
ARTIFICIAL
INTELLIGENCE
NATURAL LANGUAGE
PROCESSING

What is involved in NLP?

Phonetics / Phonology:

Recover sequence of words from audio signal.

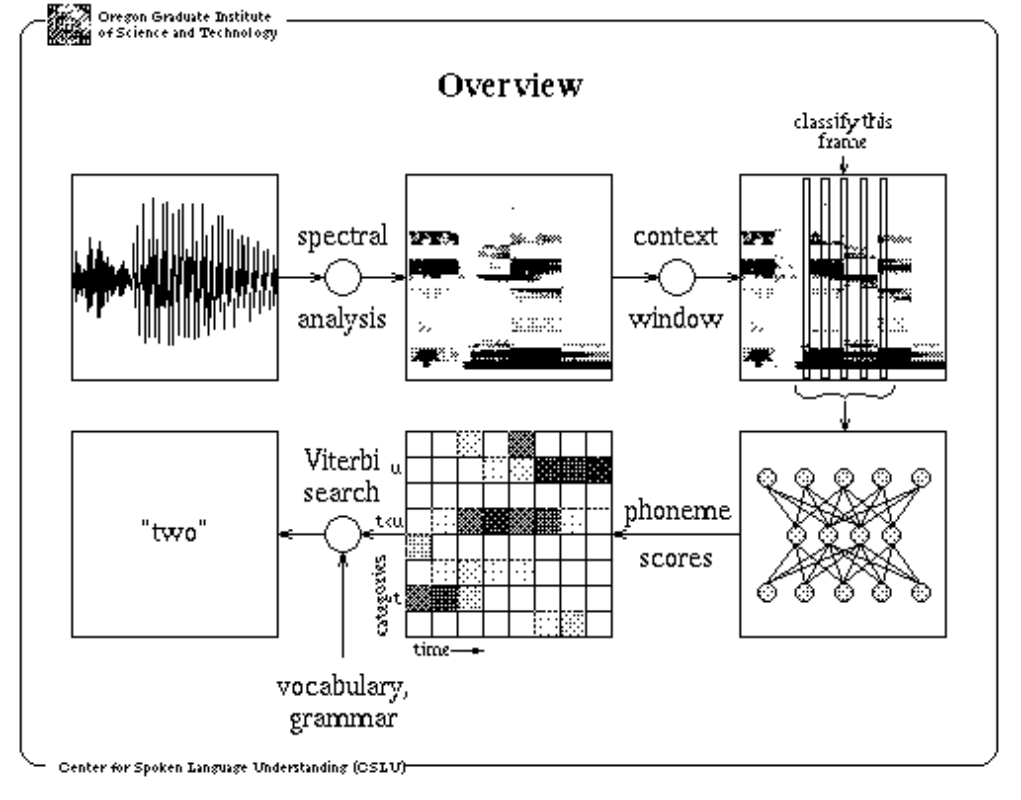
"Open the pod bay doors, HAL."

What is involved in NLP?

Phonetics / Phonology:

Recover sequence of words from audio signal.

"Open the pod bay doors, HAL."



What is involved in NLP?

Text-To-Speech:

Take a sequence of words and generate an audio signal.

"I'm sorry Dave, I'm afraid I can't do that."

What is involved in NLP?

Morphology:

Recognize plurals, contractions, etc.

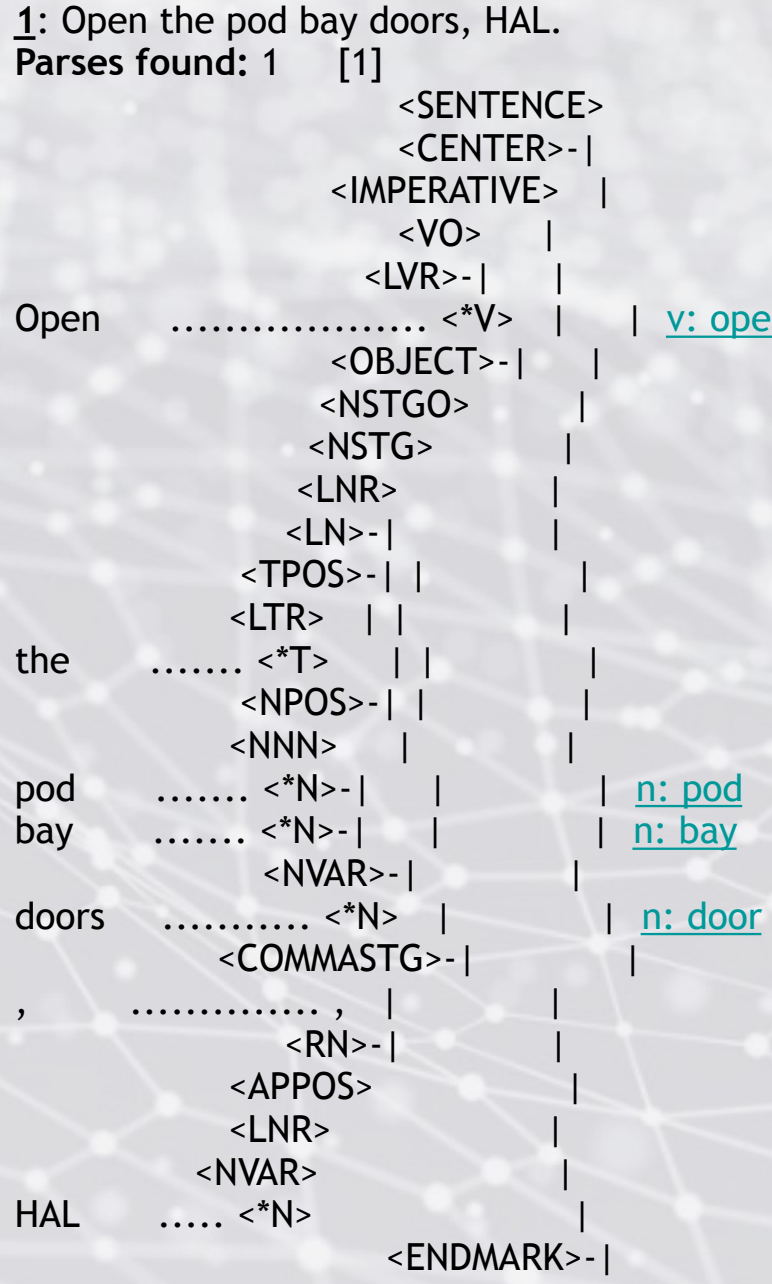
"Open the pod bay doors, HAL."

"I'm sorry Dave, I'm afraid I can't do that."

Syntax:

Parse utterance

Determine type of utterance (e.g., question, request, command)



Lexical Semantics:

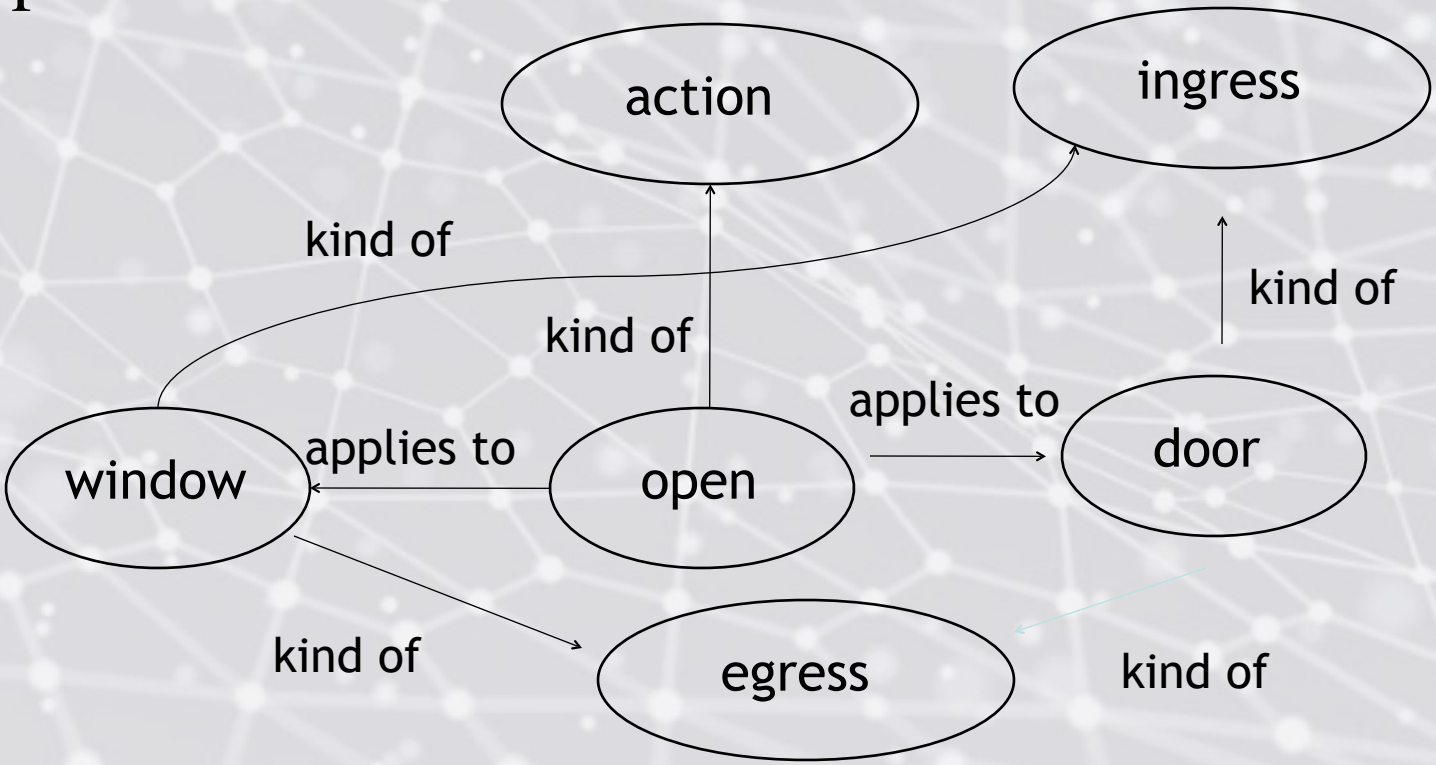
Determine meaning of component words

- 1: Open the pod bay doors, HAL.
  Parse Nr: 1
- Open
- v: open
- 1. open, open_up -- (cause to open or to become open; "Mary opened the car door")
- pod
- n: pod
- 4. fuel_pod, pod -- (a detachable container of fuel on an airplane)
- bay
- n: bay
- 1. bay -- (an indentation of a shoreline larger than a cove but smaller than a gulf)
- doors
- n: door
- 1. door -- (a swinging or sliding barrier that will close the entrance to a room or building; "he knocked on the door"; "he slammed the door as he left")

What is involved in NLP?

Compositional semantics:

Determine meaning from combination of these components.



Pragmatics:

Adapt phrasings to current situation, to accomplish goals.

E.g., politeness:

"I'm sorry Dave, I'm afraid I can't do that."

- Discourse:

  Conversational behavior follows conventions
  (don’t interrupt, respond to requests and questions,
  etc.)

NLP does all this, plus dealing with ambiguity in language

- “I made her duck”
- Meanings?

Machine Translation

- Rule-Based Systems

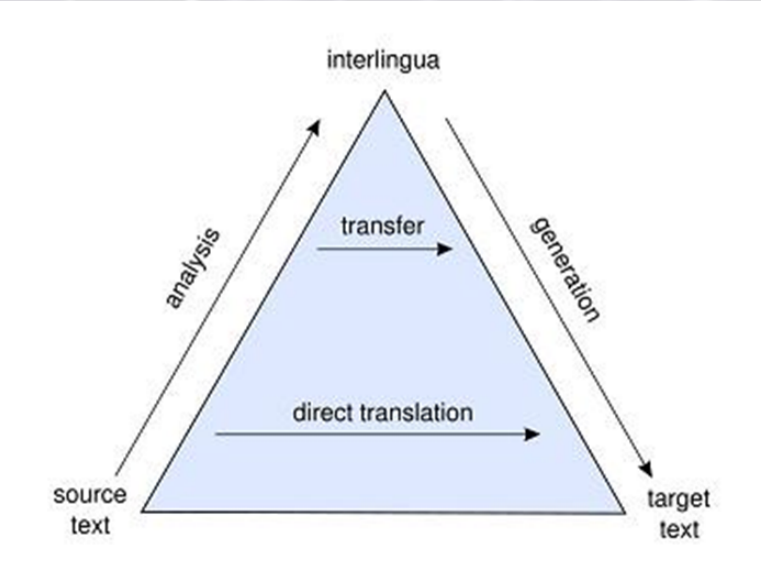
(1)Direct Translation: System that tries to produce a translation directly from a source language to a target language without any intermediate representation; generally dictionary-based; no syntactic analysis.

(2) Transfer Systems: Integrates syntactic analysis; translation process exploits structure of source sentence (avoiding limitations of word-for-word translation).

Machine Translation

- Rule-Based Systems

(3) Interlingua: formal representation of the content to be translated; English is often used as an interlingua (also: Esperanto). Translating from A to B, the system first tries to transfer the content of A to the interlingua before translating from the interlingua to language B.



Machine Translation

Warren Weaver (1949) "Translation"

- Proposed (4) principles to avoid word-to-word errors.

(1) Analyzing the context of words should make it possible to determine their precise meaning.

(2) It should be possible to determine a set of logical and recursive rules to solve the problem of machine translation.

(3) Shannon's model of communication could probably provide useful methods for machine translation .

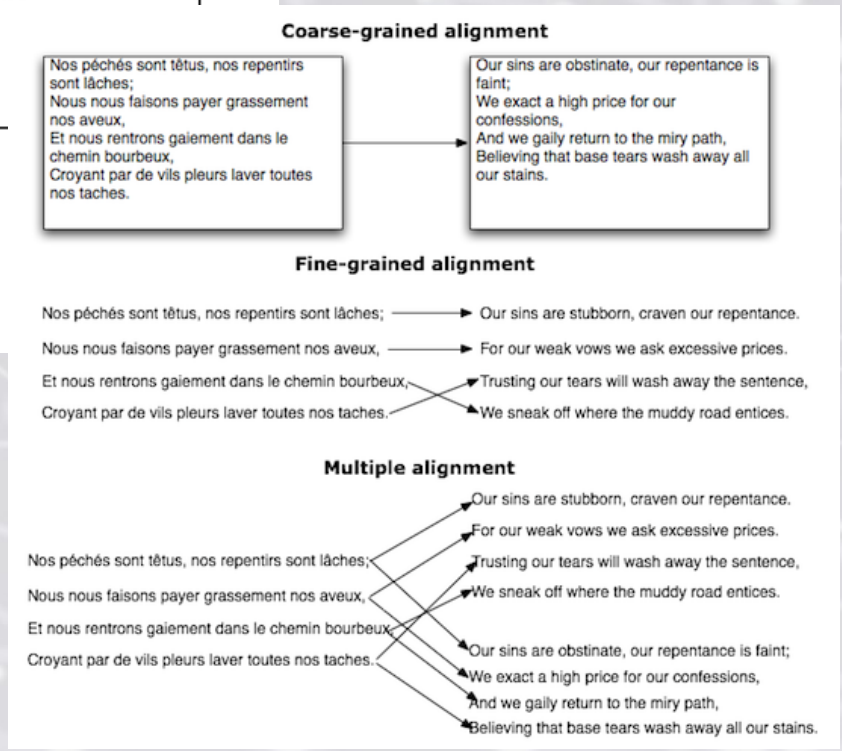
(4) Language can be described with universal elements that may help facilitate the translation process. (Instead of direct translation, search for a more universal and abstract representation that eschews verbatim rendering/ambiguity).

Statistical Language Models & Parallel Corpora

Parallel corpus **C** = a collection of text-chunks and their translations.
Parallel corpora are the by-product of *human translation*.
Every source chunk is paired with a target chunk.

Dutch	English
De prijs van het huis is gestegen.	The price of the house has risen.
Het huis kan worden verkocht.	The house can be sold.
Als het de marktprijs daalt zullen sommige gezinnen een zware tijd doormaken.	If the market price goes down, some families will go through difficult times.
. . .	. . .
. . .	. . .

- Hansards Canadian Parliament Proc. (English-French).
- European Parliament Proc. (23 languages).
- United Nations documents.
- Newspapers: Chinese-English; Arabic-English; Urdu-English.

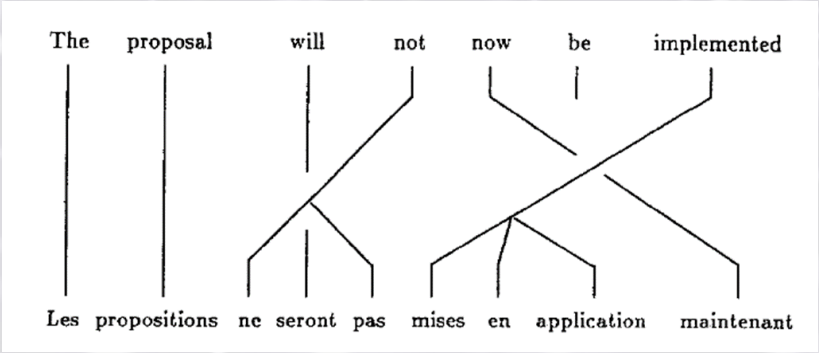


Machine Translation

- Since the late 1990s, aligned bilingual corpora have been the object of various studies aiming to extract translation equivalences between language at the word or phrase level.

- Attempts to produce entirely automatic translation systems through statistical analysis took place at this time (and continues today, also with deep learning).

- The previous slide shows examples of sentence alignment; word alignment is a considerably more complex task.

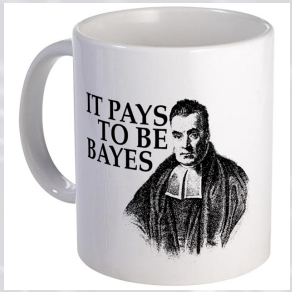


Machine Translation

- At the end of the 1980s, an IBM research

team developed a machine translation system based on techniques initially used for speech transcription.

- Given a pair of sentences: (S,T), for source and target, compute P(T|S).
- Recall: $$P(T \mid S) = \frac{P(S \mid T)P(T)}{P(S)}$$



- Where P(T) measures the prior of the target without taking the source into account (i.e. the probability that T forms a valid, and well-formed sequence in the target language).

Machine Translation

- Since the denominator does not depend on T, we compute the MAP:

$$T_{MAP} = \arg\max_{T} \left[P(T) P(S \mid T) \right]$$

- For the IBM team, this formula was the “fundamental equation of machine translation.”

Example of statistical language models: n-grams

- Estimates probability distribution of a word w, given n-1 words that have come before in the sequence. $P(w|w_1, w_2, \ldots, w_{n-1})$

- Purpose: to guess next word from previous words to disambiguate:
  - "Students have access to a list of course requirements"
  - "Would you like a drink of [garbled]?
  - "He loves going to the [bark]."

- **N grams:** Applications throughout natural language processing:
  - text classification
  - speech recognition
  - machine translation
  - intelligent spell-checking
  - handwriting recognition
  - playing “Jeopardy!”

- What is P(bark|he loves going to the) ?
- What is P(park|he loves going to the) ?

- Can estimate from a large corpus:
  - $P(w \mid w_1, .., w_{n-1})$ = frequency of $w_1 \ldots w_{n-1}\ w$ divided by frequency of $w_1 \ldots w_{n-1}$
  - Example: Use Google

Problem! Web doesn't give us enough examples to get good statistics.

One solution:

- Approximate $P(w|w_1, .., w_{n-1})$ by using small n (e.g., n =2: bigrams).

Bigram example:

P (*bark*|*the*) vs. P(*park*|*the*)

(calculate using Google)

Trigram:

P (*bark*|*to the*) vs. P (*park*|*to the*)

Typically, bigrams are used:

Let candidate utterance $s = w_1 w_2 \ldots w_n$

Then $P(s) = \prod_{k=1}^{n} P(w_k \mid w_{k-1})$ (by the chain rule)

$$P(w_k \mid w_{k-1}) = \frac{C(w_{k-1} w_k)}{C(w_{k-1})}$$ where C stands for "count"

Now can calculate probability of utterance:

P (*he loves going to the bark*) $\propto$

P(*he*|<*s*>) *P* (*loves*| *he*) *P* (*going*| *loves*) *P* (*to*| *going*) *P* (*the*| *to*) *P* (bark | *the*) P(</s> bark)

<*s*> = sentence start marker

</*s*> = sentence end marker

Mini-example
(Adapted from Jurafsky & Martin, 2000)

Corpus 1 (Class 1):

```
<s> I am Sam </s>
<s> Sam I am </s>
<s> I do not like green eggs and ham </s>
```

Corpus 2 (Class 2):

<s> I am he as you are he</s>
<s> I am the Walrus </s>
<s> I am the egg man</s>

Class 1 Bigram Probabilities (examples):

$$P(\text{I} \mid \text{<s>}) = \frac{2}{3} = .67 \qquad P(\text{Sam} \mid \text{<s>}) = \frac{1}{3} = .33 \qquad P(\text{am} \mid \text{I}) = \frac{2}{3} = .67$$
$$P(\text{</s>} \mid \text{Sam}) = \frac{1}{2} = 0.5 \qquad P(\text{Sam} \mid \text{am}) = \frac{1}{2} = .5 \qquad P(\text{do} \mid \text{I}) = \frac{1}{3} = .33$$

Class 1 Bigram Probabilities (examples):

P(I | <s>) = 1
P(are | you) = 1
P(am | I) = 1
P (egg | the) = .5
P(man | egg) = 1
P(the | am) = .67

N-gram approximation to Shakespeare
(Jurafsky and Martin, 2000)

- Trained unigram, bigram, trigram, and quadrigram model on complete corpus of Shakespeare's works (including punctuation).

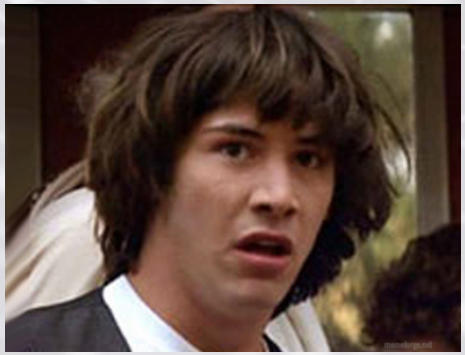
- Use these models to generate random sentences by choosing new unigram/bigram/trigram/quadrigram probabilistically

Unigram model

1. To him swallowed confess hear both. Which. Of save on trail for are ay device and rote life have.

2. Every enter now severally so, let

3. Hill he late speaks; or! a more to leg less first you enter
4. Are where exeunt and sighs have rise excellency took of...Sleep knave we. near; vile like.



Bigram model

1. What means, sir. I confess she? then all sorts, he is trim, captain.

2. Why dost stand forth thy canopy, forsooth; he is this palpable hit the King Henry. Live king. Follow.

3. What we, hath got so she that I rest and sent to scold and nature bankrupt, nor the first gentleman?

4. Thou whoreson chops. Consumption catch your dearest friend, well, and I know where many mouths upon my undoing all but be, how soon, then; we'll execute upon my love's bonds and we do you will?

Trigram model

1. Sweet prince, Falstaff shall die. Harry of Monmouth's grave.

2. This shall forbid it should be branded, if renown made it empty.

3. Indeed the duke; and had a very good friend.

4. Fly, and will rid me these news of price. Therefore the sadness of parting, as they say, 'tis done.

Quadrigram model

1. King Henry. What! I will go seek the traitor Gloucester. Exeunt some of the watch. A great banquet serv'd in;

2. Will you not tell me who I am?

3. Indeed the short and long. Marry, 'tis a noble Lepidus.

4. Enter Leonato's brother Antonio, and the rest, but seek the weary beds of people sick.

Naïve Bayes Text Classification

Classification, learning, and generalization

- General description of **classification**:

  Given a *feature vector* representing a possible instance of a class,

  $$\boldsymbol{x} = \langle x_1, x_2, \ldots, x_n \rangle,$$

  classify $\boldsymbol{x}$ as one of a set of classes $c \in C$.

Classification, learning, and generalization

- General description of **classification**:

  Given a *feature vector* representing a possible instance of a class,

  $$\boldsymbol{x} = \langle x_1, x_2, \ldots, x_n \rangle,$$

  classify $\boldsymbol{x}$ as one of a set of classes $c \in C$.

- General description of **supervised learning**:

  Given a set of *training examples*

  $$\{(\mathbf{x}, c(\mathbf{x}))_{train}\},$$

  where $c(\boldsymbol{x})$ is the correct classification of $\boldsymbol{x}$, construct a hypothesis *h* that will correctly classify these training examples (with the goal of generalization).

Classification, learning, and generalization

- General description of **classification**:

  Given a *feature vector* representing a possible instance of a class,

  $$\boldsymbol{x} = \langle x_1, x_2, \ldots, x_n \rangle,$$

  classify $\boldsymbol{x}$ as one of a set of classes $c \in C$.

- General description of **supervised learning**:

  Given a set of *training examples*

  $$\{(\mathbf{x}, c(\mathbf{x}))_{train}\},$$

  where $c(\boldsymbol{x})$ is the correct classification of $\boldsymbol{x}$, construct a hypothesis *h* that will correctly classify these training examples (with the goal of generalization).

- General description of **successful generalization**:

  Given a set of *test examples* $\{(\mathbf{x}, c(\mathbf{x}))_{test}\}$, not seen before but drawn from the same distribution of the training examples, hypothesis *h* will correctly classify these test examples.

Example: Detecting spam

From: Alibris <books@alibris.m0.net>
Reply-to: books@alibris.m0.net
To: mm@cse.ogi.edu
Subject: Melanie, reminding you to save $10 at Alibris

HOLIDAY SPECIAL: SAVE UP TO $10 ON YOUR PURCHASES
(order now and receive by Christmas)

With the holiday season rapidly approaching, we want to remind you of our most generous sale of the year. As a valued customer, we invite you to save up to $10 off your Alibris purchases with three ways to save:

$2 off your order of $20 or more: GIFT2
$5 off your order of $50 or more: GIFT5
$10 off your order of $100 or more: GIFT10

Simply enter the coupon codes above* at checkout. But hurry, this limited time offer expires on December 16, 2003. Visit Alibris now and save!

Save money on shipping too! Now through December 9, 2003, every item listed on our site should be delivered to continental U.S. and Canadian addresses by December 24th via standard shipping (our lowest cost option) or get FREE shipping when you order $49 of In Stock books.
Don't delay, start your holiday shopping now.
http://alibris.m0.net/m/S.asp?HB10950943733X2869462X274232X

```
From: "Basil Lutz" <0eynsozueb@a-city.de>
Reply-To: "Basil Lutz" <0eynsozueb@a-city.de>
To: <mm@santafe.edu>, <bonabeau@santafe.edu>
Subject: **SPAM 10.70** This tool will make your website more productive hukm

<html>
<head>
<title>hd36 8 ekj 0O9 92O 2                          </title>
<meta http-equiv=3D"Content-Type" content=3D"text/htm; charset=3Diso-8859=
-1">
</head>

<body>
<p><font face=3D"Arial, Helvetica, sans-serif">Can your website answer que=
stions
  in real time 24 hours a day, 7 days a week? Our clients websites do and =
we're
  not talking about some stale FAQ sheet either. Add <a href=3D"http://www=
dreamscaper.co.mn@click.net-click.net.ph/click.php?id=3Ddrcomnm">live
  operator support</a> to your website today and dramatically increase you=
r revenues.</font></p>
<p> </p>
<p> </p>
<p> </p>
<p> </p>
<p> </p>
<p> </p>
<p> </p>
<p> </p>
<p> </p>
<p><a href=3D"http://www.dreamscaper.co.mn@click.net-click.net.ph/click.ph=
p?id=3Ddrcomnx">stop</a>
  sending me emails</p>
</body>
</html>
```

From: =?iso-8859-1?q?james=20ken?= <ja_ken2004@yahoo.fr>
Subject: URGENT ASSISTANCE
To: ja_ken2004@yahoo.fr

FROM:JAMES KEN.

ATTN:

Dear Respectful one,

I know this proposal letter may come to you as a surprise considering the fact that we have not had any formal acquaintance before .but all the same I would want you for the sake of God to give this an immediate attention in view of the fact that the security of our live and possession is at stake .

I am Mr JAMES KEN 28 years old from war ravaged SIERRA LEONE but presently domiciled in Abidjan Ivory coast with my sister JANET who is 18 years old .My father Mr KEN who before his untimely assassination by the rebels was the Director of SIERRA LEONE Diamond corporation (SLDC) .He was killed in our government residential house along side two of my other brothers ,two house maids and one government attached security guard fortunately for I, younger sister and mother ,we were on a week end visit to our home town As we got the news of the tragedy .We immediately managed to ran into neighbouring Ivory coast for refuge .But unfortunately .As Fate would have it ,we lost our dear mother (may soulrest in peace) as a result of what the Doctor called cardiac arrest .

As we were coming into this country ,we had some documents of a deposit of $ 11 700 000 USD (eleven million seven hundred thousand USD) made by my late father in a security and trust company .According to my father, he intended to use this fund for his international business transaction after his tenure in office but was unfortunately murdered .We had located the security company where the money is deposited with the help of an attorney and established ownership .please right now ,with the bitter experiences we had in our country and the war still going on especially in diamond area which incidentally is where we hail from .coupled with the incessant political upheavals and hostilities in this country Ivory coast ,we desire seriously to leave here and live the rest of our life into a more peaceful and politically stable country like yours Hence this proposal and request .We therefore wish you can help us in the following regards :

1)To provide us with a good bank account to transfer the money into.
2)To help us invest the money into a lucrative business .
3)To assist my sister Janet get a college admission to further her education.

Please I know that , this letter may sound strange and incredible to you but the CNN and the BBC African bulletin normally have it as their major news features .Therefore for the sake of God and humanity give an immediate positive consideration and reply to me via our e-mail address. I will willingly agree to any suitable percentage of the money you will propose as your compensation for your assistance with regards to the above .please in view of our sensitive refugee status and as we are still conscious of our father 's enemies .I would like you to give this a highly confidential approach .

Best Regards .
JAMES KEN.

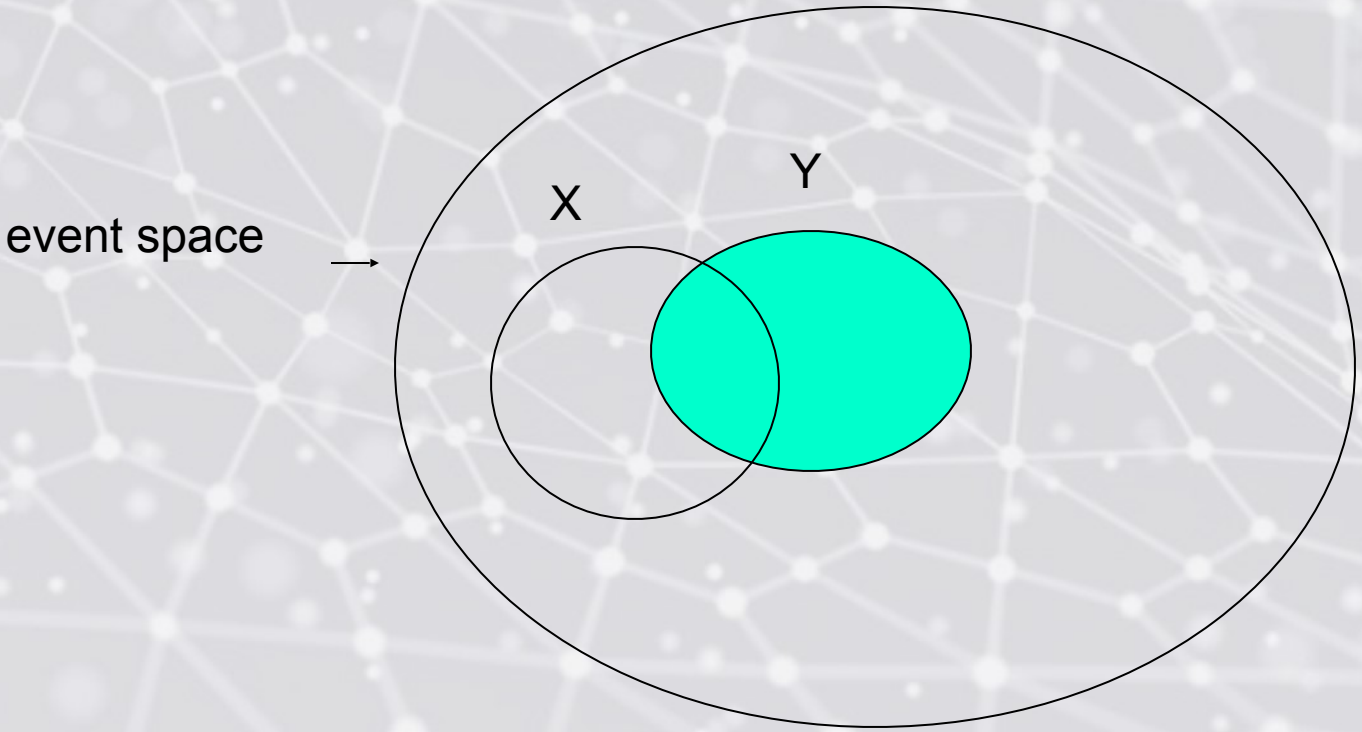
Spamassassin results

```
X-Spam-Report: ---- Start SpamAssassin results
        6.70 points, 4 required;
        *  0.4 -- BODY: Offers a limited time offer
        *  0.1 -- BODY: Free Offer
        *  0.4 -- BODY: Stop with the offers, coupons, discounts etc!
        *  0.1 -- BODY: HTML font color is red
        *  0.1 -- BODY: Image tag with an ID code to identify you
        *  2.8 -- BODY: Bayesian classifier says spam probability is 80 to 90%
                   [score: 0.8204]
        *  0.8 -- BODY: HTML font color is green
        *  0.3 -- BODY: FONT Size +2 and up or 3 and up
        *  0.1 -- BODY: HTML font color not within safe 6x6x6 palette
        *  0.1 -- BODY: HTML font color is blue
        *  0.3 -- BODY: Message is 70% to 80% HTML
        *  1.2 -- Date: is 6 to 12 hours after Received: date
        ---- End of SpamAssassin results
```

Conditional Probability

Recall the definition of conditional probability:

$$P(X \mid Y) = \frac{P(X \cap Y)}{P(Y)}$$



In general :

$$P(X \mid Y)P(Y) = P(X \cap Y) = P(Y \mid X)P(X)$$

Thus,

$$P(X \mid Y) = \frac{P(Y \mid X)P(X)}{P(Y)}$$

Bayes Theorem applied to document classification

Let *C* be a set of possible class of documents and *d* be a document.

To classify *d*, calculate *P*(*c* |*d*) for all $c \in C$ and return the class *c* for which *P*(*c*|*d*) is maximum.

We can calculate *P*(*c* | *d*) using Bayes theorem:

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

where *P*(*c*) is the prior probability of class *c*, and *P*(*d*) is the prior probability of document *d*.

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

To classify document *d*, calculate c_{MAP} (the "maximum a posteriori" class) as follows:

$$c_{MAP} = \arg\max_{c \in C} \left[P(c \mid d) \right] = \arg\max_{c \in C} \left[\frac{P(d \mid c)P(c)}{P(d)} \right]$$

$$= \arg\max_{c \in C} \left[P(d \mid c)P(c) \right]$$

Representation of documents

Let $d = (t_1, t_2, t_3, \ldots, t_n)$, where t_k is a term (from a complete vocabulary) used in document *d* (in sequence).

Example: Suppose the document is *"The rain in Spain falls mainly in the plain."*

Then *d* = (*rain*, *spain*, *fall*, *mainly*, *in*, *plain*)

(Here, a pre-processing step deleted punctuation, "stop" words, did "stemming", and put everything in lower case.)

Naïve Bayes Multinomial Model

We have: $c_{MAP} = \arg\max_{c \in C} \left[P(d \mid c) P(c) \right]$

For our example, supposed we have *C* = {**Weather**, **Not Weather**}, and suppose we have the following training examples:

Document	d	Class
1	(in, weather, rain, today, in, portland)	**Weather**
2	(yesterday, all, my, trouble, were, far, away)	**Not Weather**
3	(yesterday, mainly, rain, fall, fog, spain)	**Weather**
4	(today, spain, day, in, portland)	**Not Weather**

Suppose our vocabulary consisted of 1000 possible terms.

For our test document, we'd calculate

$$c_{MAP} = \arg\max_{c \in C} \left[P((rain, spain, fall, mainly, in, plain) \mid c) P(c) \right]$$

Problem:

$$c_{MAP} = \arg\max_{c \in C} \left[P((rain, spain, fall, mainly, in, plain) \mid c) P(c) \right]$$

We can't calculate $P((rain, spain, fall, mainly, in, plain) \mid c)$ from the training data!

Naïve Bayes independence assumption:

$$P((rain, spain, fall, mainly, in, plain) \mid c)$$
$$= P(rain \mid c) P(spain \mid c) P(fall \mid c) P(mainly \mid c) P(in \mid c) P(plain \mid c)$$

which we **can** calculate from the training data! (Well, almost…)

Let $P(t_k \mid c) =$

$$P(t_k \mid c) = \frac{\text{Number of occurrences of term } t_k \text{ in training documents of class } c}{\text{sum of lengths of documents of class } c}$$

Calculating a model from the training data

Training examples:

Document	*d*	Class
1	(in, weather, rain, today, in, portland)	**Weather**
2	(yesterday, mainly, rain, fall, fog, spain)	**Weather**
3	(yesterday, all, my, trouble, were, mainly, far, away)	**Not Weather**
4	(today, spain, day, in, portland)	**Not Weather**

Model from training data :

P(*rain* | Weather) = 2/12
P(*spain* | Weather) =1/12
P(*fall* | Weather) = 1/12
P(*mainly* | Weather) = 1/12
P(*in* | Weather) = 1/12
P(*plain* | Weather) = 0/12

P(*rain* | Not Weather) = 0/13
P(*spain* | Not Weather) = 1/13
P(*fall* | Not Weather) = 0/13
P(*mainly* | Not Weather) = 1/13
P(*in* | Not Weather) = 1/13
P(*plain* | Not Weather) = 0/13

Predicting the class from the model

Model from training data :

P(*rain* | Weather) = 2/12
P(*spain* | Weather) =1/12
P(*fall* | Weather) = 1/12
P(*mainly* | Weather) = 1/12
P(*in* | Weather) = 1/12
P(*plain* | Weather) = 0/12

P(*rain* | Not Weather) = 0/13
P(*spain* | Not Weather) = 1/13
P(*fall* | Not Weather) = 0/13
P(*mainly* | Not Weather) = 1/13
P(*in* | Not Weather) = 1/13
P(*plain* | Not Weather) = 0/13

Test document:

d = (*rain*, *spain*, *fall*, *mainly*, *in*, *plain*)

$$c_{MAP} = \arg\max_{c}\left[P(c)P(rain\,|\,c)P(spain\,|\,c)P(fall\,|\,c)P(mainly\,|\,c)P(in\,|\,c)P(plain\,|\,c)\right]$$

Denote **Weather** as W, **Not Weather** as $\overline{\mathrm{W}}$

$$P(W)P(rain\,|\,W)P(spain\,|\,W)P(fall\,|\,W)P(mainly\,|\,W)P(in\,|\,W)P(plain\,|\,W)$$
$$= (1/2)(2/12)(1/12)(1/12)(1/12)(1/12)(0) = 0$$

$$P(\overline{W})P(rain\,|\,\overline{W})P(spain\,|\,\overline{W})P(fall\,|\,\overline{W})P(mainly\,|\,\overline{W})P(in\,|\,\overline{W})P(plain\,|\,\overline{W})$$
$$= (1/2)(0)(1/13)(0)(1/13)(1/13)(0) = 0$$

Predicting the class from the model

Problem: Get zeros from sparse data

One solution: Add 1 to each count for **all** words in vocabulary ("Add-1 smoothing)

Let's assume we have a total vocabulary of 1000 terms

0/13
= 1/13
/13
= 1/13

P(*in* | Weather) = 1/12
P(*plain* | Weather) = 0/12

P(*in* | Not Weather) = 1/13
P(*plain* | Not Weather) = 0/13

Test document:
d = (*rain*, *spain*, *fall*, *mainly*, *in*, *plain*)

$$c_{MAP} = \arg\max_{c}\left[P(c)P(rain\,|\,c)P(spain\,|\,c)P(fall\,|\,c)P(mainly\,|\,c)P(in\,|\,c)P(plain\,|\,c)\right]$$

Denote **Weather** as W, **Not Weather** as $\overline{\mathrm{W}}$

$$P(W)P(rain\,|\,W)P(spain\,|\,W)P(fall\,|\,W)P(mainly\,|\,W)P(in\,|\,W)P(plain\,|\,W)$$
$$= (1/2)(2/12)(1/12)(1/12)(1/12)(1/12)(0) = 0$$

$$P(\overline{W})P(rain\,|\,\overline{W})P(spain\,|\,\overline{W})P(fall\,|\,\overline{W})P(mainly\,|\,\overline{W})P(in\,|\,\overline{W})P(plain\,|\,\overline{W})$$
$$= (1/2)(0)(1/13)(0)(1/13)(1/13)(0) = 0$$

Predicting the class from the model

Model from training data :

P(*rain* | W) = (2+1)/(12+1000)

P(*spain* | W) =(1+1)/(12+1000)

P(*fall* | W) = (1+1)/(12+1000)

P(*mainly* | W) = (1+1)/(12+1000)

P(*in* | W) = (1+1)/(12+1000)

P(*plain* | W) = (0+1)/(12+1000)

P(*rain* | $\overline{W}$) = (0+1)/(13+1000)

P(*spain* | $\overline{W}$) = (1+1)/(13+1000)

P(*fall* | $\overline{W}$) = (0+1)/(13+1000)

P(*mainly* | $\overline{W}$) = (1+1)/(13+1000)

P(*in* | $\overline{W}$) = (1+1)/(13+1000)

P(*plain* | $\overline{W}$) = (0+1)/(13+1000)

Predicting the class from the model

Model from training data :

P(*rain* | W) = (2+1)/(12+1000)
P(*spain* | W) =(1+1)/(12+1000)
P(*fall* | W) = (1+1)/(12+1000)
P(*mainly* | W) = (1+1)/(12+1000)
P(*in* | W) = (1+1)/(12+1000)
P(*plain* | W) = (0+1)/(12+1000)

P(*rain* | $\overline{W}$) = (0+1)/(13+1000)
P(*spain* | $\overline{W}$) = (1+1)/(13+1000)
P(*fall* | $\overline{W}$) = (0+1)/(13+1000)
P(*mainly* | $\overline{W}$) = (1+1)/(13+1000)
P(*in* | $\overline{W}$) = (1+1)/(13+1000)
P(*plain* | $\overline{W}$) = (0+1)/(13+1000)

$$P(W)P(rain|W)P(spain|W)P(fall|W)P(mainly|W)P(in|W)P(plain|W)$$
$$= (1/2)(3/1012)(2/1012)(2/1012)(2/1012)(2/1012)(1/1012) = \mathbf{2.23 \times 10^{-17}}$$

$$P(\overline{W})P(rain|\overline{W})P(spain|\overline{W})P(fall|\overline{W})P(mainly|\overline{W})P(in|\overline{W})P(plain|\overline{W})$$
$$= (1/2)(1/1013)(2/1013)(1/1013)(2/1013)(2/1013)(1/1013) = \mathbf{3.7 \times 10^{-18}}$$

So, *d* is classified as **Weather**.

Predicting the class from the model

Model from training data :

P(*rain* | W) = (2+1)/(12+1000)
P(*spain* | W) =(1+1)/(12+1000)
P(*fall* | W) = (1+1)/(12+1000)
P(*mainly* | W) = (1+1)/(12+1000)
P(*in* | W) = (1+1)/(12+1000)
P(*plain* | W) = (0+1)/(12+1000)

P(*rain* | $\overline{W}$) = (0+1)/(13+1000)
P(*spain* | $\overline{W}$) = (1+1)/(13+1000)
P(*fall* | $\overline{W}$) = (0+1)/(13+1000)
P(*mainly* | $\overline{W}$) = (1+1)/(13+1000)
P(*in* | $\overline{W}$) = (1+1)/(13+1000)

However, another problem: floating-point underflow

Common solution: use logs of values

$$P(W)P(rain|W)P(spain|W)P(fall|W)P(mainly|W)P(in|W)P(plain|W)$$
$$= (1/2)(3/1012)(2/1012)(2/1012)(2/1012)(2/1012)(1/1012) = \mathbf{2.23 \times 10^{-17}}$$

$$P(\overline{W})P(rain|\overline{W})P(spain|\overline{W})P(fall|\overline{W})P(mainly|\overline{W})P(in|\overline{W})P(plain|\overline{W})$$
$$= (1/2)(1/1013)(2/1013)(1/1013)(2/1013)(2/1013)(1/1013) = \mathbf{3.7 \times 10^{-18}}$$

So, *d* is classified as **Weather**.

More generally, here is the formula for the Naïve Bayes Classifier:

$$class_{NB} = \underset{c \in C}{\operatorname{argmax}} P(c) \prod_i P(t_i | c)$$

$$class_{NB} = \underset{c \in C}{\operatorname{argmax}} \left[\log P(c) + \sum_i \log P(t_i | c) \right]$$

Summary:
Naïve Bayes classifier for text classification

$$class_{NB} = \underset{c \in C}{\operatorname{argmax}} \left[\log P(c) + \sum_i \log P(t_i \mid c) \right]$$

(use with Add-1 Smoothing)

Summary:
Naïve Bayes classifier for text classification

$$class_{NB} = \underset{c \in C}{\operatorname{argmax}} \left[\log P(c) + \sum_i \log P(t_i \mid c) \right]$$

(use with Add-1 Smoothing)

Question 1: Is independence assumption a good assumption?

Question 2: If not, why does Naïve Bayes work so well in practice?

Naïve Bayes for text classification (Recap)

Assume we have a set of *C* classes, and a set of training documents
D = ($\mathbf{d}_1$, $\mathbf{d}_2$, …, $\mathbf{d}_n$).

Each document $\mathbf{d}_i$ is represented as a list of term frequencies for the terms in the document:

$$\mathbf{d}_i = (t_1, t_2, \ldots, t_n)$$

Given a new document **d**, we classify **d** as:

$$class(\mathbf{d}) = \arg\max_{c \in C} P(c \mid \mathbf{d})$$

By Bayes rule, we have:

$$class_{NB}(\mathbf{d}) = \arg\max_{c \in C} P(c \mid \mathbf{d}) = \arg\max_{c \in C} \frac{P(c)P(\mathbf{d} \mid c)}{P(\mathbf{d})}$$

$$= \arg\max_{c \in C} P(c)P(\mathbf{d} \mid c) \quad \text{(since the denominator doesn't depend on } c\text{)}$$

$$= \arg\max_{c \in C} P(c) \prod_i P(t_i \mid c) \quad \text{(by Naive Bayes independence assumption)}$$

$$= \arg\max_{c \in C} \left[\log(P(c)) + \sum_i \log(P(t_i \mid c)) \right]$$

Vector space model for text classification and information retrieval

A document is represented as a vector of word counts. E.g.,

"To be or not to be, that is the question. Whether 'tis nobler in the mind to suffer the slings and arrows of outrageous fortune, or to take arms against a sea of troubles, and by opposing end them?"

Vector $d = (c_1, c_2, \ldots, c_M)$, where M is the total number of words in the system's dictionary.

arrows … be … not … or … slings … to …

d = (1 … 2 … 1 … 2 … 1 … 4 …)

Suppose these terms are found in two on-line recipes

Recipe 1:

Chicken: 8

Fried: 2

Oil: 7

Pepper: 4

$d_i = (8, 2, 7, 4)$

Recipe 2:

Chicken: 6

Fried: 0

Oil: 0

Pepper: 0

$d_k = (6, 0, 0, 0)$

Query: fried chicken

$q = (1, 1, 0, 0)$

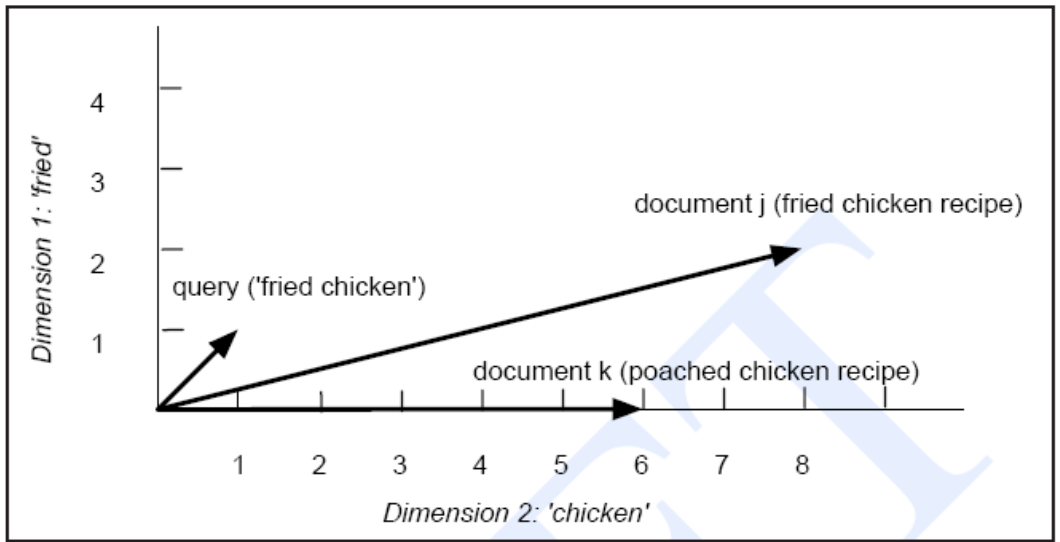
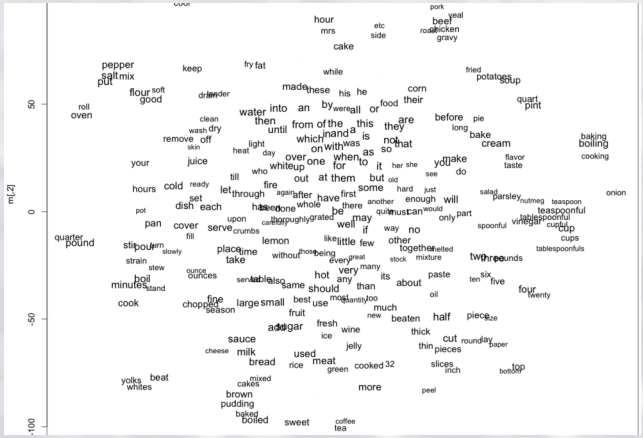


Figure 23.3 A graphical illustration of the vector model for information retrieval, showing the first two dimensions (*fried* and *chicken*) assuming that we use raw frequency in the document as the feature weights.



word2vec

(WATER - WET) + FIRE = FLAMES

(PARIS - FRANCE) + ITALY = ROME

(WINTER - COLD) + SUMMER = WARM

(MINOTAUR - MAZE) + DRAGON = SIMCITY

$$sim(\vec{q},\vec{d}_j) = \frac{\sum_{i=1}^{N} w_{i,q} \times w_{i,j}}{\sqrt{\sum_{i=1}^{N} w_{i,q}^2} \times \sqrt{\sum_{i=1}^{N} w_{i,j}^2}}$$

$$= \frac{dot\text{-}product(\mathbf{q},\mathbf{d}_j)}{length(\mathbf{q}) \times length(\mathbf{d}_j)} = \text{"normalized dot product"}$$

From Jurafsky and Martin, 2006

Suppose these terms are found in two on-line recipes
(assume vocabulary size = 4)

Recipe 1:

Chicken: 8

Fried: 2

Oil: 7

Pepper: 4

$\mathbf{d}_i = (8, 2, 7, 4)$

Recipe 2:

Chicken: 6

Fried: 0

Oil: 0

Pepper: 0

$\mathbf{d}_k = (6, 0, 0, 0)$

Query: fried chicken

$\mathbf{q} = (1, 1, 0, 0)$

$$sim(\vec{q}, \vec{d}_j) = \frac{\sum_{i=1}^{N} w_{i,q} \times w_{i,j}}{\sqrt{\sum_{i=1}^{N} w_{i,q}^2} \times \sqrt{\sum_{i=1}^{N} w_{i,j}^2}}$$

Weaknesses of vector space model?

- Homonymy/Polysemy (“lie”, “bore”, “fly”)
- Synonomy (“canine/dog”, “party/celebration”)

Improving performance of vector-space models

- Improve query:
  - Relevance feedback
  - Query expansion
- Capture “meaning” from context:
  - Latent semantic analysis

Beyond N-grams: Latent Semantic Analysis

- Problem: How to capture semantic similarity between documents in a natural corpus (e.g., problems of homonymy, polysemy, synonymy, etc.)

- In general, N-grams, word frequencies, etc. often fail to capture semantic similarity, even with query expansion, etc.

- "LSA assumes that there exists a LATENT structure in word usage – obscured by variability in word choice" (http://ir.dcs.gla.ac.uk/oldseminars/Girolami.ppt)

Latent Semantic Analysis

(Landauer et al.)

- From training data (large sample of documents), create term-by-document matrix.

Technical Memo Example

Titles:

c1: *Human* machine *interface* for Lab ABC *computer* applications
c2: A *survey* of *user* opinion of *computer system response time*
c3: The *EPS user interface* management *system*
c4: *System* and *human system* engineering testing of *EPS*
c5: Relation of *user*-perceived *response time* to error measurement

m1: The generation of random, binary, unordered *trees*
m2: The intersection *graph* of paths in *trees*
m3: *Graph minors* IV: Widths of *trees* and well-quasi-ordering
m4: *Graph minors*: A *survey*

A sample dataset consisting of the titles of 9 technical memoranda. Terms occurring in more than one title are italicized. There are two classes of documents - five about human-computer interaction (c1-c5) and four about graphs (m1-m4). This dataset can be described by means of a term by document matrix where each cell entry indicates the frequency with which a term occurs in a document.

From Deerwester et al., Indexing by latent semantic analysis

Terms	**Documents**								
	c1	c2	c3	c4	c5	m1	m2	m3	m4
human	1	0	0	1	0	0	0	0	0
interface	1	0	1	0	0	0	0	0	0
computer	1	1	0	0	0	0	0	0	0
user	0	1	1	0	1	0	0	0	0
system	0	1	1	2	0	0	0	0	0
response	0	1	0	0	1	0	0	0	0
time	0	1	0	0	1	0	0	0	0
EPS	0	0	1	1	0	0	0	0	0
survey	0	1	0	0	0	0	0	0	1
trees	0	0	0	0	0	1	1	1	0
graph	0	0	0	0	0	0	1	1	1
minors	0	0	0	0	0	0	0	1	1

- Now apply “singular value decomposition” to this matrix

- SVD is similar to principal components analysis (if you know what that is)

- Basically, reduce dimensionality of the matrix by re-representing matrix in terms of “features” (derived from eigenvalues and eigenvectors), and using only the ones with highest value.

- Result: Each document is represented by a vector of features obtained by SVD.

- Given a new document (or query), compute its representation vector in this feature space, compute its similarity with other documents using cosine between vector angles. Retrieve documents with highest similarities.

Result of Applying LSA

From http://lair.indiana.edu/courses/i502/lectures/lect6.ppt

	DOC1	DOC2	DOC3	DOC4	DOC5	DOC6	DOC7	DOC8	DOC9
HUMAN	0.16	0.4	0.38	0.47	0.18	-0.05	-0.12	-0.16	-0.09
INTERFACE	0.14	0.37	0.33	0.4	0.16	-0.03	-0.07	-0.1	-0.04
COMPUTER	0.15	0.51	0.36	0.41	0.24	0.02	0.06	0.09	0.12
USER	0.26	0.84	0.61	0.7	0.39	0.03	0.08	0.12	0.19
SYSTEM	0.45	1.23	1.05	1.27	0.56	-0.07	-0.15	-0.21	-0.05
RESPONSE	0.16	0.58	0.38	0.42	0.28	0.06	0.13	0.19	0.22
TIME	0.16	0.58	0.38	0.42	0.28	0.06	0.13	0.19	0.22
EPS	0.22	0.55	0.51	0.63	0.24	-0.07	-0.14	-0.2	-0.11
SURVEY	0.1	0.53	0.23	0.21	0.27	0.14	0.31	0.44	0.42
TREES	-0.06	0.23	-0.14	-0.27	0.14	0.24	0.55	0.77	0.66
GRAPH	-0.06	0.34	-0.15	-0.3	0.2	0.31	0.69	0.98	0.85
MINORS	-0.04	0.25	-0.1	-0.21	0.15	0.22	0.5	0.71	0.62

In the above matrix we can now observe correlations:

r(human.user) = 0.94

r(human.minors) = -0.83

How does it find the latent associations?

From http://lair.indiana.edu/courses/i502/lectures/lect6.ppt

- By analyzing the contexts in which the words appear
- The word user has co-occurred with words that human has co-occurred with (e.g., system and interface)
- It downgrades associations when such contextual similarities are not found

Some General LSA Based Applications

From http://lsa.colorado.edu/~quesadaj/pdf/LSATutorial.pdf

Information Retrieval

Text Assessment

Compare document to documents of known quality / content

Automatic summarization of text

Determine best subset of text to portray same meaning

Categorization / Classification

Place text into appropriate categories or taxonomies

Application: Automatic Essay Scoring (in collaboration with Educational Testing Service)

Create domain semantic space

Compute vectors for essays, add to vector database

To predict grade on a new essay, compare it to ones previously scored by humans

From http://lsa.colorado.edu/~quesadaj/pdf/LSATutorial.pdf

Mutual information between two sets of grades:

human – human .90

LSA – human .81

From http://lsa.colorado.edu/~quesadaj/pdf/LSATutorial.pdf