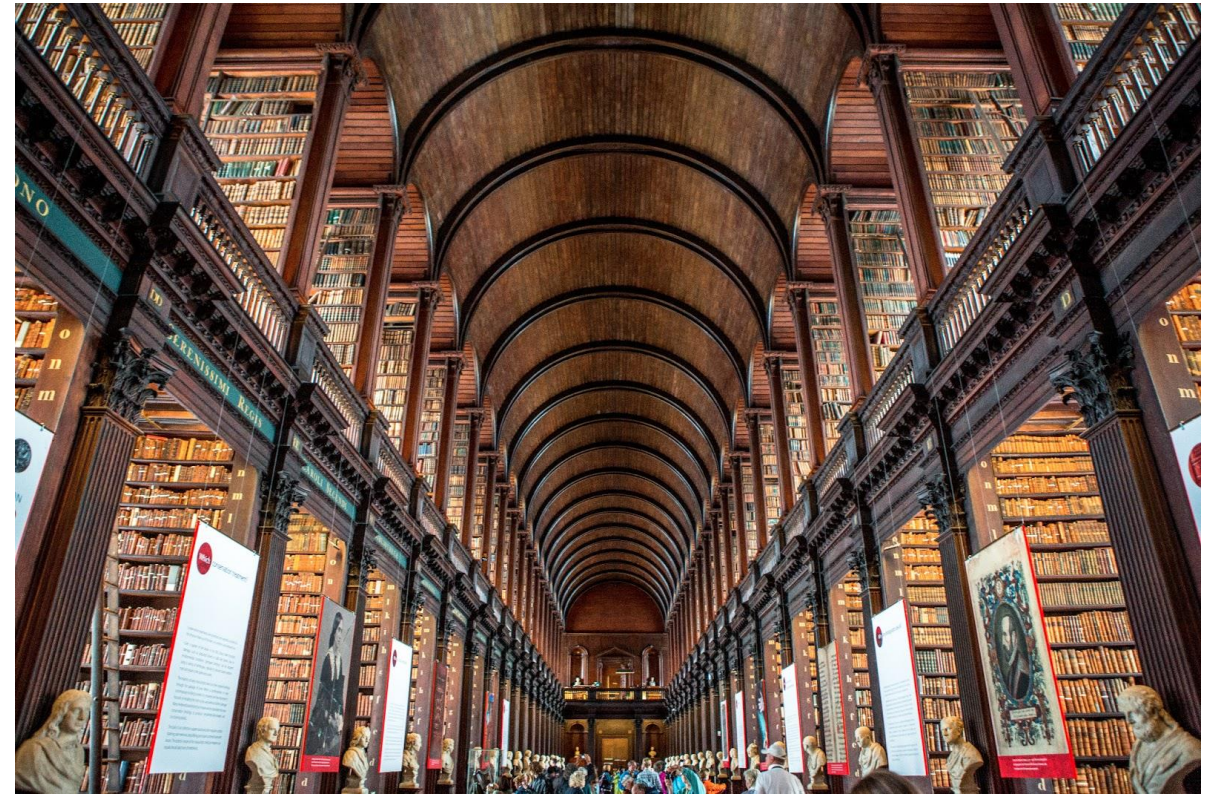
WHAT IS TOPIC MODELLING?

Grouping documents based on the probability of words occurring in each document



WHY IS IT IMPORTANT?

- Discover topics in large groups of documents
- Use these labels to understand the body of text and documents more effectively



CONTEXT IS KEY

Blind application of complex modelling will yield results which deliver incorrect classification

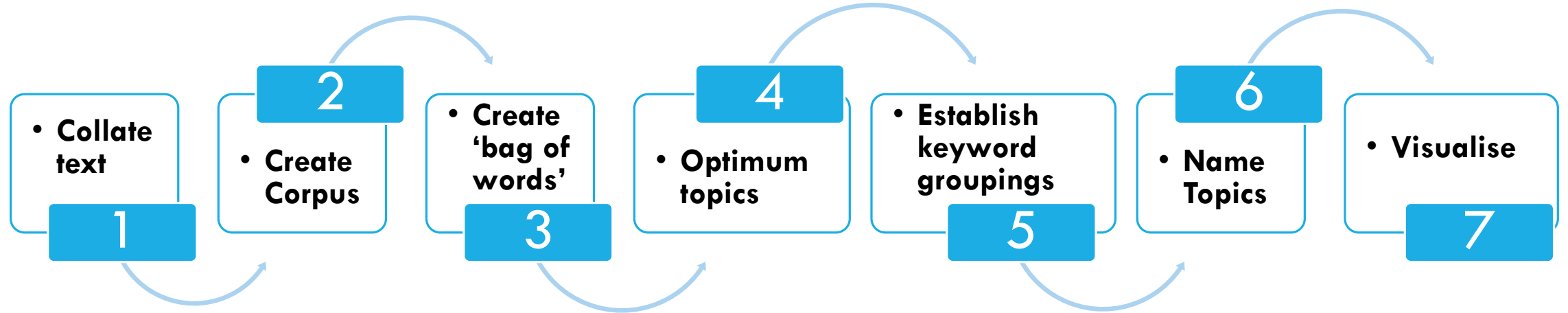
The final deliverable and key features must be defined before embarking on the analysis

Let's eat grandma!



Let's eat, grandma!

**PUNCTUATION
SAVES LIVES!**

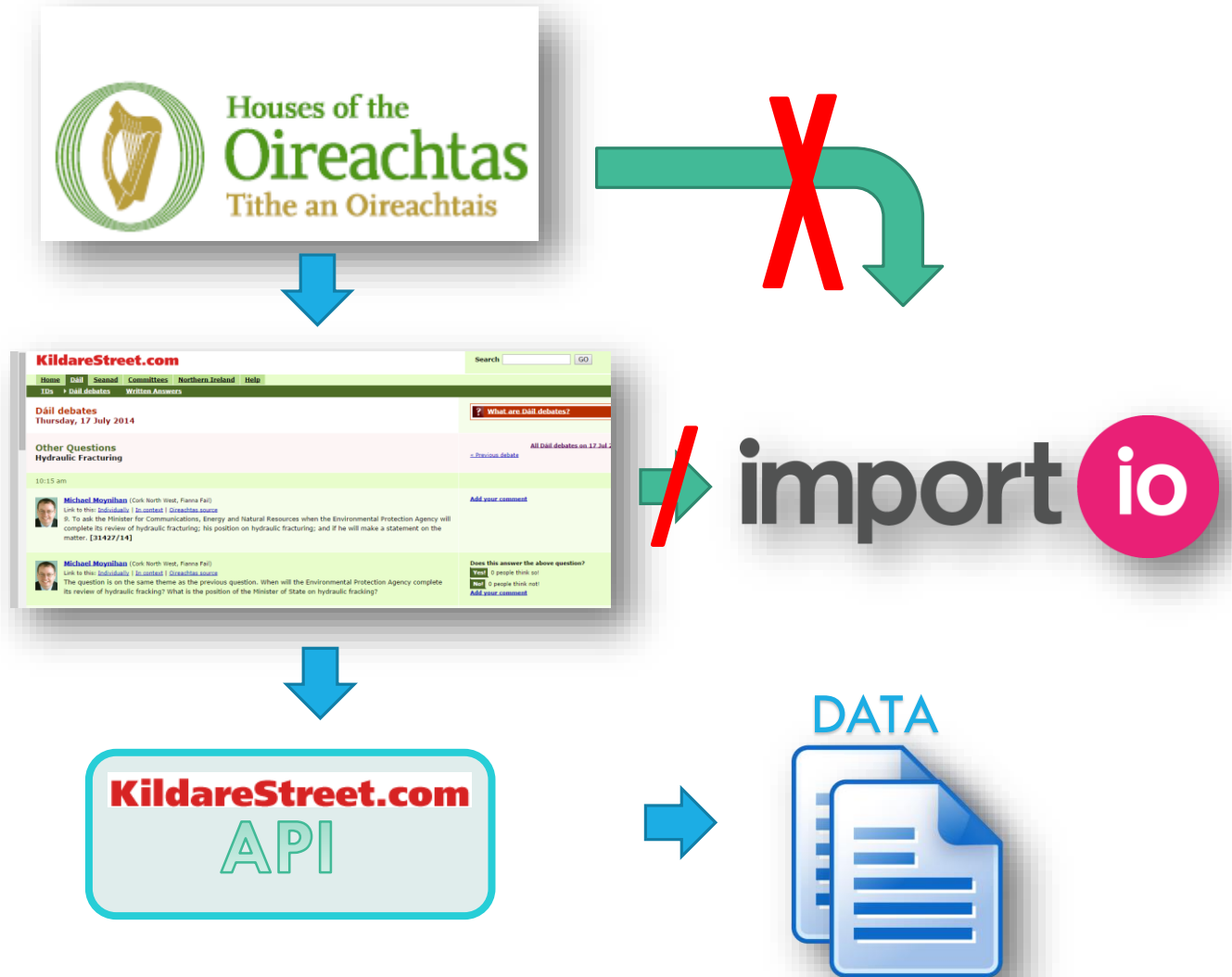


HOW TO USE IT?

There are 7 key stages to model topics in a data set effectively

SOURCING THE DATA

- Using a public available crawler which obeys the robot.txt was unsuccessful at both Public and privately run websites
- Modern websites will provide an API to bulk query the data in XML or JSON format



DATA CARPENTRY

- No data set is ever ready to operate on 'out-of-the-box'
- Challenges included:
 - Character encoding
 - Irish characters
 - 2-3 features in a field

'tm' & Regular Expression to the Rescue!



THERE ARE A NUMBER OF PACKAGES REQUIRED SHAPING YOUR DATA WITH 'TM' AND MODELLING THE TOPICS WITH 'TOPICMODELS' ARE THE MOST FUNDAMENTAL

'tm' package

Methods for data import, corpus handling, preprocessing, metadata management, and creation of term-document matrices.

Get familiar with it, it is important

'topicmodels' package

Provides an interface to the *C code for Latent Dirichlet Allocation (LDA) models and Correlated Topics Models (CTM) the **C++ code for fitting LDA models using Gibbs sampling

Alternatives -> LDA, Mallet, Gensim etc.
but this is the easiest to manage

* David M. Blei and co-authors

** Xuan-Hieu Phan and co-authors.

WHAT IS A TEXT CORPUS?

A **corpus** (plural *corpora*) is a large and structured set of texts

In R a corpus is an abstract concept.

A plain text files in the directory `txt` containing Latin (lat) texts by the Roman poet Ovid can be read in with following code:

```
corp <- Corpus(VectorSource(daildata))
```

TOKENISATION- *WHAT IS A DOCUMENT TERM MATRIX (DTM)?*

A **document-term matrix** or **term-document matrix** is a mathematical matrix that describes the frequency of terms that occur in a collection of documents.

The Corpus is converted into a set of tokens of N word length, and the 'occurrence frequency' of each 'token' is stored in the matrix

Before a DTM can be created the tm package is then used to clean the corpus

```
dtm.control <- list(  
  tolower = TRUE,  
  removePunctuation = TRUE,  
  removeNumbers = TRUE,  
  stopwords = stopwords("english"),  
  stemming = TRUE,  
  wordLengths = c(3, Inf),  
  weighting = weightTf  
)  
corp.dtm <- DocumentTermMatrix(corp, control = dtm.control)
```

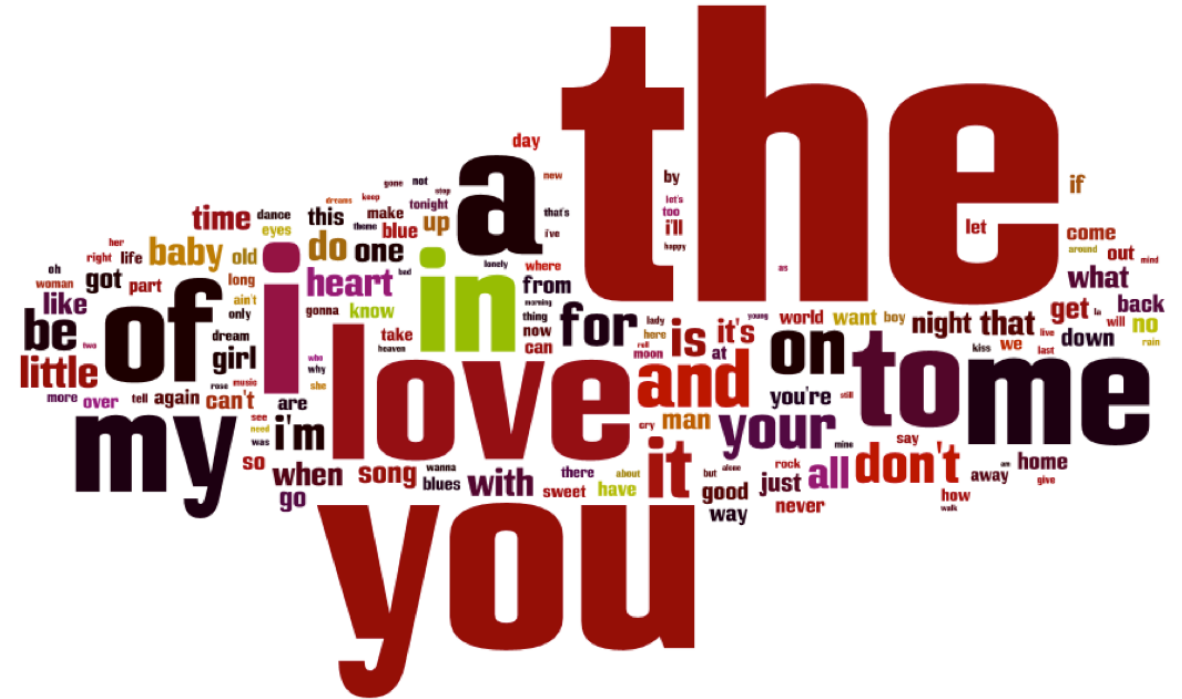
The properties of a DTM give an explanation of its purpose

```
DocumentTermMatrix (documents: 6, terms: 4)  
Non-/sparse entries: 8/16  
Sparsity : 67%  
Maximal term length: 8  
Weighting : term frequency (tf)
```

STOPWORDS!

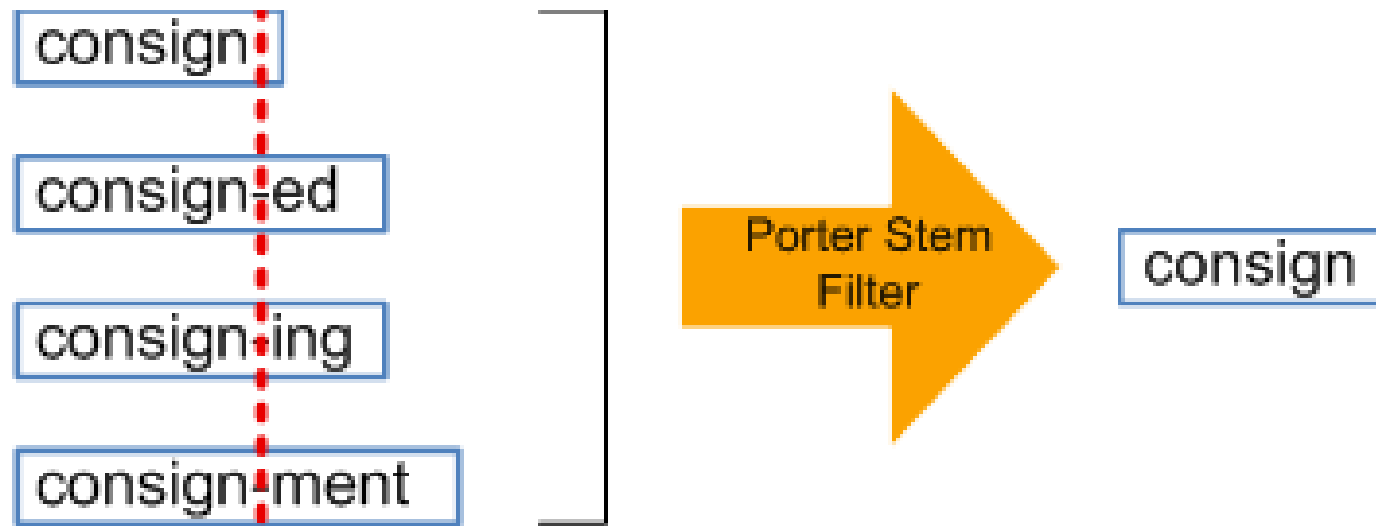
- Words such as the, of, to, and, a, etc. will pervade the learned topics, hiding the actual topics
- Stop-words must be removed in a pre-processing step.
- While stop-word removal does a good job at solving this problem, it is an ad hoc measure that results in a model resting on a non-coherent theoretical basis.
- Stop-word lists must often be domain-dependent and there are inevitably cases where filtering causes noise, or missing patterns that may be of interest to us

After processing, be wary of empty 'documents'



```
dailystops =  
c("text", "also", "minister", "I", "this", "the", "deputy", "deputies", "government",  
  "we", "people", "matter", "issue", "country", "it", "to", "state")
```

```
corp <- tm_map(corp, function(x) removeWords(x,dailstops ))
```

STEMMING

make life easier

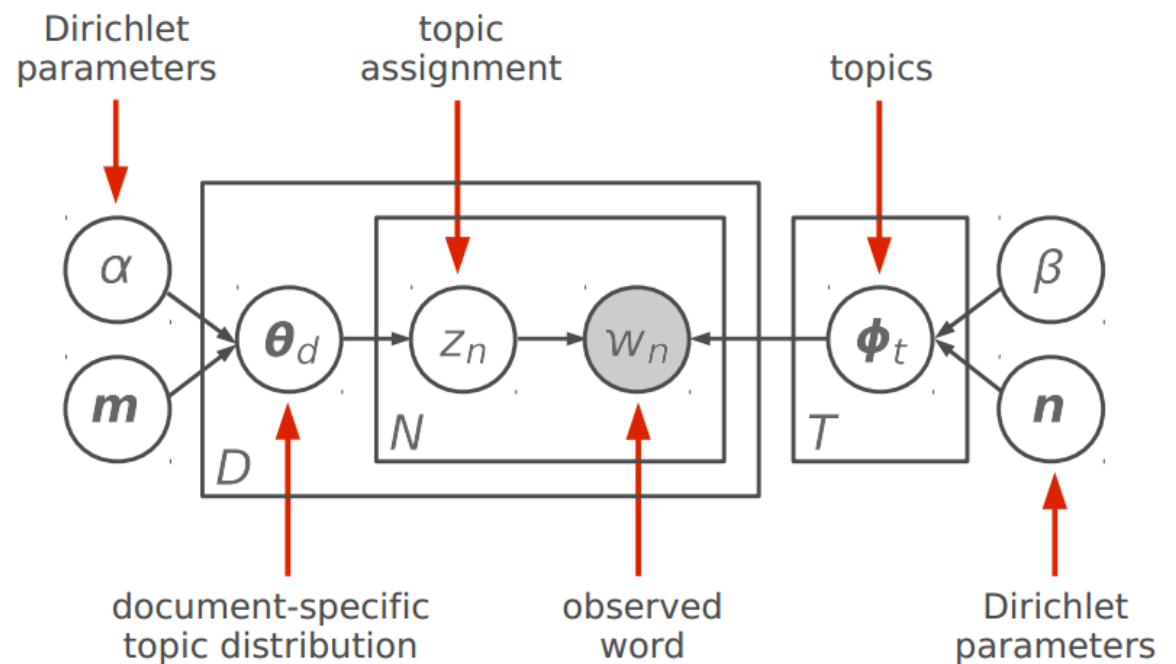
LATENT DIRICHLET ALLOCATION

(Blei et al., 2003)

LDA represents documents as **mixtures of topics** that spit out words with certain probabilities.

It creates a **generative model** for a collection of documents, LDA then tries to backtrack from the documents to find a set of topics that are likely to have generated the collection.

Word probabilities are maximized by dividing the words among the topics. (More terms means more mass to be spread around.) In a mixture, **this is enough to find clusters of co-occurring words**



```
Lda.mdoel = LDA(rowcorp.dtm2, topiccount, method = "Gibbs",  
control = list(burnin = burnin, iter = iter, keep = keep)))
```

IN LAYMANS TERMS

1. Suppose you've just moved to a new city. You want to know where the other hipsters and data science geeks tend to hang out. Of course, as a hipster, you know you can't just ask, so what do you do?
2. So you scope out a bunch of different establishments (**documents**) across town, making note of the people (**words**) hanging out in each of them (e.g., Alice hangs out at the mall and at the park, Bob hangs out at the movie theater and the park, and so on).
3. Crucially, you don't know the typical interest groups (**topics**) of each establishment, nor do you know the different interests of each person.
4. So you pick some number K of interests to learn (i.e., you want to learn the K most important kinds of interests people may have), and start by making a guess as to why you see people where you do
5. Go through each place and person over and over again. Your guesses keep getting better and better
6. For each interest ('topic'), you can count the people assigned to that interest to figure out what people have this particular interest. By looking at the people themselves, you can interpret the category as well (e.g., if category X contains lots of tall people wearing jerseys and carrying around basketballs, you might interpret X as the "basketball players" group).



```

harmonicMean <- function(logLikelihoods, precision=2000L) {
  llMed <- median(logLikelihoods)
  as.double(llMed - log(mean(exp(-mpfr(logLikelihoods,
                                     prec = precision) + llMed))))
}

# The log-likelihood values are then determined by first fitting the model using for example
k = 20
burnin = 1000
iter = 1000
keep = 50

fitted <- LDA(corp.dtm, k = k, method = "Gibbs", control = list(burnin = burnin, iter = iter, keep = keep) )

# where keep indicates that every keep iteration the log-likelihood is evaluated and stored. This returns all log-likelihood values including burnin,
# i.e., these need to be omitted before calculating the harmonic mean:

logLiks <- fitted@logLiks[-c(1:(burnin/keep))]

# assuming that burnin is a multiple of keep and

harmonicMean(logLiks)

# generate numerous topic models with different numbers of topics, for such a small corpus >15 is probably a bad idea
sequ <- seq(2, 15, 1) # in this case a sequence of numbers from 1 to 50, by ones.
fitted_many <- lapply(sequ, function(k) LDA(corp.dtm, k = k, method = "Gibbs", control = list(burnin = burnin, iter = iter, keep = keep) ))

# extract logliks from each topic
logLiks_many <- lapply(fitted_many, function(L) L@logLiks[-c(1:(burnin/keep))])

# compute harmonic means
hm_many <- sapply(logLiks_many, function(h) harmonicMean(h))

# inspect
plot(sequ, hm_many, type = "l")

```

MODEL SELECTION BY HARMONIC MEAN

- MARTIN PONWEISER

Optimum number of
Topics (1/2)

***The drawback however is that the estimator might have infinite variance.
The problem is that adding more parameters to the model will always increase the likelihood.*

OPTIMUM NUMBER OF TOPICS (2/2)

Akaike information criterion – AIC

AIC deals with the trade-off between the goodness of fit of the model and the complexity of the model.

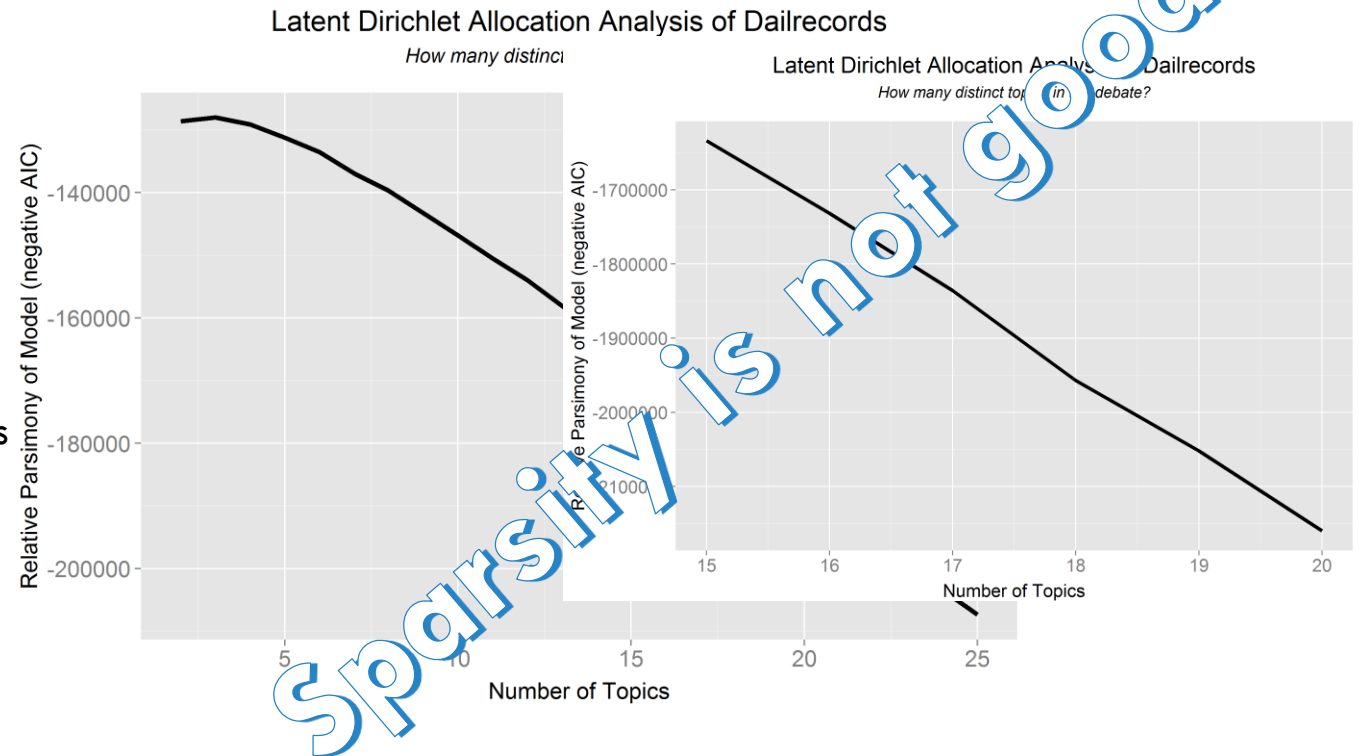
$$AIC = 2k - 2\ln(L)$$

k = no. of parameters

L = maximized value of the likelihood function



Winner of
Kyoto prize
2006



```
#only going as high as 25, as manually classifying any higher than that will take a lot of manual effort
ks <- seq(2, 25, by = 1)
#creating 24 lda models of the corpus
models <- lapply(ks, function(k) LDA(rowcorp.dtm, k, method = "Gibbs", control = list(burnin = burnin, ite

#maximized value of likelihood function
logLiks <- lapply(models, function(L) L@logLiks[-c(1:(burnin/keep))])

hm <- sapply(logLiks, function(h) harmonicMean(h))
#number of Parameters
k = sapply(models, function(L) sum(length(L@beta) + length(L@gamma)))

#getting the AIC of the model
AICS = 2*k - 2*hm
```

THE MODEL STILL NEEDS TO BE CONVERTED TO 'TOPICS'

Using the LDATool package the topic groups can be graphed using Kullback-Leibler divergence

Defining the Saliency and Conditioning of each word in the topic group

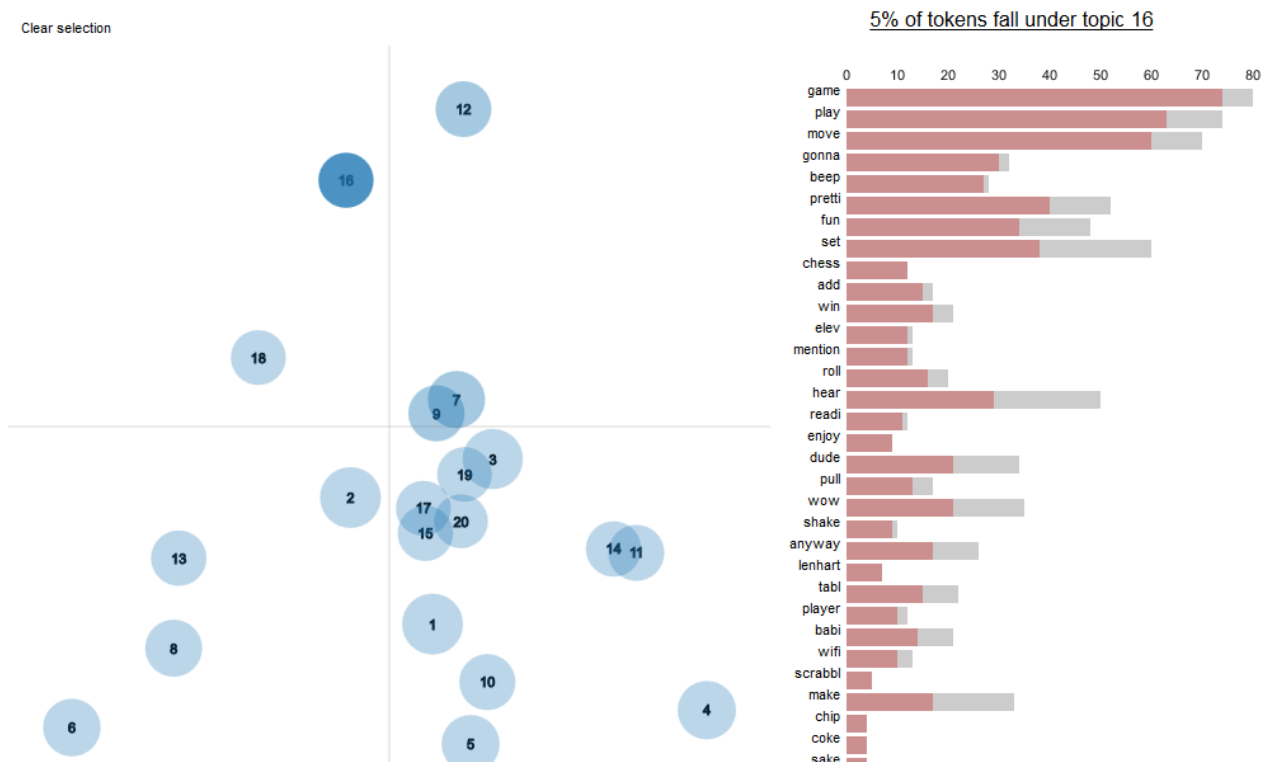
This allows us convert the groupings into meaningful topics

```
# Extract the 'guts' of the optimal model
doc.id <- opt@wordassignments$i
token.id <- opt@wordassignments$j
topic.id <- opt@wordassignments$v
vocab <- opt@terms

# Get the phi matrix using LDAviz
dat <- getProbs(token.id, doc.id, topic.id, vocab, K = max(topic.id), sort.topics = "byTerms")
phi <- t(dat$phi.hat)

token.frequency <- as.numeric(table(token.id))
topic.id <- dat$topic.id
topic.proportion <- as.numeric(table(topic.id)/length(topic.id))

# Run the visualization locally using LDAvis
z <- check.inputs(K=max(topic.id), w=max(token.id), phi, token.frequency, vocab, topic.proportion)
json <- with(z, createJSON(K=max(topic.id), phi, token.frequency,
                           vocab, topic.proportion))
```



EACH TOPIC CONSISTS OF WORDS WITH A LIKELYHOOD OF THERE OCCURRENCE IN TOPIC DISTRIBUTION

Magdelene Laundry	Non-topic	Bank Debit	Shannon Airport No fly	Water Charges
0.103	0.013	0.021	0.034	0.061
woman	one	debt	force	water
0.019	0.013	0.02	0.029	0.03
mother	u	bank	defence	charge
0.018	0.01	0.013	0.018	0.024
laundry	time	billion	airport	irish
0.015	0.008	0.013	0.013	0.016
institution	would	fiscal	member	company
0.015	0.008	0.012	0.012	0.013
home	know	economy	data	tax
0.012	0.008	0.011	0.012	0.011
report	many	economic	shannon	service
0.011	0.008	0.011	0.011	0.011
life	year	year	operation	pay
0.01	0.008	0.011	0.011	0.01
baby	say	irish	international	bord
0.01	0.008	0.01	0.01	0.01

Taking the top 10 words we can check for skew or 'strong' words in each topic

THE TOPICS NEED TO BE 'NAMED', THIS IS A MANUAL PROCESS, AS WHILE THE KEY WORDS INDICATE THE GROUP, ONLY THE DOCUMENTS WITH HIGH TOPIC SCORES REVEAL THE TOPIC NAME

Text for topic 8	Topic Score
The banks have a veto.	1.35
The banks were not dealing with them. Will there will be independent oversight of the deals the banks strike with individuals? Will this oversight have teeth and ensure that the restructuring that may take place will be effective and will have real meaning for people who find themselves in arrears?	0.953571429
There are not 30,000 families facing eviction, nor will there be. However, there are 30,000 families who would be facing that prospect if we had not put in place the measures we have put in place to deal with the issue of mortgage arrears.	0.948684211
Thousands of Irish families are facing eviction. Ulster Bank alone has 4,700 repossession cases before the courts.	0.925
David Hall has said that up to 50 new repossession cases are coming before the courts every month.	0.91875
There is enormous relief for distressed mortgage holders and their families when it becomes clear that they are not in danger of losing their houses.	0.91875

Topic	Frequency	% occurrence
Dail_order of business		
Passing Legislation		
Jobs and enterprise		
Taxation and the Budget		
Deputies been called out		
Question time		
Rent,property and local housing		
Parties naming each other		
Regional discussion		
HSE		
Non-topic		
Medical Cards		
Credit Union		

EACH RESPONSE CAN HAVE A LIKELIHOOD OF MULTIPLE TOPICS, FOR BREVITY WE NEED TO SELECT THE MOST LIKELY

@gamma is the probability of each topic for each response, but taking the 'max' value each TD's response can be tagged with 'the 'most likely' topic

```
#getting table of who spoke about what
gammaDF <- as.data.frame(opt@gamma)
names(gammaDF) <- c(1:top.opt)

toptopics <- as.data.frame(cbind(document = row.names(gammaDF), topic = apply(gammaDF, 1, function(x) names(gammaDF)[which(x==max(x))])))
#remove the rows that made not sense
parsedset <- data[-blankrow[],]
finaltable <- cbind(parsedset, toptopics)

data <- write.csv(finaltable, file = "dailrecordsparsed.csv")
```

AFTER “GENERAL DISCOURSE” HAS BEEN FILTERED OUT (>50% OF CORPUS) SOME TOPICS DOMINATE

Topic	Share of discourse
Jobs and enterprise	
Taxation and the Budget	
Rent,property and local housing	
Regional discussion	
HSE	
Medical Cards	
Credit Union	
Family and child services	
Irish_discourse	
Europe	
Northern Ireland	
Criminal Justice System	
Garda / GSOC et.c	
Flood damage	
Geopolitics / foreign policy	
Agriculture	
Coillte	
School and educations	
Social_housing	
Mortgage crisis	
Bank Debit	
Water Charges	
Funding for govovernment projects	
Magdelene Laundry	
Shannon Airport No fly	

LDA V. LATENT SEMANTIC INDEXING (LSI)

LSI is based on the principle that words that are used in the same contexts tend to have similar meanings. A key feature of LSI is its ability to extract the conceptual content of a body of text by establishing associations between those terms that occur in similar contexts

LDA

1. Topics typically look more coherent and easier to interpret.
2. Better at classifying static corpuses
3. Ability to shard and map-reduce
4. Priors are critical


LSI

1. LSI requires relatively high computational performance and memory
2. Topics are difficult to interrupt
3. Better for continuous feeds as models have larger scope to change



**UNDERSTANDING OF THE TEXT AND
DOCUMENT STRUCTURE IS CRITICAL**

Conclusion



baby steps...
to big dreams.

SO MUCH MORE TO EXPLORE!

- Modelling with Stopwords
- N-grams > 1
- Bayesian approach to Priors
- Back-fitting corpus
- Structural Topic Models (STM) package

THE SECOND ANNUAL FRONT LINE DEFENDERS LECTURE
in association with University College Dublin and Trinity College Dublin

IS IT POSSIBLE TO BE SAFE ONLINE?

Human Rights Defenders
and the Internet

Bruce Schneier

6.30pm, Monday, 6th October 2014

Trinity Biomedical Science Institute
152-160 Pearse Street, Dublin 2



TRINITY
COLLEGE
DUBLIN TCD School of Law



UCD School of Philosophy
UCD School of Politics and International Relations (SPIRe)
UCD Sutherland School of Law

Admission: €5 unwaged / €15 waged
Tickets Available at:
<https://bruceschneierdublin2014.eventbrite.ie>

**f FRONT LINE
DEFENDERS**
THE INTERNATIONAL FOUNDATION FOR THE
PROTECTION OF HUMAN RIGHTS DEFENDERS

REFERENCES

<http://blog.echen.me/2011/08/22/introduction-to-latent-dirichlet-allocation/>

<http://stackoverflow.com/questions/21355156/topic-models-cross-validation-with-loglikelihood-or-perplexity/21394092#21394092>

<https://stats.stackexchange.com/questions/25820/why-lda-latent-dirichlet-allocation-works-i-e-why-put-co-occurring-words-together>

http://en.wikipedia.org/wiki/Non-negative_matrix_factorization

https://de.dariah.eu/tatom/topic_model_python.html

http://en.wikipedia.org/wiki/Gibbs_sampling

http://en.wikipedia.org/wiki/Latent_Dirichlet_allocation

http://en.wikipedia.org/wiki/Latent_semantic_indexing

http://en.wikipedia.org/wiki/Latent_semantic_analysis

<https://lists.cs.princeton.edu/pipermail/topic-models/2006-September/000008.html>

<https://groups.google.com/forum/#!topic/gensim/ufqclSEJees>

SAMPLING METHODS?

Gibbs sampling method

Gibbs sampling is commonly used as a means of statistical inference, especially Bayesian inference. It is a randomized algorithm (i.e. an algorithm that makes use of random numbers, and hence may produce different results each time it is run)

With Gibbs sampling, we can approximate the posterior probability as tight as we want.

How does it work?

Technically LDA Gibbs sampling works because we intentionally set up a Markov chain that converges into the posterior distribution of the model parameters, or word–topic assignments

The equations for collapsed Gibbs sampling, there is a factor for words, another for documents. Probabilities are higher for assignments that "don't break document boundaries", that is, words appearing in the same document have a slightly higher odds of ending up in the same topic.

The same holds for document assignments, they to a degree follow "word boundaries". These effects mix up and spread over clusters of documents and words.

OTHER METHODS OF OPTIMUM TOPIC SELECTION

Perplexity

Is a measurement of how well a probability distribution or probability model predicts a sample

A weighted geometric average of the inverses of the probabilities.

Hierarchical Dirichlet Processes

A common base distribution is selected which represents the countably-infinite set of possible topics for the corpus, and then the finite distribution of topics for each document is sampled from this base distribution.

WHY PRIORS MATTER!

Careful thinking about priors can yield new insights

- e.g., priors and STOPWORD handling are related

For LDA the choice of prior is surprisingly important:

- Asymmetric prior for document-specific topic distributions
- Symmetric prior for topic-specific word distributions



Almost all work on LDA uses symmetric Dirichlet priors

- Two scalar concentration parameters: α and β
 - Concentration parameters are usually set heuristically
 - Some recent work on inferring optimal concentration parameter values from data (Asuncion et al., 2009)



MATHEMATICAL THEORY



LATENT DIRICHLET ALLOCATION(1/2)

LDA represents documents as **mixtures of topics** that spit out words with certain probabilities. It assumes that documents are produced in the following fashion: when writing each document, you

Assuming this generative model for a collection of documents, LDA then tries to backtrack from the documents to find a set of topics that are likely to have generated the collection.

Word probabilities are maximized by dividing the words among the topics. (More terms means more mass to be spread around.) In a mixture, **this is enough to find clusters of co-occurring words**. In LDA, the Dirichlet on the topic proportions can encourage sparsity, i.e., a document is penalized for using many topics. Loosely, this can be thought of as **softening the strict definition of “co-occurrence” in a mixture model**. This flexibility leads to sets of terms that more tightly co-occur

Decide on the number of words N the document will have (say, according to a Poisson distribution).

Choose a topic mixture for the document (according to a Dirichlet distribution over a fixed set of K topics). For example, assuming that we have the two food and cute animal topics above, you might choose the document to consist of $1/3$ food and $2/3$ cute animals.

Generate each word w_i in the document by:

- First picking a topic (according to the multinomial distribution that you sampled above; for example, you might pick the food topic with $1/3$ probability and the cute animals topic with $2/3$ probability).
- Using the topic to generate the word itself (according to the topic's multinomial distribution). For example, if we selected the food topic, we might generate the word “broccoli” with 30% probability, “bananas” with 15% probability, and so on.

LATENT DIRICHLET ALLOCATION(2/2)

In probability and statistics, the **Dirichlet distribution** (after Peter Gustav Lejeune Dirichlet), often denoted $\text{Dir}(\alpha)$, is a family of continuous multivariate probability distributions parametrized by a vector α of positive reals. It is the multivariate generalization of the beta distribution.^[1] Dirichlet distributions are very often used as prior distributions in Bayesian statistics, and in fact the Dirichlet distribution is the conjugate prior of the categorical distribution and multinomial distribution. That is, its probability density function returns the belief that the probabilities of K rival events are θ_k given that each event has been observed n_k times.

The infinite-dimensional generalization of the Dirichlet distribution is the Dirichlet process.

*The Dirichlet on the topic proportions can encourage sparsity, i.e., a document is penalized for using many topics. Loosely, this can be thought of as **softening the strict definition of “co-occurrence” in a mixture model. This flexibility leads to sets of terms that more tightly co-occur***

VEM

EM is a partially non-Bayesian, maximum likelihood method. Its final result gives a probability distribution over the latent variables (in the Bayesian style) together with a point estimate for ϑ (either a maximum likelihood estimate or a posterior mode). We may want a fully Bayesian version of this, giving a probability distribution over ϑ as well as the latent variables. In fact the Bayesian approach to inference is simply to treat ϑ as another latent variable. In this paradigm, the distinction between the E and M steps disappears. If we use the factorized Q approximation as described above (variational Bayes), we may iterate over each latent variable (now including ϑ) and optimize them one at a time. There are now k steps per iteration, where k is the number of latent variables. For graphical models this is easy to do as each variable's new Q depends only on its Markov blanket, so local message passing can be used for efficient inference.

OTHER NEAT FUNCTIONS IN TOPICMODELS

Treebanks

- Treebanks are often created on top of a corpus that has already been annotated with part-of-speech tags. In turn, treebanks are sometimes enhanced with semantic or other linguistic information.

terms_and_topics

perplexity

Posterior

IN LAYMANS TERMS

In case the discussion above was a little eye-glazing, here's another way to look at LDA in a different domain.

Suppose you've just moved to a new city. You're a hipster and an anime fan, so you want to know where the other hipsters and anime geeks tend to hang out. Of course, as a hipster, you know you can't just ask, so what do you do?

Here's the scenario: you scope out a bunch of different establishments (**documents**) across town, making note of the people (**words**) hanging out in each of them (e.g., Alice hangs out at the mall and at the park, Bob hangs out at the movie theater and the park, and so on). Crucially, you don't know the typical interest groups (**topics**) of each establishment, nor do you know the different interests of each person.

So you pick some number K of categories to learn (i.e., you want to learn the K most important kinds of categories people fall into), and start by making a guess as to why you see people where you do. For example, you initially guess that Alice is at the mall because people with interests in X like to hang out there; when you see her at the park, you guess it's because her friends with interests in Y like to hang out there; when you see Bob at the movie theater, you randomly guess it's because the Z people in this city really like to watch movies; and so on.

Of course, your random guesses are very likely to be incorrect (they're random guesses, after all!), so you want to improve on them. One way of doing so is to:

Pick a place and a person (e.g., Alice at the mall).

Why is Alice likely to be at the mall? Probably because other people at the mall with the same interests sent her a message telling her to come.

In other words, the more people with interests in X there are at the mall and the stronger Alice is associated with interest X (at all the other places she goes to), the more likely it is that Alice is at the mall because of interest X .

So make a new guess as to why Alice is at the mall, choosing an interest with some probability according to how likely you think it is.

Go through each place and person over and over again. Your guesses keep getting better and better (after all, if you notice that lots of geeks hang out at the bookstore, and you suspect that Alice is pretty geeky herself, then it's a good bet that Alice is at the bookstore because her geek friends told her to go there; and now that you have a better idea of why Alice is probably at the bookstore, you can use this knowledge in turn to improve your guesses as to why everyone else is where they are), and eventually you can stop updating. Then take a snapshot (or multiple snapshots) of your guesses, and use it to get all the information you want:

For each category, you can count the people assigned to that category to figure out what people have this particular interest. By looking at the people themselves, you can interpret the category as well (e.g., if category X contains lots of tall people wearing jerseys and carrying around basketballs, you might interpret X as the "basketball players" group).

For each place P and interest category C , you can compute the proportions of people at P because of C (under the current set of assignments), and these give you a representation of P . For example, you might learn that the people who hang out at Barnes & Noble consist of 10% hipsters, 50% anime fans, 10% jocks, and 30% college students.

KULLBACK–LEIBLER DIVERGENCE

In [probability theory](#) and [information theory](#), the **Kullback–Leibler divergence**^{[1][2][3]} (also **information divergence**, [information gain](#), **relative entropy**, or **KLIC**; here abbreviated as KL divergence) is a non-symmetric measure of the difference between two probability distributions P and Q . Specifically, the Kullback–Leibler divergence of Q from P , denoted $D_{\text{KL}}(P \parallel Q)$, is a measure of the information lost when Q is used to approximate P .^[4] The KL divergence measures the expected number of extra bits required to [code](#) samples from P when using a code based on Q , rather than using a code based on P . Typically P represents the "true" distribution of data, observations, or a precisely calculated theoretical distribution. The measure Q typically represents a theory, model, description, or approximation of P .

Although it is often intuited as a [metric or distance](#), the KL divergence is not a true [metric](#) — for example, it is not symmetric: the KL divergence from P to Q is generally not the same as that from Q to P . However, its infinitesimal form, specifically its [Hessian](#), is a [metric tensor](#): it is the [Fisher information metric](#).

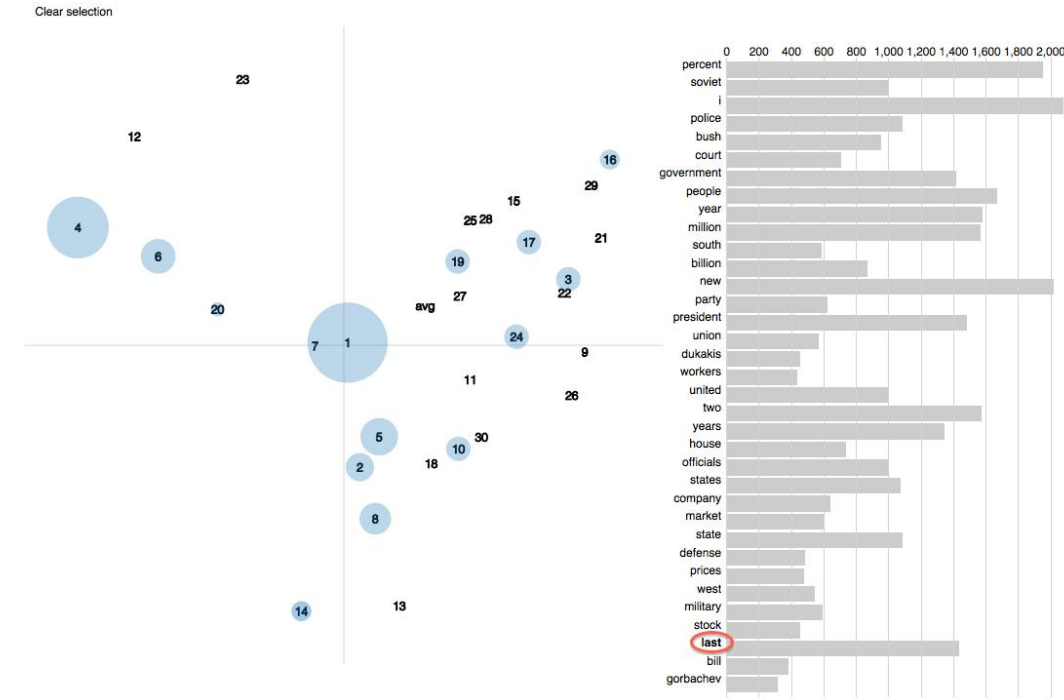
KL divergence is a special case of a broader class of [divergences](#) called [f-divergences](#). It was originally introduced by [Solomon Kullback](#) and [Richard Leibler](#) in 1951 as the **directed divergence** between two distributions. It can be derived from a [Bregman divergence](#).

SALIENCY

The other main component in the visualization is a bar chart with the most “important” keywords and their corresponding frequencies given the current state. By default, the most important keywords are defined by a measure of saliency ([Chuang, Heer, Manning 2012](#)).

Saliency is a compromise between a word's overall frequency and it's *distinctiveness*. A word's *distinctiveness* is a measure of that word's distribution over topics (relative to the marginal distribution over topics). A word is highly distinctive if that word's mass is concentrated on a small number of topics. Very obscure (or rare) words are usually highly distinctive which is why the overall frequency of the word is incorporated in the weight.

For example, on the y-axis of the bar chart on the Overview tab, we see the words “gorbachev” and “last”. Last is not very distinctive (it appears in many different topics), but it is still salient since it has such a high overall frequency. On the other hand, gorbachev does not have a very high overall frequency, but is highly distinctive since it appears almost solely in topic 17. The changing in the size of the circles below reflects the difference in distinctiveness between gorbachev and last.



CONDITIONING

Our definition of “important” words changes depending on whether the user indicates an interest in a particular cluster or topic. Here we define the *relevance* of a word given a topic as:

$$\lambda \log(p(\text{word} | \text{topic})) + (1 - \lambda) \log\left(\frac{p(\text{word} | \text{topic})}{p(\text{word})}\right)$$

The *relevance* of a word given a cluster can also be defined in a similar way:

$$\lambda \log(p(\text{word} | \text{topic})) + (1 - \lambda) \log\left(\frac{p(\text{word} | \text{cluster})}{p(\text{word})}\right)$$

where $p(\text{word} | \text{cluster}) = \sum_{\text{topic}} p(\text{word} | \text{topic})$

Once the top relevant words are chosen, these words are then ordered according to their frequency within the relevant topic/cluster. To keep these relevant words in place on the bar chart, the user must click the desired cluster region or topic circle. To resume to the default plots, click the “clear selection” text above the scatterplot. In the left hand screenshot below, cluster 2 is selected via click, then “soviet” is hovered upon to expose its conditional distribution over topics. Similarly, on the right hand screenshot, topic 17 is selected via click, then “union” is hovered upon.