BUSN 20800: Big Data

Lecture 8: Text Data

Dacheng Xiu

University of Chicago Booth School of Business

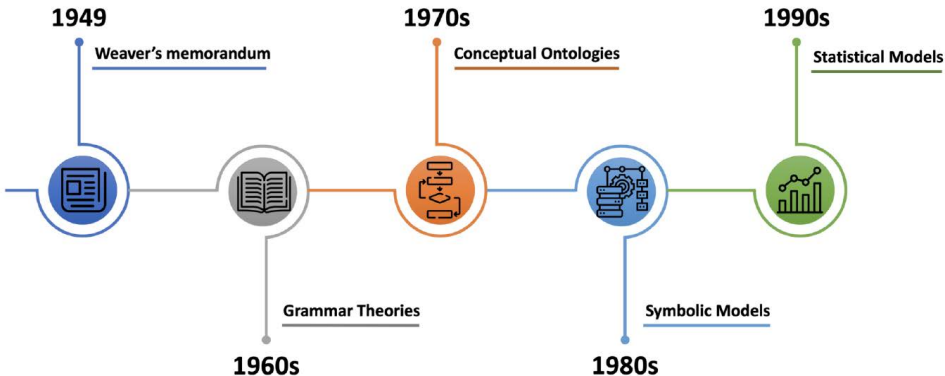
Outline

- Introduction
- Classification: Sentiment Analysis
- Clustering: Topic Modeling
- Word Embedding

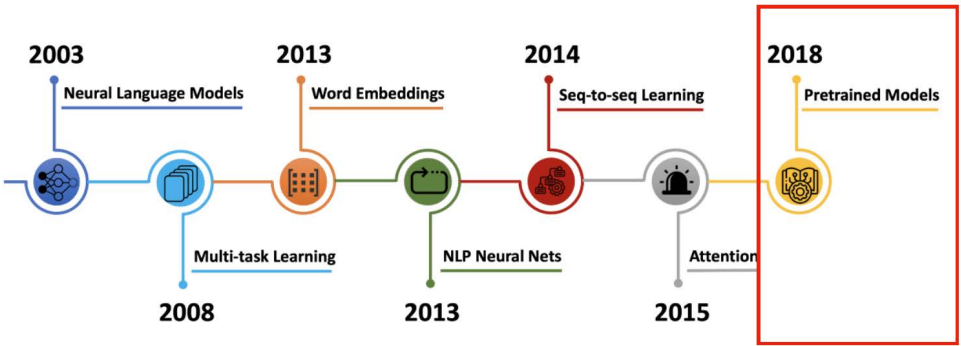
What is NLP about?

- Natural language processing (NLP, aka computational linguistics) is a branch of artificial intelligence that helps computers understand, interpret and manipulate human language.
- NLP is an extremely rapidly-evolving field
  - a vast increase in computing power
  - the availability of very large amounts of linguistic data
  - the development of highly successful machine learning models
  - a much richer understanding of the structure of human language and its deployment in social contexts

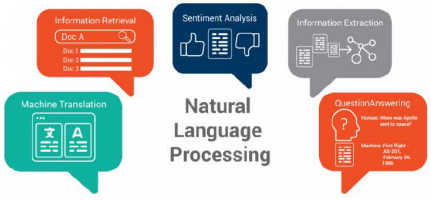
A brief history of NLP

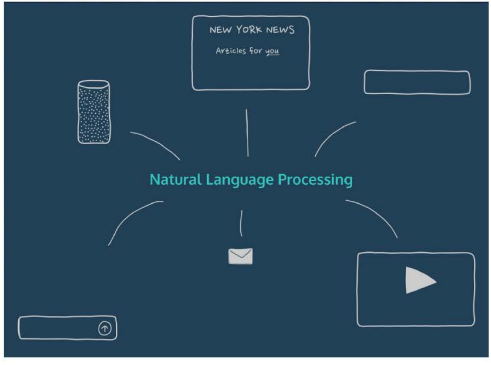


A brief history of NLP



What are NLP tasks?





What are NLP tasks?

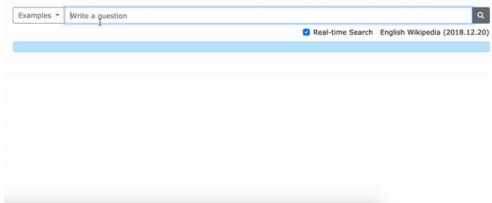
Question answering

The first recorded travels by Europeans to China and back date from this time. The most famous traveler of the period was the Venetian Marco Polo, whose account of his trip to "Cambaluc," the capital of the Great Khan, and of life there astounded the people of Europe. The account of his travels, Il milione (or, The Million, known in English as the Travels of Marco Polo), appeared about the year 1299. Some argue over the accuracy of Marco Polo's accounts due to the lack of mentioning the Great Wall of China, tea houses, which would have been a prominent sight since Europeans had yet to adopt a tea culture, as well the practice of foot binding by the women in capital of the Great Khan. Some suggest that Marco Polo acquired much of his knowledge through contact with Persian traders since many of the places he named were in Persian.

How did some suspect that Polo learned about China instead of by actually visiting it?

Answer: through contact with Persian traders

(reading comprehension)



(open-domain QA)

What are NLP tasks?

Another popular pre-trained model

Text summarization

Source Document (abbreviated)	BART Summary
The researchers examined three types of coral in reefs off the coast of Fiji ... The researchers found when fish were plentiful, they would eat algae and seaweed off the corals, which appeared to leave them more resistant to the bacterium Vibrio coralliilyticus, a bacterium associated with bleaching. The researchers suggested the algae, like warming temperatures, might render the corals' chemical defenses less effective, and the fish were protecting the coral by removing the algae.	Fisheries off the coast of Fiji are protecting coral reefs from the effects of global warming, according to a study in the journal Science.
Sacoolas, who has immunity as a diplomat's wife, was involved in a traffic collision ... Prime Minister Johnson was questioned about the case while speaking to the press at a hospital in Watford. He said, "I hope that Anne Sacoolas will come back ... if we can't resolve it then of course I will be raising it myself personally with the White House."	Boris Johnson has said he will raise the issue of US diplomat Anne Sacoolas' diplomatic immunity with the White House.
According to Syrian state media, government forces began deploying into previously SDF controlled territory yesterday. ... On October 6, US President Donald Trump and Turkish President Recep Tayyip Erdoan spoke on the phone. Then both nations issued statements speaking of an imminent incursion into northeast Syria ... . On Wednesday, Turkey began a military offensive with airstrikes followed by a ground invasion.	Syrian government forces have entered territory held by the US-backed Syrian Democratic Forces (SDF) in response to Turkey's incursion into the region.

What are NLP tasks?

Machine translation

SOURCE

作为一名艺术家，联系对我来说是非常重要的。通过我的艺术作品我试着阐明人类不是与自然分隔开而是每一件事都是互相联系的。大约10年前我第一次去了南极洲，我也第一次看到了冰山。我感到敬畏。我的心快速地呼动，头晕目眩，试着理解在我面前的这到底是什么。在我身边的冰山浮出水面几乎200英尺。我只能感到很奇怪这就是一片雪花覆盖在另一片雪花，年复一年形成的。冰山的形成是当它们从冰川断裂开或者从冰架上断裂开。每个冰山都有它们自己的独特个性。它们与其周边的环境和其情况的互动具有一个鲜明的方式。有些冰山拒绝妥协坚持到底，而另一些冰山就不能忍受在一时剧烈激情喷涌下就冰崩冰裂。当你看到冰山，很容易就想到它们都是孤立的，它们是独立的，单独一体的，更像是我们人类有时候对自身的看法。但现实远不止这个。随着冰山融化，我呼吸到它古老的气味。随着冰山融化，它释放了富有矿物质的鲜水它们滋养了万物。我着手拍摄这些冰山好似我在拍摄我祖先的肖像，了解到在这些个别的时刻冰山原是以那样方式存在但再也不会像那样存在了。当它们融化时，这绝不是死亡；也绝不是一个终结，而是通往生生不息之路的一个延续。我拍摄过的冰山，有些从是非常年轻-- 几千年年龄。有些冰超过十万年。我想给大家展示的最后图片是我在格陵兰岛的 Kekertsuatsiak上拍摄的一个冰山。这是一个非常难得的机会大家实际上得以见证一个冰山翻滚。所以这就如图所示。在左边你能看到一个小船。这是一个约15英尺的船。我想让你注意冰山的形状它在水面上的变形。在这儿你看到它开始翻滚，小船移动到另一边，一个男人站在那里。这是一个平均尺寸的格陵兰冰山。它浮出水面大约有120英尺高或者40米高。这视频是实时拍摄的。就像这冰山，它们展示给大家的是其个性的不同方面。谢谢。

mBART25 SENT-MT

As an artist, connection is very important to me. And through my artwork, I try to illustrate that humans are not separate from nature, but that everything is connected to each other. I went to Antarctica for the first time about 10 years ago, and I saw the iceberg for the first time. I felt awe. And my heart was pounding, my head was pounding, trying to figure out what was going on in front of me. And the iceberg that was right next to me was floating almost 200 feet above the surface of the water, and I just couldn't help but feel that this was a snowflake covering another snowflake, forming over and over and over again. The formation of glaciers is when they break off from glaciers, or break off from ice shelves. Every iceberg has its own unique personality. They interact in a very distinctive way with the environment around them and with the circumstances around them. Some glaciers refused to compromise and insisted, and some glaciers couldn't stand the heat of passion as it poured down the ice. When you look at icebergs, it's easy to think that they're isolated, that they're independent, that they're separate, that they're more like the way we think about ourselves sometimes. But the reality is much more than that. And as the glacier melts, I breathe in its ancient smell. And as the glacier melts, it releases fresh water of minerals that nourish everything. I started photographing these icebergs like I was photographing my ancestors, and I learned that in these individual moments, the icebergs existed in that way, but they never existed like that again. When they melt, it's not about death; it's not about the end, it's about the continuation of a life-long path. I photographed glaciers, and some of them were very young -- thousands of years old. Some of the ice has been there for more than 100,000 years. And the last picture I want to show you is an iceberg that I photographed in Kekertsuatsiak on the island of Greenland. It's a very difficult opportunity to actually witness the rolling of an iceberg. So this is what it looks like. You can see a little boat on the left. This is a 15-foot boat. I want you to notice that the shape of the iceberg changes as it moves over the surface. And here you see it rolling, and the boat moves to the other side, and a man is standing there. This is an average size glacier in Greenland. It floats about 120 feet up or 40 meters above the surface. This video was taken in real time. And like this iceberg, they show you different aspects of their personality. Thank you.

What are NLP tasks?

Sentiment analysis

Loves the German bakeries in Sydney. Together with my imported honey it feels like home	Positive
@VivaLaLauren Mine is broken too! I miss my sidekick	Negative
Finished fixing my twitter...I had to unfollow and follow everyone again	Negative
@DinahLady I too, liked the movie! I want to buy the DVD when it comes out	Positive
@frugaldougal So sad to hear about @OscarTheCat	Negative
@Mofette briliant! May the fourth be with you #starwarsday #starwars	Positive
Good morning thespians a bright and sunny day in UK, Spring at last	Positive
@DowneyisDOWNEY Me neither! My laptop's new, has dvd burning/ripping software but I just can't copy the files somehow!	Negative

What are NLP tasks?

Natural language inference

Text	Judgments	Hypothesis
A man inspects the uniform of a figure in some East Asian country.	contradiction C C C C C	The man is sleeping
An older and younger man smiling.	neutral N N E N N	Two men are smiling and laughing at the cats playing on the floor.
A black race car starts up in front of a crowd of people.	contradiction C C C C C	A man is driving down a lonely road.
A soccer game with multiple males playing.	entailment E E E E E	Some men are playing a sport.
A smiling costumed woman is holding an umbrella.	neutral N N E C N	A happy woman in a fairy costume holds an umbrella.

What are NLP tasks?

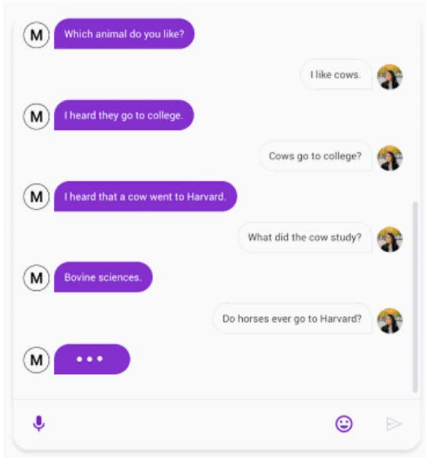
Fact checking

Claim: Claire Danes is wedded to an actor from England.
[wiki/Claire_Danes] She is married to actor Hugh Dancy, with whom she has one child.
[wiki/Hugh_Dancy] Hugh Michael Horace Dancy (born 19 June 1975) is an English actor and model.
Verdict: SUPPORTED

Claim: Rogue appears in Canadian comic books.
[wiki/Rogue_(comics)] Rogue is a fictional superhero appearing in American comic books published by Marvel Comics, commonly in association with the X Men .
Verdict: REFUTED

What are NLP tasks?

Dialogue



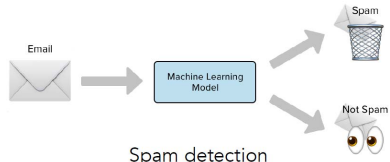
Text Classification

Applications

- Authorship attribution
- Language detection
- News categorization
- . . . . and many more!

Text classification

- Inputs:
  - A document d
  - A set of classes C=$\{c_1, c_2, ..., c_m\}$
- Output:
  - Predicted class c for document d



Spam detection

Sentiment analysis

Rule-based classification

IF there exists word w in document d such that w in [good, great, extra-ordinary,...]

- THEN output Positive (e.g., VADER)

IF email address ends in [ithelpdesk.com, makemoney.com, spinthewheel.com, . . .]

- THEN output SPAM

Pros & cons:

- + Can be very accurate
- - Rules may be hard to define (and some even unknown to us!)
- - Expensive
- - Not easily generalizable

Supervised Learning: Let's use statistics!

Let the machine figure out the best patterns using data

Inputs:

- Set of m classes C=$\{ c_1, c_2, ..., c_m \}$
- Set of n 'labeled' documents: $(d_1, c_1), (d_2, c_2), ..., (d_n, c_n)$

Outputs:

- Trained classifier, $F : d \rightarrow c$

Key questions:

- What is the form of F?
- How do we learn F?

Supervised classifiers

We have seen some of them! e.g. Logistic regression, k-nearest neighbors, Neural networks, etc.

Simple classification model making use of Bayes rule

$$P(c|d) = \frac{P(c)P(d|c)}{P(d)}$$

(d: document, c: class)

Best class: maximum a posteriori (MAP) estimate

$$c_{MAP} = \arg\max_c P(c|d) = arg\max_c P(c)P(d|c)$$

How to represent $P(d|c)$?

Option 1: represent the entire sequence of words

- $P(w_1, w_2, ..., w_K|c)$
- typical English vocabulary $\sim$ 40K words. too many sequences!

Option 2: Bag of words

- Assume position of each word is irrelevant (both absolute and relative)
- $P(w_1, w_2, ..., w_K|c) = P(w_1|c)P(w_2|c)...P(w_k|c)$
- Probability of each word is conditionally independent of the other words given class c

Bag of words (BoW)

I love this movie! It's sweet, but with satirical humor. The dialogue is great and the adventure scenes are fun... It manages to be whimsical and romantic while laughing at the conventions of the fairy tale genre. I would recommend it to just about anyone. I've seen it several times, and I'm always happy to see it again whenever I have a friend who hasn't seen it yet!

it	6
I	5
the	4
to	3
and	3
seen	2
yet	1
would	1
whimsical	1
times	1
sweet	1
satirical	1
adventure	1
genre	1
fairy	1
humor	1
have	1
great	1
…	…

Predicting with Naive Bayes

Overall process:

Input: Set of annotated documents $\{(d_i, c_i)\}_{i=1}^{n}$

- Compute vocabulary V of all words
- Calculate $\hat{P}(c_j) = \frac{count(c_j)}{n}$
- Calculate $\hat{p}(w_i|c_j) = \frac{count(w_i,c_j)+a}{\sum_w [count(w,c_j)+a]}$
- Prediction, given document $d = (w_1, w_2, ..., w_k)$

$$c_{MAP} = argmax \hat{P}(c) \prod_{i=1}^{K} \hat{P}(w_i|c)$$

Naive Bayes as a language model

Which class assigns the higher probability to s?

Sentence s: I love this fun film

Positive:

0.1, 0.1, 0.01, 0.05, 0.1

Negative:

0.2, 0.001, 0.01, 0.005, 0.1

Model pos		Model neg	
0.1	I	0.2	I
0.1	love	0.001	love
0.01	this	0.01	this
0.05	fun	0.005	fun
0.1	film	0.1	film

Naive Bayes as a language model

Which class assigns the higher probability to s?

Sentence s: I love this fun film

Positive:

0.1, 0.1, 0.01, 0.05, 0.1

Negative:

0.2, 0.001, 0.01, 0.005, 0.1

$$P(s|pos) > P(s|neg)$$

Model pos	
0.1	I
0.1	love
0.01	this
0.05	fun
0.1	film

Model neg	
0.2	I
0.001	love
0.01	this
0.005	fun
0.1	film

Advantages of Naive Bayes

- Very fast, low storage requirements
- Robust to irrelevant features
  - Irrelevant features cancel each other without affecting results
- Very good in domains with many equally important features
  - Decision trees suffer from fragmentation in such cases — especially if little data
- Optimal if the independence assumptions hold
  - If assumed independence is correct, this is the 'Bayes optimal' classifier
- A good dependable baseline for text classification
  - However, other classifiers can give better accuracy

Sentiment Analysis

- Sentiment analysis is the detection of attitudes: Is the attitude of this text positive or negative?
- Rule-based: VADER (Valence Aware Dictionary for Sentiment Reasoning)

```
import nltk
nltk.download('vader_lexicon')
from nltk.sentiment.vader import SentimentIntensityAnalyzer
sid = SentimentIntensityAnalyzer()
```

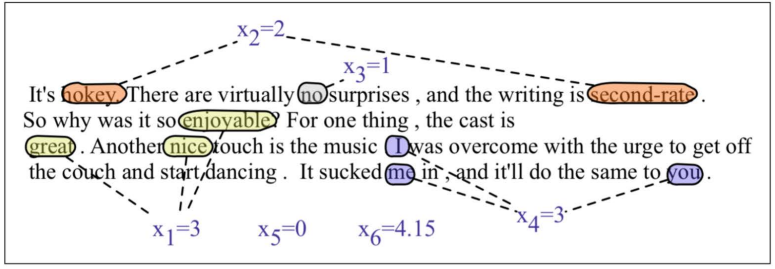
VADER's SentimentIntensityAnalyzer() takes in a string and returns a dictionary of scores in each of four categories: negative, neutral, positive, compound.

Sentiment Classification

- Given that sentiment analysis is a classification problem, we can directly import tools learned earlier!
- Logistic regression:

$$P(Y = 1) = \sigma(wx + b).$$

Example: Sentiment classification



Var	Definition	Value
x_1	count(positive lexicon) $\in$ doc)	3
x_2	count(negative lexicon) $\in$ doc)	2
x_3	$\begin{cases} 1 & \text{if "no"} \in \text{doc} \\ 0 & \text{otherwise} \end{cases}$	1
x_4	count(1st and 2nd pronouns $\in$ doc)	3
x_5	$\begin{cases} 1 & \text{if "!"} \in \text{doc} \\ 0 & \text{otherwise} \end{cases}$	0
x_6	log(word count of doc)	$\ln(64) = 4.15$

Example: Sentiment classification

Var	Definition	Value
x_1	count(positive lexicon) $\in$ doc)	3
x_2	count(negative lexicon) $\in$ doc)	2
x_3	$\begin{cases} 1 & \text{if "no"} \in \text{doc} \\ 0 & \text{otherwise} \end{cases}$	1
x_4	count(1st and 2nd pronouns $\in$ doc)	3
x_5	$\begin{cases} 1 & \text{if "!"} \in \text{doc} \\ 0 & \text{otherwise} \end{cases}$	0
x_6	log(word count of doc)	$\ln(64) = 4.15$

Assume weights $w = [2.5, -5.0, -1.2, 0.5, 2.0, 0.7]$ and bias $b = 0.1$

$$
\begin{aligned}
p(+|x) = p(Y = 1|x) &= \sigma(wx + b) \\
&= \sigma([2.5, -5.0, -1.2, 0.5, 2.0, 0.7])([3, 2, 1, 3, 0, 4.15] + 0.1) \\
&= \sigma(0.805) \\
&= 0.69 \\
p(-|x) = p(Y = 0|x) &= 1 - \sigma(wx + b) \\
&= 0.31
\end{aligned}
$$

Designing features

- Most important rule: Data is key!
- Linguistic intuition (e.g. part of speech tags, parse trees)
- Feature templates
- Complex combinations

$$x_1 = \begin{cases} 1 & \text{if } \text{“}Case(w_i) = \text{Lower”} \\ 0 & \text{otherwise} \end{cases}$$

$$x_2 = \begin{cases} 1 & \text{if } \text{“}w_i \in \text{AcronymDict”} \\ 0 & \text{otherwise} \end{cases}$$

$$x_3 = \begin{cases} 1 & \text{if } \text{“}w_i = \text{St. \& } Case(w_{i-1}) = \text{Cap”} \\ 0 & \text{otherwise} \end{cases}$$

Advanced: Representation learning

Logistic Regression: what's good and what's not

Pros:

- More freedom in designing features
- No strong independence assumptions like Naive Bayes
- More robust to correlated features ("San Francisco" vs "Boston")
  - —LR is likely to work better than NB
- Can even have the same feature twice! (why?)

Cons:

- May not work well on small datasets (compared to Naive Bayes)
- Interpreting learned weights can be challenging

Textural analysis in finance

Rule-based:

- Tetlock (2007): Harvard-IV psychosocial dictionary
- Loughran and McDonald (2007): Create a new dictionary for finance context

Classification based:

- Cf. "Predicting Returns with Text Data," Ke, Kelly, and Xiu (2022) on SSRN
- In stock market prediction, it is natural to use the sign of stock returns as labels.

The IBM Example: Raw Article

IBM Profit Falls as Revenue Declines – 4th Update By Robert McMillan

International Business Machines Corp. is trying to reinvent itself as a modern technology innovator, but it is proving to be a tough act for the century-old company.

On Monday, IBM reported second-quarter revenue fell 13.5%, adding to a string of quarterly declines that now spans 13 periods despite scaling back on legacy hardware and pushing into cloud-based software and services.

IBM remains under assault from computing in the cloud, which threatens to undermine its hardware and infrastructure businesses and erode profit margins in the computing business. To win this fight, the company trimmed itself over the past year, exiting unprofitable server and chip-making businesses to focus instead on data analytics and security software as well as cloud and mobile computing products. ... IBM says that these newer businesses are growing, but the company reported a year-over-year decline in all of its major lines. Technology services revenue was down 10%; business services fell 12%; software dropped 10%; and overall hardware revenue sank 32%. IBM profit dipped 16.6% to $3.45 billion, weighed down by acquisition-related charges.

Tess Stynes contributed to this article.
Write to Tess Stynes at tess.stynes@wsj.com and Robert McMillan at Robert.Mcmillan@wsj.com
Access Investor Kit for International Business Machines Corp.
Visit `http://www.companyspotlight.com/partner?cp_code=P479&isin=US4592001014`.
July 20, 2015 19:06 ET (23:06 GMT)

How Machine Reads the IBM News ...

'ibm', 'profit', 'fall', 'revenue', 'decline', 'update'

'by', 'try', 'reinvent', 'technology', 'innovator', 'prove', 'tough', 'act', 'century', 'old', 'company', 'on', 'reported', 'second', 'quarter', 'revenue', 'fall', 'add', 'string', 'quarterly', 'decline', 'span', 'period', 'despite', 'scale', 'back', 'legacy', 'push', 'cloud', 'base', 'software', 'service', 'remain', 'assault', 'compute', 'cloud', 'threatens', 'undermine', 'infrastructure', 'business', 'erode', 'profit', 'margin', 'compute', 'business', 'to', 'win', 'fight', 'company', 'trim', 'past', 'year', 'exit', 'unprofitable', 'chip', 'making', 'business', 'focus', 'instead', 'analytics', 'security', 'software', 'well', 'cloud', 'compute', 'product', 'say', 'new', 'business', 'grow', 'company', 'report', 'year', 'year', 'decline', 'major', 'line', 'service', 'revenue', 'business', 'service', 'fall', 'software', 'drop', 'overall', 'revenue', 'sank', 'and', 'worryingly', 'profit', 'margin', 'service', 'software', 'business', 'appear', 'shrink', 'say', 'analyst', 'always', 'move', 'high', 'value', 'snapshot', 'show', 'trouble', 'move', 'margin', 'say', 'low', 'expect', 'tax', 'bill', 'small', 'restructuring', 'cost', 'help', 'company', 'aposs', 'profit', 'soften', 'underperformance', 'core', 'business', 'say', 'result', 'pretty', 'much', 'line', 'expect', 'say', 'dropped', 'hour', 'trade', 'company', 'say', 'cloud', 'compute', 'revenue', 'rise', 'year', 'ago',

...

The IBM Example: A Bag of Words Representation

say	11	focus	2	article	1	despite	1	increase	1	much	1	result	1	technology	1
business	9	go	2	assault	1	dip	1	independent	1	necessarily	1	rise	1	tend	1
revenue	9	help	2	back	1	divestiture	1	industry	1	number	1	sale	1	to	1
company	7	move	2	bank	1	division	1	infrastructure	1	on	1	sank	1	tough	1
service	7	new	2	base	1	do	1	innovator	1	overall	1	scale	1	trade	1
cloud	5	old	2	because	1	dollar	1	instead	1	percentage	1	security	1	transaction	1
profit	5	past	2	bill	1	drop	1	insurance	1	period	1	sell	1	transition	1
year	5	product	2	build	1	erode	1	interview	1	platform	1	show	1	trim	1
billion	4	quarter	2	but	1	estimate	1	introduction	1	pretty	1	shrink	1	trouble	1
compute	4	second	2	by	1	exclude	1	investor	1	prove	1	small	1	try	1
fall	4	acquisition	1	century	1	exit	1	job	1	push	1	snapshot	1	undermine	1
line	4	act	1	change	1	expectation	1	jury	1	quarterly	1	soften	1	underpin	1
margin	3	add	1	charge	1	fight	1	keep	1	question	1	span	1	unprofitable	1
still	3	age	1	chip	1	get	1	lately	1	refresh	1	spur	1	use	1
account	2	always	1	computer	1	grow	1	legacy	1	reinvent	1	storage	1	value	1
ago	2	analytics	1	contribute	1	high	1	low	1	relate	1	string	1	weighed	1
analyst	2	and	1	core	1	hour	1	major	1	relevant	1	strong	1	well	1
boost	2	anywhere	1	cost	1	hurt	1	making	1	remain	1	struck	1	widely	1
decline	2	appear	1	currency	1	idea	1	masterful	1	remarkable	1	successfully	1	win	1
expect	2	application	1	deal	1	important	1	month	1	report	1	tax	1	worryingly	1

In total, there is a large number (5 digits typically, depending on cleaning, bi-grams, etc) of words in the dictionary, only 160 of which appear in this article.

Data: Dow Jones Newswires 1989–2017

Filter	Remaining Sample Size	Observations Removed
Total Number of Dow Jones Newswire Articles	31,492,473	
Combine chained articles	22,471,222	9,021,251
Remove articles with no stocks tagged	14,044,812	8,426,410
Remove articles with more than one stocks tagged	10,364,189	3,680,623
Number of articles whose tagged stocks have three consecutive daily returns from CRSP between Jan 1989 and Dec 2012	6,540,036	
Number of articles whose tagged stocks have open-to-open returns from CRSP since Feb 2004	6,790,592	
Number of articles whose tagged stocks have high-frequency returns from TAQ since Feb 2004	6,708,077	

- 10,364,189 articles tagged with stock tickers
- Sources: Press release wire, WSJ, Barron's, MarketWatch + host of Dow Jones realtime services

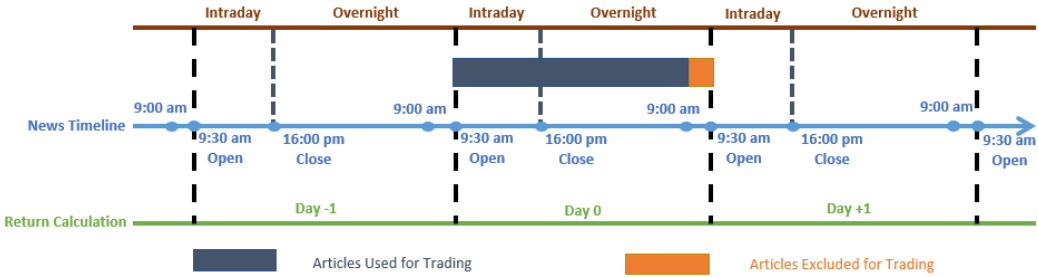
Empirical Strategy

Training (In-sample, 15-year rolling window, 10 year training + 5 year validation)

- Match articles published on day t with return from days $t-1$ through $t+1$

Testing (Out-of-sample, from 2004 to 2017)

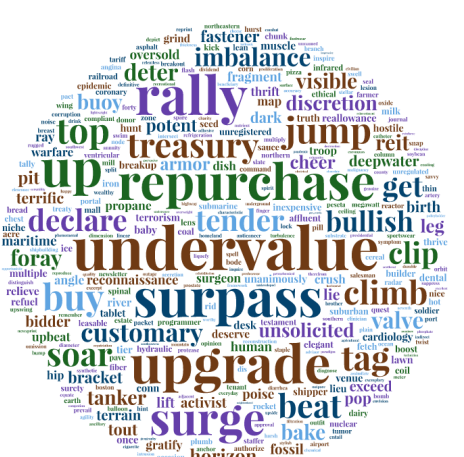
- Using sentiment on day t to predict return on day $t+1$



Sentiment Word Lists

Negative Words

Positive Words

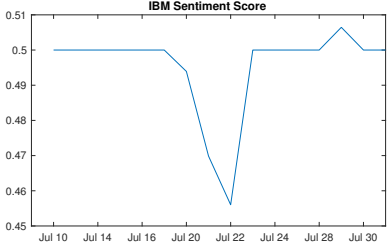


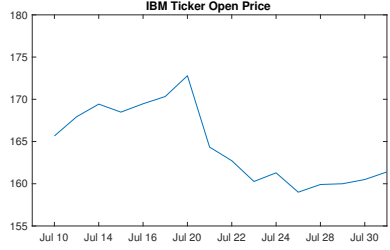
The IBM Example

- The Bag of Words Representation Post Screening:

S	Count
fall	4
erode	1
soften	1
hurt	1
article	1

- The Sentiment Score and IBM stock price:





Forecast Performance Evaluation: A Trading Strategy

Each day, construct out-of-sample estimates of $\hat{p}_i$ for all articles that day

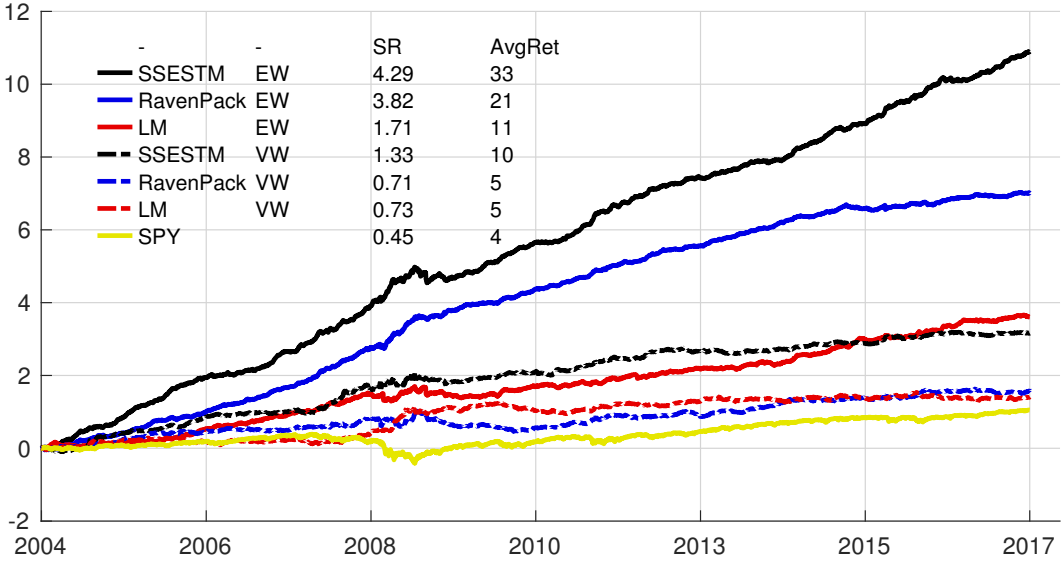
Buy 50 stocks with highest $\hat{p}_i$, sell 50 stocks with lowest

- Equal-weighted and value-weighted constructions
- Zero net investment construction

Evaluate performance from day -10 to day $+10$ relative to article publication date

- Sharpe ratios
- Average daily returns

Comparison with RavenPack and Dictionary Methods



Example: Movie Reviews

- 5000 movie-review snippets from rottentomatoes
- Source: Pang, Bo and Lee, Lillian (2004). Link

Accuracy comparison

- Naive Bayes: 72.0%
- Logistic Regression: 88.2%

Clustering Text: Topic Models

Often, we expect low-dimensional structure underlies Text

- happy/sad, document purpose, topic subject.

We can cluster text based on the topics of different documents.

- We've already learned K-means as a clustering tool
- We now turn to a fancier generative model:
  Instead of each document being from a cluster, we assume each word is from a different topic and the document is a mixture of topics.

Document-Term Matrix

Build matrix of **term counts** as numerical input to statistical analysis $\underbrace{W}_{T \times V}$

- T Rows: Documents (an article of the WSJ)
- V Columns: Terms (1- and 2-grams)

	Terms							
Docum.	brexit	military	dna	whitehouse	yahoo	bomb	chinesecompany	...
6/1/17	2	5	17	0	0	0	5	...
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	
6/8/17	7	2	12	3	0	0	0	
6/9/17	19	2	32	0	15	0	0	
6/12/17	17	2	13	0	0	4	0	
6/13/17	18	2	21	0	0	0	0	
⋮								⋱

LDA Topic Model Blei, Ng, and Jordan (2003)

Data

$\underbrace{W}_{T \times V}$	is the DTM
$\underbrace{w_t}_{V \times 1}$	is the "bag of terms" for document t

Model DGP

$$w_t \sim \text{Mult}(q_t), \quad \underbrace{q_t}_{V \times 1} = \underbrace{\Phi}_{V \times K} \underbrace{\theta_t}_{K \times 1}$$

$\underbrace{\theta_t}_{K \times 1}$	Document t's distribution over K topics
$\underbrace{\phi_k}_{V \times 1}$	Topic k's distribution over V terms

- This is a linear factor model for term counts embedded in a multinomial distribution
- Infer θ_t, ϕ_k via Gibbs sampling

Topic Modeling, or, How to Write a *WSJ* Article

Structure:

- Words grouped into "TOPICS:" Some topics more likely than others

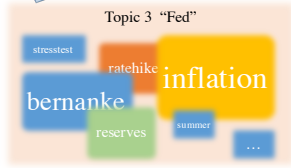
Topic Modeling, or, How to Write a *WSJ* Article

Structure:

- Words grouped into "TOPICS:" Some topics more likely than others

I. Draw topic from urn

- Each topic is itself an urn of words
- Some words more likely than others



Topic Modeling, or, How to Write a *WSJ* Article

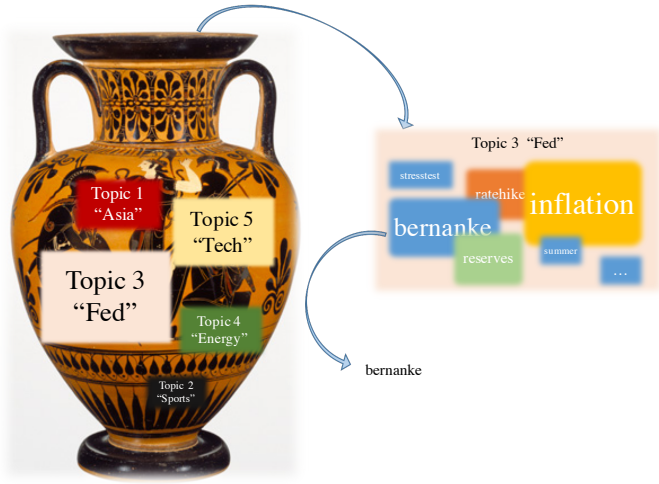
Structure:

- Words grouped into "TOPICS:" Some topics more likely than others

I. Draw topic from urn

- Each topic is itself an urn of words
- Some words more likely than others

II. Draw word from topic



Topic Modeling, or, How to Write a *WSJ* Article

Structure:

- Words grouped into "Topics:" Some topics more likely than others

I. Draw topic from urn

- Each topic is itself an urn of words
- Some words more likely than others

II. Draw word from topic

III. Rinse and repeat

bernanke

Topic Modeling, or, How to Write a *WSJ* Article

Structure:

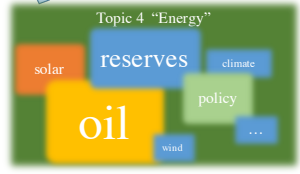
- Words grouped into "Topics:" Some topics more likely than others

I. Draw topic from urn

- Each topic is itself an urn of words
- Some words more likely than others

II. Draw word from topic

III. Rinse and repeat



bernanke

Topic Modeling, or, How to Write a *WSJ* Article

Structure:

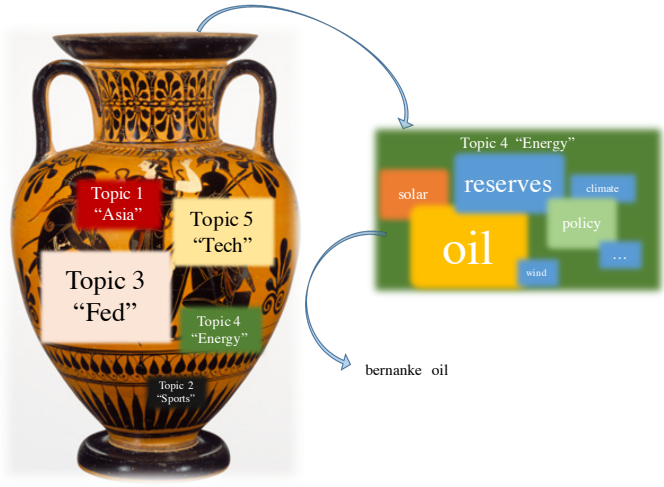
- Words grouped into "TOPICS:" Some topics more likely than others

I. Draw topic from urn

- Each topic is itself an urn of words
- Some words more likely than others

II. Draw word from topic

III. Rinse and repeat



Topic Modeling, or, How to Write a *WSJ* Article

Structure:

- Words grouped into "Topics:" Some topics more likely than others

I. Draw topic from urn

- Each topic is itself an urn of words
- Some words more likely than others

II. Draw word from topic

III. Rinse and repeat

bernanke oil

Topic Modeling, or, How to Write a *WSJ* Article

Structure:

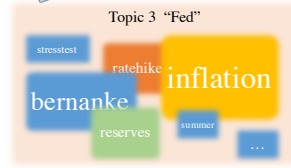
- Words grouped into "TOPICS:" Some topics more likely than others

I. Draw topic from urn

- Each topic is itself an urn of words
- Some words more likely than others

II. Draw word from topic

III. Rinse and repeat



bernanke oil

Topic Modeling, or, How to Write a *WSJ* Article

Structure:

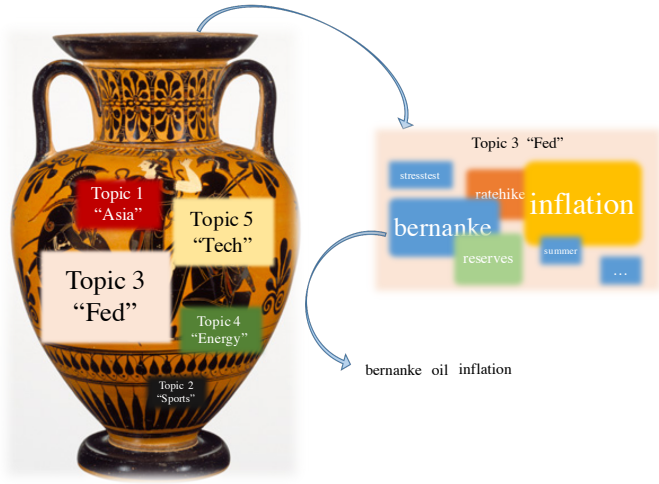
- Words grouped into "Topics:" Some topics more likely than others

I. Draw topic from urn

- Each topic is itself an urn of words
- Some words more likely than others

II. Draw word from topic

III. Rinse and repeat



Topic Modeling, or, How to Write a *WSJ* Article

Structure:

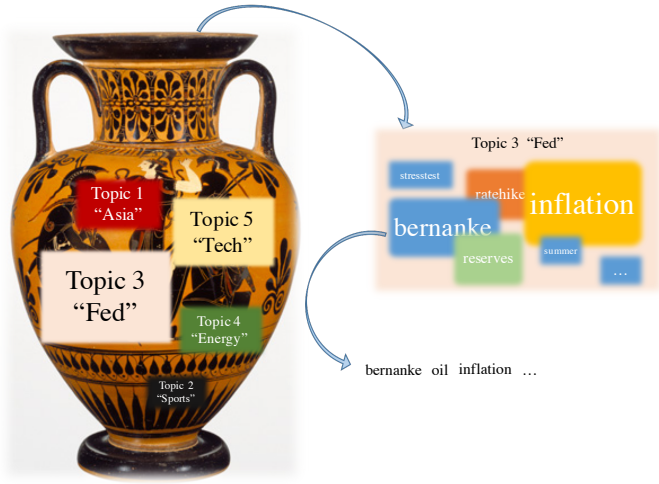
- Words grouped into "TOPICS:" Some topics more likely than others

I. Draw topic from urn

- Each topic is itself an urn of words
- Some words more likely than others

II. Draw word from topic

III. Rinse and repeat



Restaurant reviews from we8there.com

2978 bigrams from 6260 reviews (average of 90 words) with 5-star *overall*, *atmosphere*, *food*, *service*, and *value* ratings.

Great Service: Waffle House #1258, Bossier City LA

I normally would not revue a Waffle House but this one deserves it. The workers, Amanda, Amy, Cherry, James and J.D. were the most pleasant crew I have seen. While it was only lunch, B.L.T. and chili, it was great. The best thing was the 50's rock and roll music, not to loud not to soft. This is a rare exception to what we all think a Waffle House is. Keep up the good work.

Terrible Service: Sartin's Seafood, Nassau Bay TX

Had a very rude waitress and the manager wasn't nice either.

LDA: we8there example

```
texts
 list(['even though', 'larg portion', 'mouth water', 'red sauc', 'babi back',
 'back rib', 'chocol mouss', 'veri satisfi']),
       list(['minut after', 'veri happi']),
       list(['definit recommend', 'waiter veri']), ...,


dictionary = corpora.Dictionary(texts)
bow_corpus = [dictionary.doc2bow(doc) for doc in texts]
lda_model = gensim.models.LdaMulticore(corpus=bow_corpus,id2word=dictionary,
 num_topics=10, passes=20,random_state = 1)

bow_corpus
 [(0, 1), (1, 1), (2, 1), (3, 1), (4, 1), (5, 1), (6, 1), (7, 1)],
 [(8, 1), (9, 1)],
 [(10, 1), (11, 1)]...
```

LDA: we8there example

```
Topic: 0
Words: 0.011*"veri friend" + 0.009*"veri nice" + 0.009*"veri good" + 0.008*"veri pleasant" +
0.007*"came out" + 0.007*"make feel" + 0.006*"server veri" + 0.006*"staff veri" + 0.005*"go back" +
0.005*"come back"
Topic: 1
Words: 0.016*"veri good" + 0.012*"can eat" + 0.010*"ice cream" + 0.006*"food alway" +
0.006*"mexican food" + 0.006*"everi time" + 0.006*"bad experi" + 0.006*"great servic" +
0.005*"chines food" + 0.005*"food veri"
Topic: 2
Words: 0.007*"san francisco" + 0.007*"main cours" + 0.007*"take out" + 0.006*"wide varieti" +
0.006*"veri friend" + 0.005*"fri rice" + 0.005*"can wait" + 0.005*"can go" + 0.005*"mani year" +
0.004*"restaur locat"
Topic: 3
Words: 0.019*"veri good" + 0.016*"dine experi" + 0.011*"food veri" + 0.009*"veri nice" +
0.008*"veri reason" + 0.007*"good food" + 0.006*"food excel" + 0.006*"crab cake" +
0.006*"food servic" + 0.005*"alway great"
......
```

A Large-Scale Example: Business News

All articles published in WSJ from January 1984 to June 2017

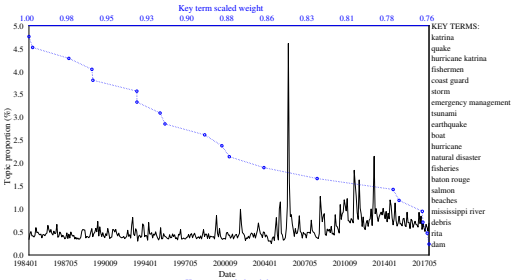
- Most extensive text corpus of business news studied to date
- Cf. "Business News and Business Cycles," Bybee, Kelly, Manela, and Xiu (2021)

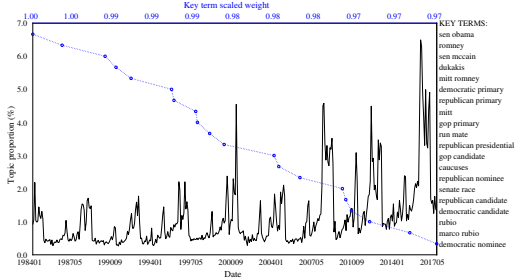
Some choices

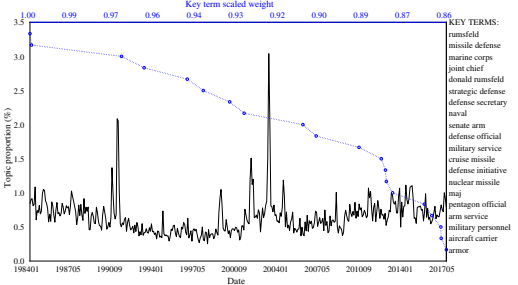
- Data back to 1979, but drop prior to 1984 due to incompleteness
- Focus on three core sections "Section One," "Marketplace," and "Money and Investing"
- Exclude non-core, e.g., "Personal Journal" (initiated 2002); "Weekend Journal" (initiated 2005)
- Mild set of term filters and lemmatization of derivative words
- Vocabulary include all 1-grams and 2-grams after filters

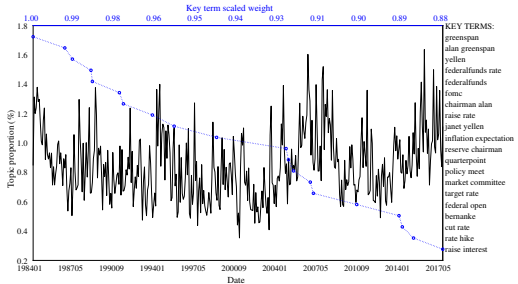
800,000+ articles and 18,000+ terms in final data set

Topic Attention Proportions

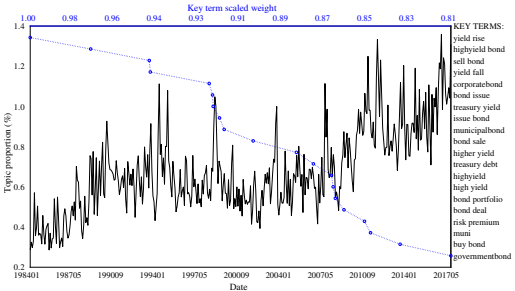


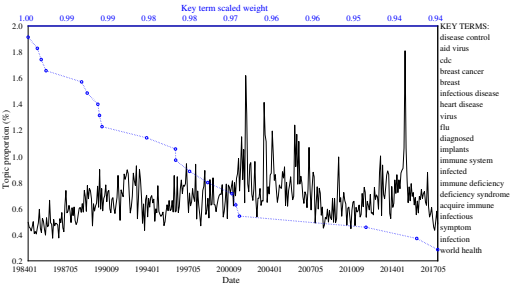


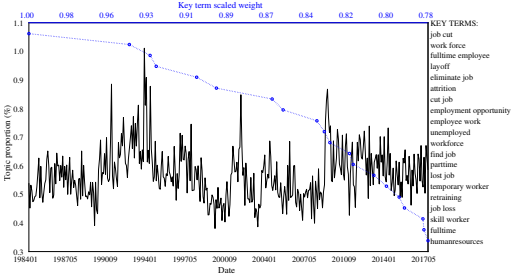


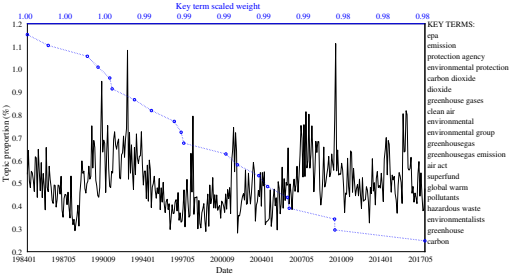


Topic Attention Proportions

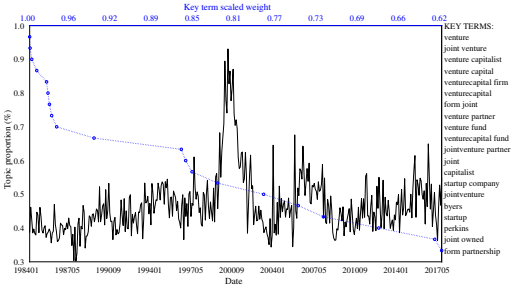


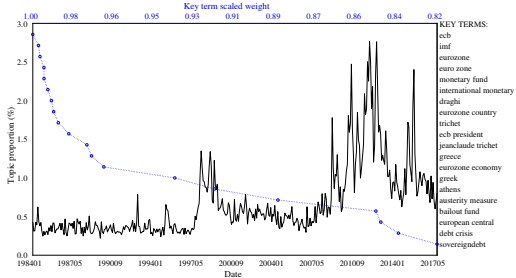


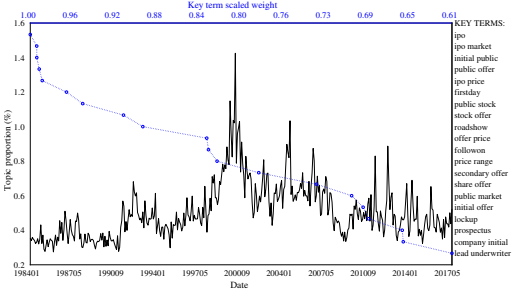


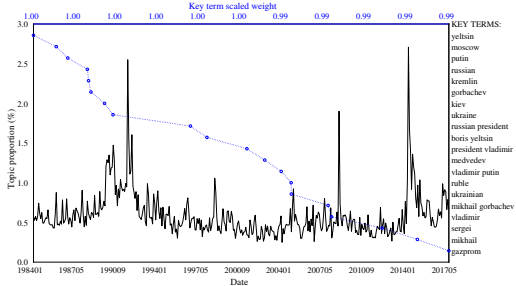


Topic Attention Proportions







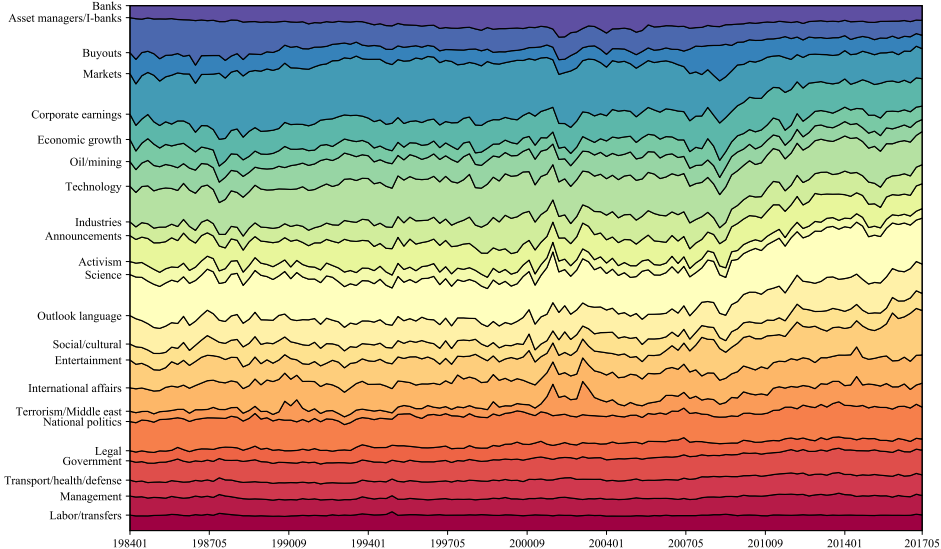


Hierarchical Clustering of WSJ News

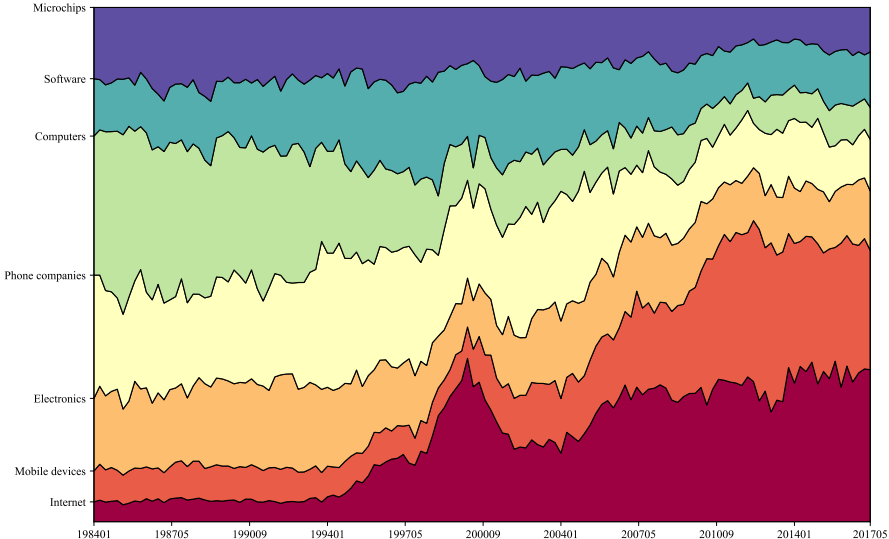
We have data from 180 topics, each being a vector of length 18,433.

See full graph on this website. Source: Bybee, Kelly, Manela, and Xiu, The Structure of Economic News (2021).

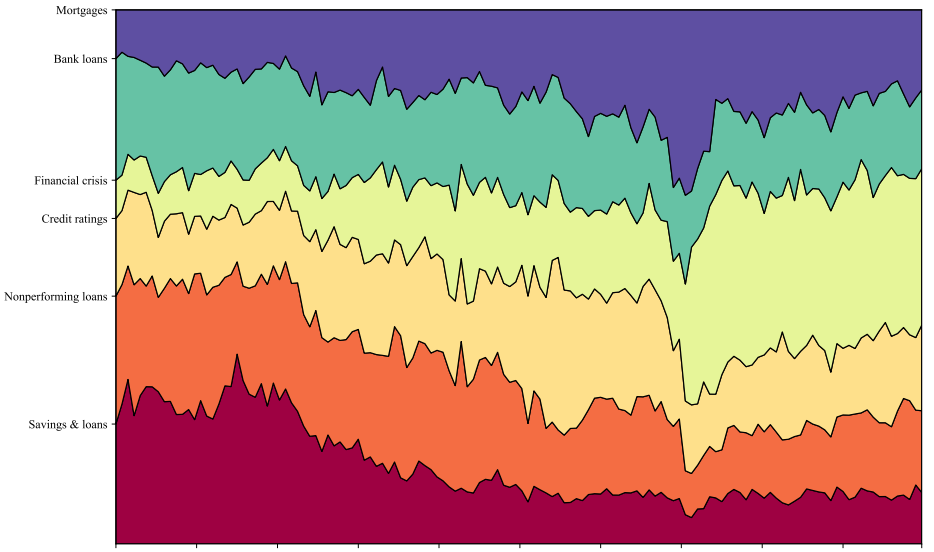
News Attention Across Metatopics



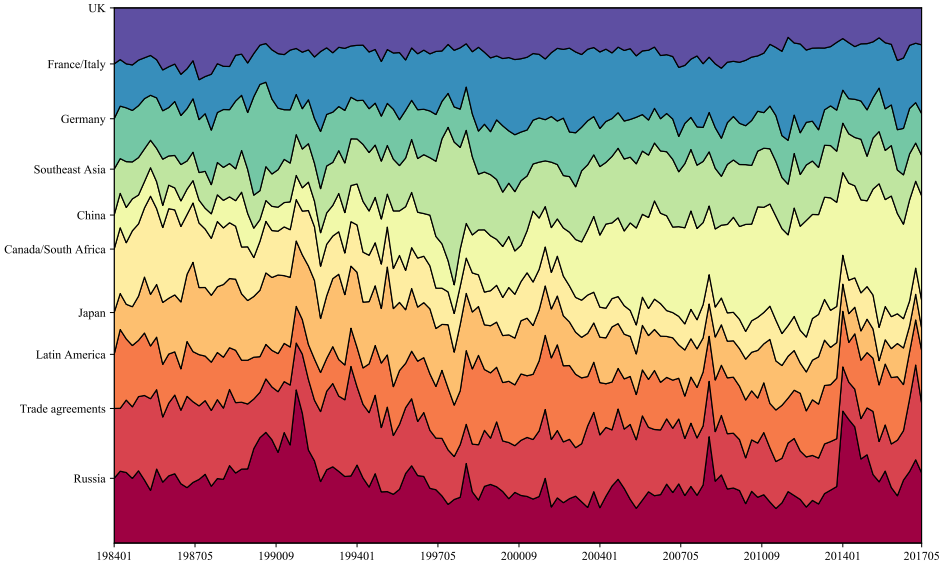
News Attention Within Metatopics – Technology



News Attention Within Metatopics – Banks



News Attention Within Metatopics – International Affairs



Trump Tweet Example

LDA:

```
df.head()
```

```
   date                    text
0 9/26/19 2:46 One of our best fundraising days EVER!
1 9/26/19 2:45 So true but it will never work!
2 9/25/19 20:17 I have informed @GOPLeader Kevin McCarthy and ...
3 9/25/19 18:13 Wow! \Ukraine Whistleblower's lead attorney do...
4 9/25/19 15:56 \He (President Trump) didn't specifically ment...
```

```
dictionary = corpora.Dictionary(processed_docs)
bow_corpus = [dictionary.doc2bow(doc) for doc in processed_docs]
lda_model = gensim.models.LdaMulticore(corpus=bow_corpus,id2word=dictionary,num_topics=10,
passes=20,random_state = 1)
```

LDA: Trump Tweet Example

```
 Topic: 0
Words: 0.026*"great" + 0.015*"united" + 0.014*"today" + 0.014*"country" + 0.013*"states" +
0.012*"honor" + 0.011*"american" + 0.010*"people" + 0.009*"day" + 0.009*"america"
Topic: 1
Words: 0.014*"people" + 0.011*"god" + 0.010*"obamacare" + 0.010*"thank" + 0.008*"bless" +
0.007*"rico" + 0.007*"puerto" + 0.006*"hurricane" + 0.006*"america" + 0.006*"come"
Topic: 2
Words: 0.057*"great" + 0.018*"thank" + 0.018*"job" + 0.015*"big" + 0.014*"america" +
0.012*"state" + 0.011*"win" + 0.008*"love" + 0.008*"vote" + 0.008*"congratulation"
Topic: 3
Words: 0.026*"news" + 0.026*"fake" + 0.017*"medium" + 0.014*"trump" + 0.011*"collusion" +
0.010*"report" + 0.010*"russia" + 0.009*"people" + 0.009*"witch" + 0.009*"hunt"
Topic: 4
Words: 0.034*"border" + 0.020*"democrats" + 0.020*"vote" + 0.017*"want" + 0.016*"security" +
0.014*"wall" + 0.010*"work" + 0.010*"need" + 0.010*"immigration" + 0.010*"dem"
......
```

Representing words in a linguistic way

e.g., WordNet,a thesaurus containing lists of synonym sets and hypernyms

e.g., synonym sets containing "good":

```
from nltk.corpus import wordnet as wn
poses = { 'n':'noun', 'v':'verb', 's':'adj (s)', 'a':'adj', 'r':'adv'}
for synset in wn.synsets("good"):
    print("{}: {}".format(poses[synset.pos()],
            ", ".join([l.name() for l in synset.lemmas()])))
```

```
noun: good
noun: good, goodness
noun: good, goodness
noun: commodity, trade_good, good
adj: good
adj (sat): full, good
adj: good
adj (sat): estimable, good, honorable, respectable
adj (sat): beneficial, good
adj (sat): good
adj (sat): good, just, upright
...
adverb: well, good
adverb: thoroughly, soundly, good
```

e.g., hypernyms of "panda":

```
from nltk.corpus import wordnet as wn
panda = wn.synset("panda.n.01")
hyper = lambda s: s.hypernyms()
list(panda.closure(hyper))
```

```
[Synset('procyonid.n.01'),
Synset('carnivore.n.01'),
Synset('placental.n.01'),
Synset('mammal.n.01'),
Synset('vertebrate.n.01'),
Synset('chordate.n.01'),
Synset('animal.n.01'),
Synset('organism.n.01'),
Synset('living_thing.n.01'),
Synset('whole.n.02'),
Synset('object.n.01'),
Synset('physical_entity.n.01'),
Synset('entity.n.01')]
```

- Great as a resource but missing nuance
- Missing new meanings of words. Requires human labor to create and adapt.
- Subjective
- Can't compute accurate word similarity

Wordnet

```
from nltk.corpus import wordnet as wn
poses = 'n':'noun','v':'verb','s':'adj(s)','a':'adj','r':'adv'
for synset in wn.synsets('good'):
    print(":".format(poses[synset.pos()],",".join([l.name() for l in synset.lemmas()])))
 noun:good,goodness
noun:commodity,trade_good,good
adj:good
adj(s):adept,expert,good,practiced,proficient,skillful,skilful
adj(s):dear,good,near
adv:thoroughly,soundly,good
panda = wn.synset("panda.n.01")
hyper = lambda s: s.hypernyms()
list(panda.closure(hyper))
 [Synset('carnivore.n.01'),
 Synset('placental.n.01'),
 Synset('mammal.n.01'),
 Synset('vertebrate.n.01')
```

Representing words as discrete symbols

Example: hotel, conference, motel – a localist representation. Such symbols for words can be represented by one-hot vectors:

$$\text{motel} = [\,0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\,]$$
$$\text{hotel} = [\,0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\,]$$

Vector dimension = number of words in vocabulary (e.g., 500,000)

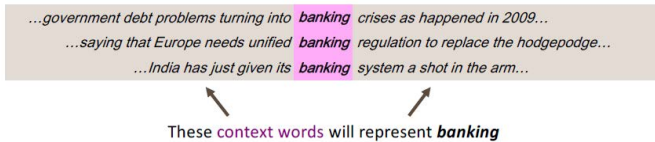
In web search, if user searches for "Seattle motel", we would like to match documents containing "Seattle hotel"

But

These two vectors are orthogonal. There is no natural notion of similarity for one-hot vectors!

Representing words by their context

- Distributional semantics: A word's meaning is given by the words that frequently appear close-by
  - One of the most successful ideas of modern statistical NLP!
- When a word **w** appears in a text, its **context** is the set of words that appear nearby (within a fixed-size window).
- Use the many contexts of **w** to build up a representation of **w**



Word vectors

We will build a dense vector for each word, chosen so that it is similar to vectors of words that appear in similar contexts

$$\textit{banking} = \begin{bmatrix} 0.286 \\ 0.792 \\ -0.177 \\ -0.107 \\ 0.109 \\ -0.542 \\ 0.349 \\ 0271 \end{bmatrix}$$

Note: word vectors are also called word embeddings or (neural) word representations. They are a distributed representation

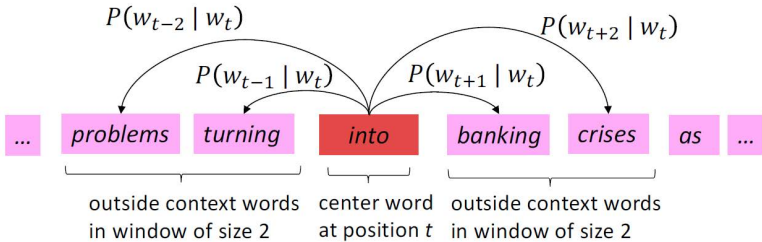
Word2vec: overview

Word2vec (Mikolov et al. 2013) is a framework for learning word vectors

- We have a large corpus ("body") of text
- Every word in a fixed vocabulary is represented by a vector
- Go through each position t in the text, which has a center word c and context ("outside") words o
- Use the similarity of the word vectors for c and o to calculate the probability of o given c (or vice versa)
- Keep adjusting the word vectors to maximize this probability

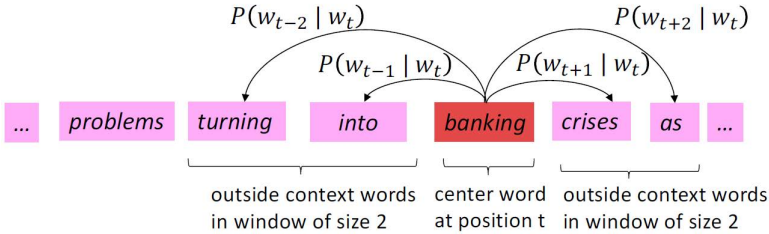
Word2vec: overview

Example windows and process for computing $P(w_{t+j}|W_t)$



Word2vec: overview

Example windows and process for computing $P(w_{t+j}|W_t)$



Word2vec: objective function

For each position $t = 1, ..., T$, predict context words within a window of fixed size m, given center word w_j. Data likelihood:

$$L(\theta) = \prod_{t=1}^{T} \prod_{-m \leq j \leq m, j \neq 0} P(w_{t+j} | w_t; \theta)$$

θ is all variables to be optimized

The objective function (cost / loss function) $J(\theta)$ is the (average) negative log likelihood:

$$J(\theta) = -\frac{1}{T} logL(\theta) = -\frac{1}{T} \sum_{t=1}^{T} \sum_{-m \leq j \leq m, j \neq 0} P(w_{t+j} | w_t; \theta)$$

Minimizing objective function = maximizing predictive accuracy.

Word2vec: objective function

We want to minimize the objective function:

$$J(\theta) = -\frac{1}{T} logL(\theta) = -\frac{1}{T} \sum_{t=1}^{T} \sum_{-m \leq j \leq m, j \neq 0} P(w_{t+j}|w_t; \theta)$$

How to calculate $P(w_{t+j}|w_t; \theta)$?

We will use two vectors per word w:

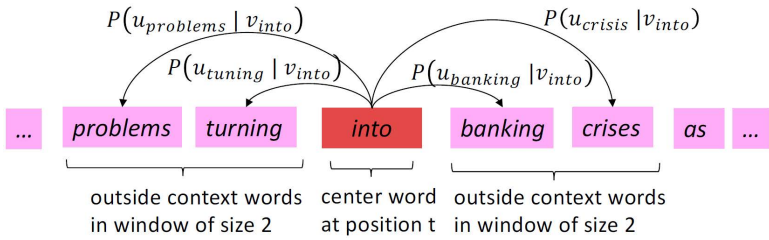
- v_w when w is a center word
- u_w when w is a context word

Then for a center word c and a context word o:

$$P(o|c) = \frac{exp(u_o^T v_c)}{\sum_{w \in V} exp(u_w^T v_c)}$$

Word2vec overview with vectors

- Example windows and process for computing $P(w_{t+j}|w_t)$
- $P(u_{problems}|v_{into})$ short for $P(problems|into; u_{problems}, v_{into}, \theta)$



Word2vec: prediction function

$$P(o|c) = \frac{exp(u_o^T v_c)}{\sum_{w \in V} exp(u_w^T v_c)}$$

- $u_o^T v_c$: dot product compares similarity of o and c. Larger dot product = larger probability
- $exp(.)$: exponentiation makes anything positive
- $1/\sum_{w \in V} exp(u_w^T v_c)$: normalize over entire vocabulary to give probability distribution

$$softmax(x_i) = \frac{exp(x_i)}{\sum_{j=1}^{n} exp(x_j)} = p_i$$

The softmax function maps arbitrary values x_i to a probability distribution p_i

- 'max': amplifies probability of largest x_i
- 'soft': still assigns some probability to smaller x_i
- Frequently used in Deep Learning

To train the model: optimize value of parameters to minimize loss

$$\theta = \begin{bmatrix} v_{aardvark} \\ v_a \\ \dots \\ v_{zebra} \\ u_{aardvark} \\ u_a \\ \dots \\ u_{zebra} \end{bmatrix}$$

- Recall: θ represents all the model parameters, in one long vector
- Remember: every word has two vectors
- We optimize these parameters by walking down the gradient

Word2vec: Hotel Reviews Example

OpinRank Data – Reviews From TripAdvisor

- Full reviews of hotels in 10 different cities (Chicago, Dubai, Beijing, London, New York City, New Delhi, San Francisco, Shanghai, Montreal, Las Vegas)
- There are about 80-700 hotels in each city
- Total number of reviews: about 259,000

Let's take a look at the first review:

"Oct 12 2009 Nice trendy hotel location not too bad.I stayed in this hotel for one night. As this is a fairly new place some of the taxi drivers did not know where it was and/or did not want to drive there. Once I have eventually arrived at the hotel, I was very pleasantly surprised with the decor of the lobby/ground floor area. It was very stylish and modern. As I have a Starwood Preferred Guest member, I was given a small gift upon-check in. It was only a couple of fridge magnets in a gift box, but nevertheless a nice gesture.My room was nice and roomy, there are tea and coffee facilities in each room and you get two complimentary bottles of water plus some toiletries by 'bliss'.The location is not great..."

Word2vec: Hotel Reviews Example

Training the Word2Vec model

```
model = gensim.models.Word2Vec (documents, size=150, window=10, min_count=2, workers=10)
model.train(documents,total_examples=len(documents),epochs=10)
```

Find the similarity

```
w1 = "dirty"
model.wv.most_similar (positive=w1, topn=8)
 [('filthy', 0.8607036471366882),
 ('unclean', 0.7988770008087158),
 ('stained', 0.7778783440589905),
 ('grubby', 0.7598243355751038),
 ('smelly', 0.7596879601478577),
 ('dusty', 0.7550373077392578),
 ('mouldy', 0.7391794323921204),
 ('dingy', 0.7339099049568176)]
```

Word2vec: Hotel Reviews Example

```
 # look up top 6 words similar to 'polite'
w1 = ["polite"]
model.wv.most_similar (positive=w1,topn=6)
 [('courteous', 0.9257561564445496),
 ('friendly', 0.8255083560943604),
 ('cordial', 0.8168084621429443),
 ('curteous', 0.7994595170021057),
 ('professional', 0.7888669967651367),
 ('attentive', 0.7753226161003113)]
# look up top 6 words similar to 'france'
w1 = ["france"]
model.wv.most_similar (positive=w1,topn=6)
 [('spain', 0.6672524213790894),
 ('germany', 0.6538873910903931),
 ('canada', 0.6490622758865356),
 ('england', 0.5917144417762756),
 ('mexico', 0.588631272315979),
 ('hawaii', 0.5879128575325012)]
```

Word2vec: Hotel Reviews Example

```
 # get everything related to stuff on the bed
w1 = ["bed",'sheet','pillow']
w2 = ['couch']
model.wv.most_similar (positive=w1,negative=w2,topn=5)
 [('duvet', 0.7076194882392883),
 ('mattress', 0.6860425472259521),
 ('blanket', 0.6852975487709045),
 ('quilt', 0.6786033511161804),
 ('matress', 0.6751226186752319)]
# Which one is the odd one out in this list?
model.wv.doesnt_match(["cat","dog","france"])
 'france'
# Which one is the odd one out in this list?
model.wv.doesnt_match(["bed","pillow","duvet","shower"])
 'shower'
```