Text Analysis in R

Processing unstructured data

Unstructured Data

In the past, we can only process numerical data (e.g., sales, profit, volume, quantity, etc.).

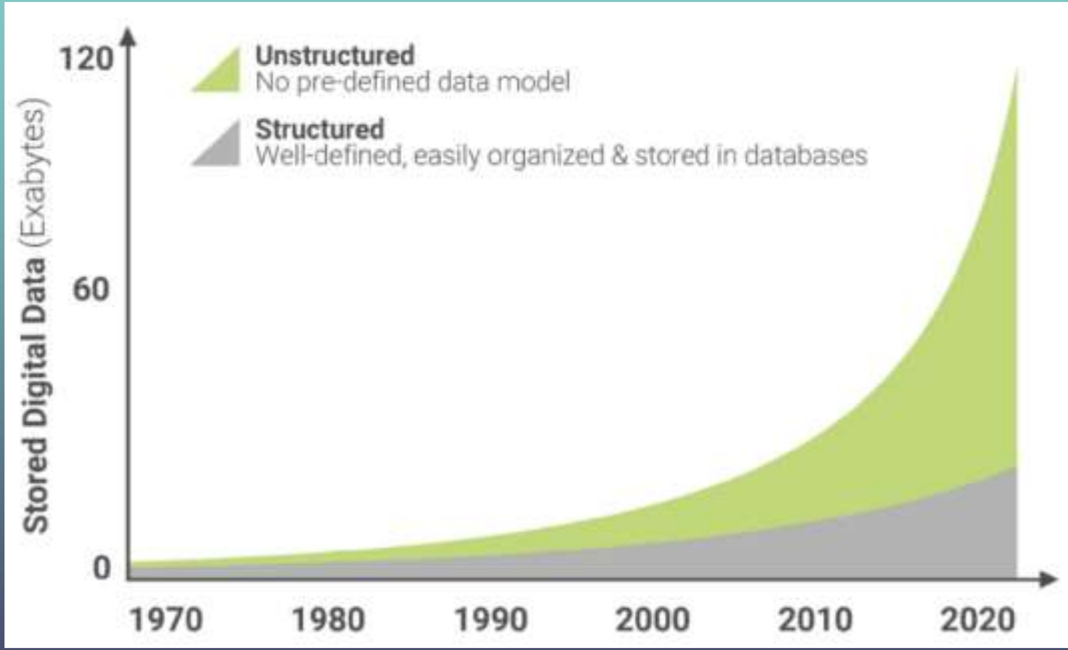
However, today, more and more data are unstructured.

Text, video, audio, images, etc.

To take advantage of unstructured data, we have to find a way to extract information from unstructured data.


What is
BIG DATA

Unstructured Data is Growing Exponentially…



- 1 Exabyte = 1,000,000,000,000,000,000 bytes

Text Data

Text data is one of the most commonly used types of unstructured data.

Text data is typically generated by users themselves.

Online reviews, movie critics, Tweets, SMS, WhatsAPP/WeChat/Facebook messages…

Text Data

Yet text data cannot be easily analyzed.

For example, how to run a regression with consumer reviews?

We need to extract meaningful measures from text!

Discussion: *Which measure can be extracted from text data?*

Analyzing Text with R

The simplest answer: The length of the text (e.g., number of letters/characteristics). It captures the amount of information in the text.

Let’s do this in R.

Analyzing Text with R

Here, we resort to the R package “stringi”.

```
install.packages("stringi")
library("stringi")
text = "What is the length of this sentence?"
print(stri_length(text))
```

What is your output? (It should be 36).

Analyzing Text with R

You can also work on Chinese:

```
text = "欢迎学习大数据"
print(stri_length(text))
```

Or emojis:

```
text = "@_@!!😀😃😄"
print(stri_length(text))
```

Analyzing Text with R

Now let us count the number of words in a text. Here, we assume that words are separated by spaces.

```
text = "Welcome to HKU!"
word_count = sapply(strsplit(text, " "), length)
print(word_count)
```

Note: in some cases, words are separated by other things such as a hyphen (e.g., “big-data”), in this case you need to write another code.

Analyzing Text with R

Counting sentences is a bit more difficult because you can have multiple stops in a single sentence. Instead of explaining the mechanism, you can just try the following codes which help you make the count:

```
text <- 'Hello world!! Here are two sentences for you...'
length(gregexpr('[[:alnum:] ][.!?]', text)[[1]])
```

Sentiment Analysis

Sentiment Analysis is arguably the most important type of text analysis. Basically, we want to classify text based on the *valence,* which can be either positive or negative (sometimes it can also be neutral).

Sentiment Analysis

Sentiment analysis can generate not only the valence of the text, but also the degree (e.g., strongly positive vs. slightly positive). Consider the following two sentences:

> Strongly positive: HKU is doing a great job in academic research.
>
> Slightly positive: HKU is doing well in academic research.

Sentiment Analysis Matters!

Today, many hedge funds collect data from social media (e.g., twitter, Facebook, 微博), detect individual sentiment toward a company, and infer its stock price accordingly.

Wisdom of Crowds: The Value of Stock Opinions Transmitted Through Social Media

Sentiment Analysis Matters!

Finextra

Quant trader turns to reddit for sentiment forecaster

New York-based quantitative hedge fund Cindicator Capital is advertising for an active member of the wallstreetbets subreddit community to ...

3 weeks ago

Business Wire

Join the Swarm of Retail Investors Driving Sentiment. New ...

An investment in VanEck Vectors® Social Sentiment ETF (BUZZ) may be ... participant concentration, new fund, absence of prior active market, ...

5 days ago

基于情感分析的交易策略：加密对冲基金如何利用AI实现绝对收益能力

It Integrates

Cognitive Science

Machine Learning

Natural Language Processing

Discussion

Question: In your own opinion, how should we do sentiment analysis?

Sentiment Analysis

The basic idea of sentiment analysis is rather simple.

We can build two lexicons (i.e., dictionaries) of positive and negative words. Here are some examples:

Positive: great, amazing, fantastic, excel, …

Negative: ugly, terrible, awful, failed, …

Click here to see examples of sentiment lexicons.

Sentiment Analysis

You can also do this for other languages, as long as you have a “sentiment lexicon”.

You can find one on our course website.

词语	极性
脏乱	2
糟报	2
早衰	2
责备	2
贼眼	2
战祸	2
招灾	2
折辱	2
中山狼	2
清峻	0
清莹	1
轻倩	1
晴丽	1
求索	1
热潮	1
仁政	1
荣名	1
柔腻	1
瑞雪	1

Sentiment Analysis

Naturally, if a sentence contains more positive words, it likely expresses some positive feeling.

Instead, a sentence containing many negative words are likely to express a negative emotion.

In addition, some words are “more positive” than others. For example, “great” and “awesome” are stronger than “OK” and “so so”. In this case, we can assign different weights to different words.

Sentiment Analysis

We can further go beyond sentiment analysis to achieve other purpose.

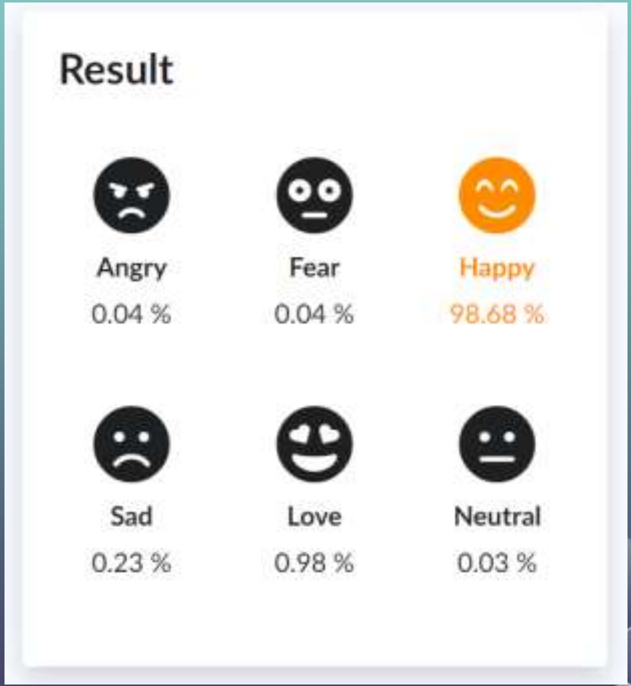
- What is the emotion of the text (e.g., sadness, happiness, excitement, joy, anger).
- Detect illegal content from text message.
- You can try the following (Chinese) one provided by JD.

Sentiment Analysis

We can further go beyond sentiment analysis to achieve other purpose.

What is the emotion of the text (e.g., sadness, happiness, excitement, joy, anger).

Here is a place where you can extract emotion from the text.



Sentiment Analysis

Let's try the following functions on R. "syuzhet" an algorithm that does sentiment analysis. It returns a positive (negative) value when the sentence is positive (negative).

```
library("syuzhet")
text = "HKU is a fantastic school, I love it."
syuzhet_vector <- get_sentiment(text,
method="syuzhet")
head(syuzhet_vector)
```

Sentiment Analysis

There also other algorithms such as “bing” and “afinn”. They use different scales, but the idea is similar. For details about these functions, please click here.

```
text = "HKU is a nice school and I like it."
bing_vector <- get_sentiment(text, method="bing")
head(bing_vector)
```

Sentiment Analysis

The function afinn is more of less the same, though the scale is larger:

```
text = "HKU is a nice school and I like it."
afinn_vector <- get_sentiment(text,
method="afinn")
head(afinn_vector)
```

Getting Emotions

The package also allows you to get emotions through the function nrc:

```
text = "HKU is a terrible school."
print(get_nrc_sentiment(text))
```

Exercise

Try the three algorithms yourself.

Do you find any issues with the sentiment analysis? What have you found? Discuss the issues with your classmates!

Issues

The algorithms are not great in processing negations. Here are some examples:

I don’t think HKU is a great school.
None of us like HKU.

In Chinese/Cantonese, we have “不”,”非”,”否” ”冇”,”唔”,”未”…

Solution

Dealing with negations is a complex issue, and here we only introduce the simplest methods. Let us take "not" as an example.

When we see "not", we take the opposite of every word after it until we reach the first punctuation. Here is one example:

I did not like this movie, but I enjoy this cinema…

Now, like -> not like, this-> not this, movie -> not movie…

But we keep "enjoy", "cinema" unchanged.

Other Tricks

We can assign lower weights to a sentence that ends with a question mark. For example, "Do you think HKU is a great school?" carries lower weight than "I think HKU is a good school".

We can assign higher weight to a sentence that uses capital letters. For example, "HKU is a GREAT school" carries higher weight than "HKU is a great school".

Other Tricks

If a sentence used a “booster word”, then we should assign higher/lower weights to the sentence.

Examples of booster words with a/an

Increased weight: completely, remarkably, tremendously, …

Decreased weight: marginally, occasionally, slightly, …

Other Tricks

We also look for “but” in a sentence. Words that appear before “but” deserve a lighter weight, and words that appear after “but” deserve a higher weight. For example

HKU has a lot of problems, but it is still my favorite school.

“problems” appear before “but”, so it receives a lower weight.
“favorite” appears after “but”, so it receives a higher weight.

Other Tricks

In addition to looking at positive and negative words, we should also pay attention to phrases. Here are some examples:

Installing Tableau Public is a piece of cake.

Buying the new iPhone costs me an arm and a leg.

我为了买iPhone去卖了肾

Other Tricks

In addition to looking at positive and negative words, we should also consider emojis:

I got an email from HKU admission office today 😛

I got an email from HKU admission office today 😭

Sentiment Analysis

Previously, we showed that when we only calculate sentiment based on positive/negative words, we are very likely to make mistakes. Here, we introduce a more accurate algorithm in R that alleviates these issues.

```
install.packages('sentimentr')
library("sentimentr")
text = 'I am good'
sentiment_by(text)
```

Sentiment Analysis

Compare the following pairs of sentences:

I like HKU. (0.288; positive)

I do not like HKU. (-0.223; negative)

Compare the following pairs of sentences:

I like HKU. (0.288; positive)

I really like HKU. (0.45; more positive)

Sentiment Analysis

Compare the following pairs of sentences:

HKU is my favorite school. (0.335; positive)

HKU is my FAVORITE school. (no adjustments made)

Compare the following pairs of sentences:

HKU is my favorite school. (0.335; positive)

Is HKU my favorite school? (no adjustments made)

More Challenges: Sarcasm

This phone has an awesome battery back-up of 38 hours.
This phone has an awesome battery back-up of 2 hours.

It's +25 outside and I am so hot.
It's -25 outside and I am so hot.

This is the best laptop bag ever. It is so good that within two months of use, it is worthy of being used as a grocery bag.

More Challenges: Comparison

This product is second to none.

This one is better than the old one.

This one is better than nothing.

More Challenges: Ambiguity

In English, a word can have different meanings in different contexts. Here are some examples:

> Low cost is positive whereas low salary is negative.
> Limited edition is positive while limited food is negative.
> "The story is unpredictable." -- It is hard to say whether this is a positive or negative description.

Objective Sentences

I followed his recommendation and bought the stock at $200. Now the stock price is $100.

There are neither positive words nor negative words in the sentence, yet the sentence is negative.

He started learning R last year. Today, he does not know how to run a linear regression.

Conditional Sentences

If I see a great smelling cake in the supermarket, I will buy it home.

There is a positive word “great” in the sentence, but this sentence does not contain any sentiment itself.

If you are looking for an awesome MSc program, join HKU.

Still, this is a conditional sentence, but it has its own sentiment.

Aspirations

I am dying to see the new movie "Matrix 4".

Even though you have the word "dying", this sentence is clearly positive.

I really want to join HKU.

There are no positive words but the sentence is positive.

Intensions

I am going to throw my Lenovo laptop out of the window.

Clearly negative.

I am returning this table to IKEA tomorrow.

Again, this is negative.

Indirect Opinions

With R, I can complete all data analysis in 1 hour that used to take me 3 hours in the past.

After getting the new lenses, I am able to drive at night again.

More Challenges

For some issues, we can develop better lexicons (i.e., dictionaries) to detect the sentiment of specific words (e.g., we can label “low cost” as positive but “low salary” as negative).

But this does not completely solve the issue. For example, it would be rather difficult to understand sarcasm using the lexicon approach.

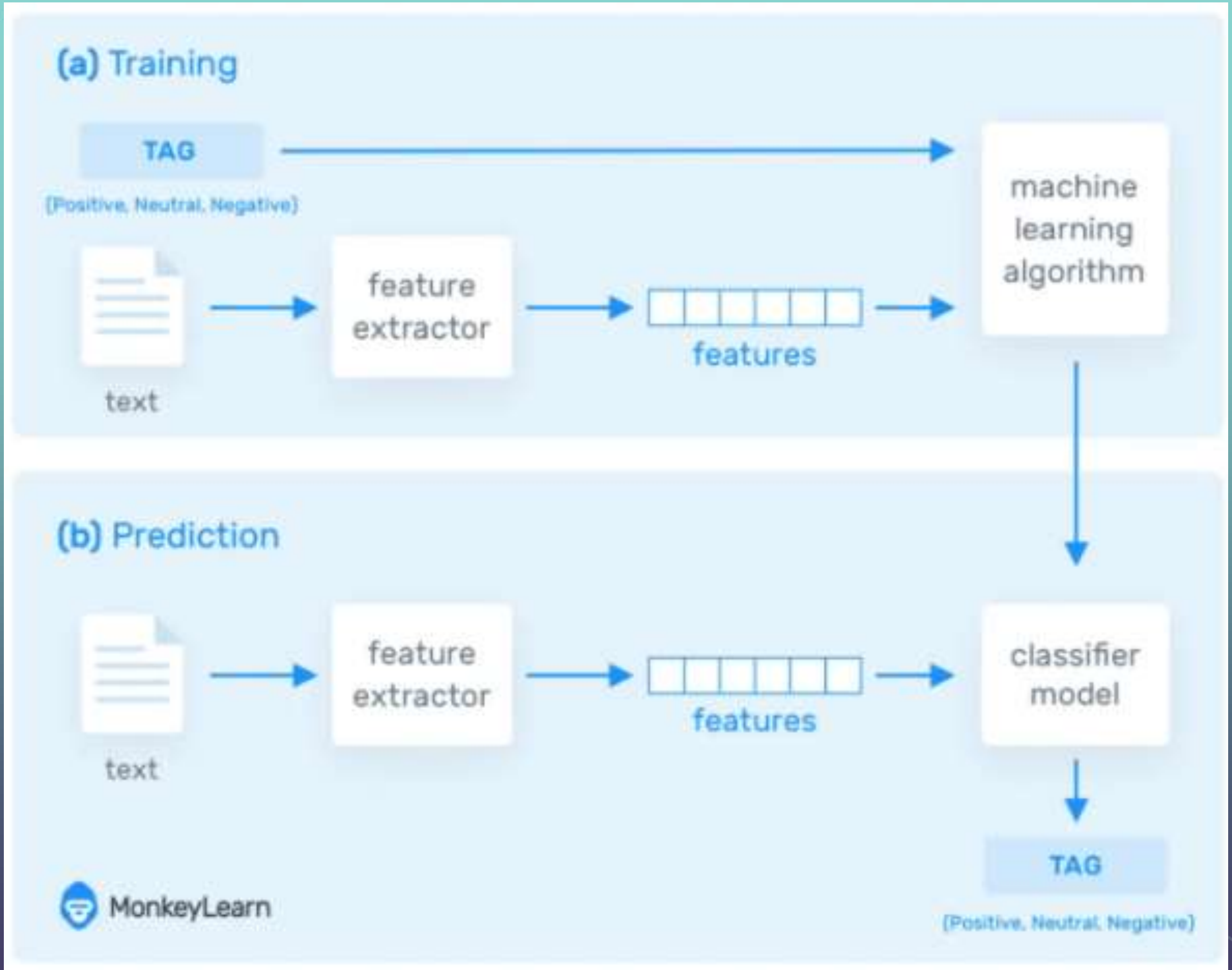
Let’s watch a video…

Two Approaches of Sentiment Analysis

Sentiment Analysis

As illustrated in the video, an alternative approach to sentiment analysis is to use machine learning methods.

In general, machine-learning based sentiment analysis has two steps: training and prediction.



(a) Training
TAG
(Positive, Neutral, Negative)
text
feature extractor
features
machine learning algorithm
(b) Prediction
text
feature extractor
features
classifier model
TAG
(Positive, Neutral, Negative)
MonkeyLearn

Sentiment Analysis

Training data: We need to first use humans to code text data. For example, we rate "My salary is very low" as a "negative" sentence.

Here, you can think "My salary is very low" as your independent variable (X) and "negative" as your dependent variable (Y).

In training, we look for a function $f(\cdot)$ such that $f(X) \approx Y$.

Sentiment Analysis

Basically, we can turn each word into a number, so that we can turn a sentence into a number (X).

For the output, you can let $Y \in [-1, 1]$, and when it is positive (negative), it represents a positive (negative) sentiment.

What about the function $f: X \to Y$? How does this function look like?

Sentiment Analysis

In machine-learning based sentiment analysis, several algorithms are adopted to match your X with Y. These algorithms are:

Naïve Bayesian, Support Vector Machine, Deep Learning, …

They are just like linear regression, though more complex that it.

Beyond Sentiment Analysis

With machine learning methods, we can achieve a lot more than simple sentiment analysis. For example, we can "calculate" a couplet (in Chinese: 对联) based on your input. Let's play a small game here.

Topic Models

A story has its own topics, a novel has its own topics, and a film also has its own topic. Similarly, when we have text documents, they must also contain a few topics.

In topic modeling, we are concerned with the fundamental question: “What are the topics that a document is about?”

Do we need topic modeling?

In today's world, there are huge volumes of text information generated everyday. Of course, we don't have time to read them, and we often lack the expertise to understand them.

But we often need to know what they are talking about. For example, Tim Cook, Apple's CEO, may be concerned about how consumers think about Apple.

Motivating Examples

What are the topics that a document is about?
Given one document, can we find other documents about the same topics?
How do topics in a field change over time?

Discussion

In your opinion, what is a topic?

Discussion

In your opinion, what is a topic?

A topic is a distribution over words. For example, when you mention the following words frequently, you are likely to talk about the topic "marketing": advertising, distribution channel, pricing, product, sales, market research, …

Discussion

In your opinion, what is a topic?

A topic is a distribution over words. For example, when you mention the following words frequently, you are likely to talk about the topic “flight”: airport, ticket, delay, shuttle bus, check-in, business-class …

Discussion

How to find out topics from a number of documents?

Discussion

How to find out topics from a number of documents?

This is a complex process. The basic idea is, words of the same topic would often appear together. For example, “airport” and “shuttle bus” often appear in the same sentence because they belong to the same topic; “university” and “education” often appear in the same sentence as well. However, “airport” and “education” do not appear frequently in the same sentence.

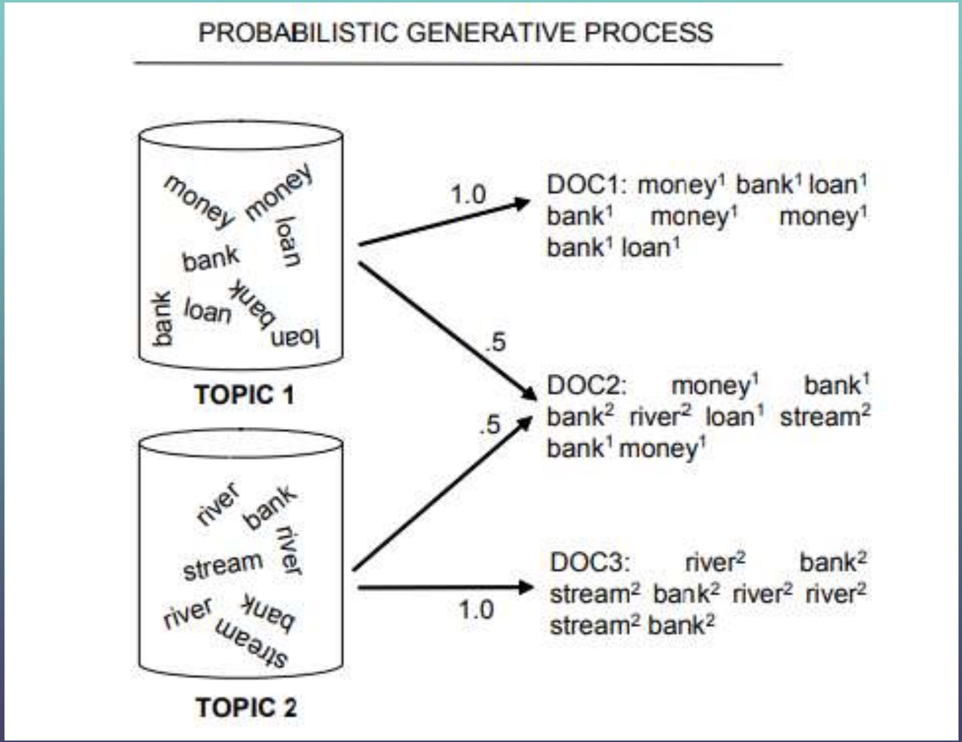
Example

Excellent company to deal with. I use this company all the time. The only problem I had is they say no money is taken from your account until the order has been fully processed. I have come across and order that has been cancelled then have to wait for the money to be put back into my account because...

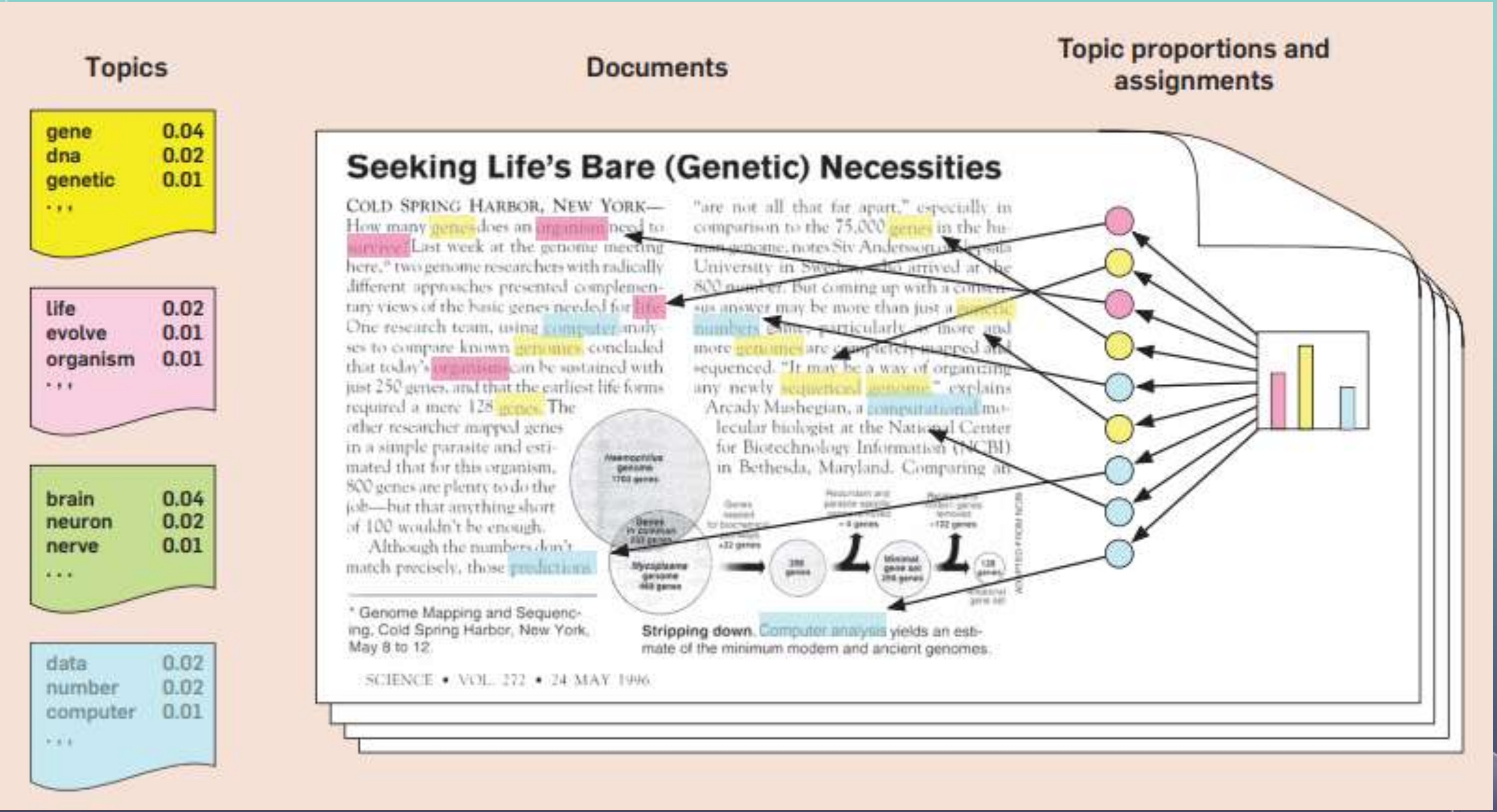
Example

Excellent company to deal with. I use this company all the time. The only problem I had is they say no money is taken from your account until the order has been fully processed. I have come across and order that has been cancelled then have to wait for the money to be put back into my account because...

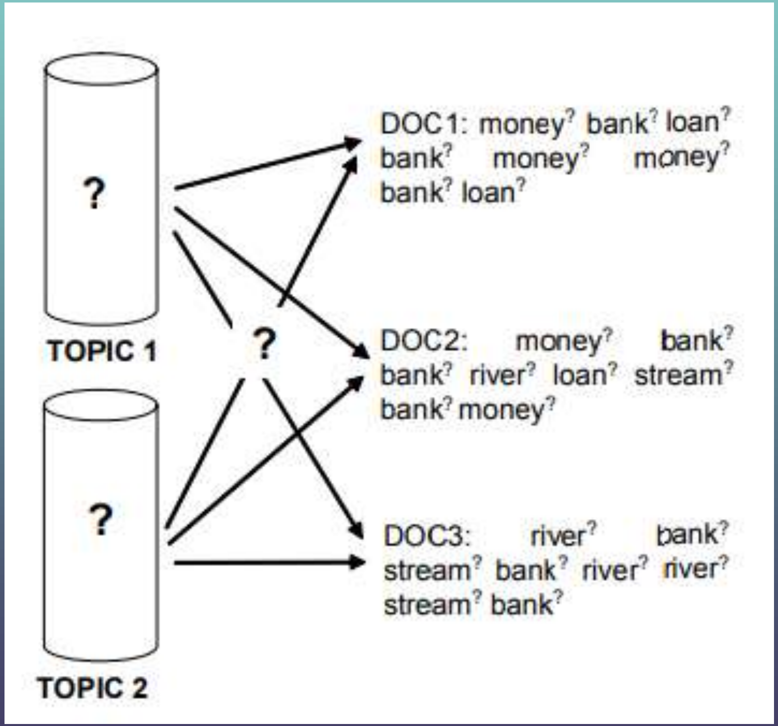
The Big Picture



Each document can be generated from multiple topics (e.g., half topic 1 and half topic 2).


Topics
Documents
Topic proportions and assignments
gene 0.04
dna 0.02
genetic 0.01
...
life 0.02
evolve 0.01
organism 0.01
...
brain 0.04
neuron 0.02
nerve 0.01
...
data 0.02
number 0.02
computer 0.01
...
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

The Big Picture



However, what we want do is not to generate documents using the topics, but to infer topics from the documents.

The Big Picture

Theoretically, you can try different topics and see which topics can likely generate the documents you have at hand.

> Of course, this is an impossible task because there are infinitely many topics that can be used.

Computer scientists use statistical methods to infer the topics from the document. This is very complex; the methods include Gibbs sampling, variational inference, …

> You need a stat degree to understand this; so, we will not cover it.

If you are interested…

If you are interested in topic modeling, please check the original publication “Latent Dirichlet allocation (LDA)” here. If you can understand it, you could be a great data scientist.

Topic Modelling

Demonstration of LDA

Please visit here for an online demonstration of LDA.

https://mimno.infosci.cornell.edu/jsLDA/jslda.html

The source files are available on the course website.

An LDA Example of Chinese Medical Text

Topic 1	Topic 2	Topic 3	Topic 4	Topic 5
消化	风湿	血糖	眼球	胰腺
胃炎	关节	糖尿病	眼睑	肝癌
幽门	激素	空腹	结膜炎	腹部
食管	抗体	胰岛素	角膜	多发
名称	名称	肾上腺	睫毛	腹水
…	…	…	…	…
Topic 6	**Topic 7**	**Topic 8**	**Topic 9**	**Topic 10**
云芝糖肽胶囊	血管瘤	白内障	输尿管	输尿管
右旋布洛芬栓	痣	模糊	肾积水	膀胱
左克	皮肤	青光眼	睾丸	肾结石
腔隙	红斑	视网膜	左侧	双肾
尿急	素	名称	扩张	血尿
…	…	…	…	…
Topic 11	**Topic 12**	**Topic 13**	**Topic 14**	**Topic 15**
皮肤	月经	黄疸	消化不良	胆碱
头发	怀孕	胆红素	下舌段	利福喷汀胶囊
湿疹	卵泡	茵栀黄	前臂骨折	肝脾康
脱发	输卵管	肺炎	直头	心静脉
皮炎	卵巢	胆汁淤积	尼麦角林片	胃好
…	…	…	…	…
Topic 16	**Topic 17**	**Topic 18**	**Topic 19**	**Topic 20**
鼻腔恶性淋巴瘤	脑神经	胆囊	脾恶性肿瘤	胎儿
抗药性	硝苯地平缓释片	胆结石	急性炎症	怀孕
中央部	泰诺	形态	清热解毒胶囊	羊水
硼酸醇	自身抗体	扩张	合酶	胎盘
肠溶胶囊	右缘	胆囊炎	健脾丸	流产
…	…	…	…	…
Topic 21	**Topic 22**	**Topic 23**	**Topic 24**	**Topic 25**
胆红素	肌酐	心电图	超声	先舒
抗体	肾病	胸闷	坐月子	止血
肝病	激素	高血压	善存	间变
抗病毒	肾炎	心率	腰椎骨折	磷霉素
肝硬化	血尿	名称	胃好	后角
…	…	…	…	…
Topic 26	**Topic 27**	**Topic 28**	**Topic 29**	**Topic 30**
失眠	头孢地尼	纵隔囊肿	甲亢	肥胖
焦虑	中耳炎	海藻	甲状腺	无血管
抑郁症	盐酸左氧氟沙星胶囊	脊柱后凸	优甲乐	测试
入睡	骨化	体重减轻	怀孕	磷酸肌酸
名称	呼吸困难	骨髓炎	抗体	善宁
…	…	…	…	…

Beyond Text Information

Image recognition allows us to identify objects from photos. This is a machine-learning based approach.

Question

How can market participants (firms, consumers, government, platforms) benefit from image recognition technologies?

Beyond Text Information

Vocal analysis allows us to identify emotions from audio pitches. How can we benefit from this technology?

Beyond Text Information

The Power of Voice: Managerial Affective States and Future Firm Performance

WILLIAM J. MAYEW and MOHAN VENKATACHALAM*

Beyond Text Information