Machine Learning for NLP

Data preparation and evaluation

Aurélie Herbelot

2022

Centre for Mind/Brain Sciences
University of Trento

Introduction

Building a statistical NLP application (recap)

- Choose your data *according to the task of interest*: e.g. a Wikipedia dump, a collection of novels, etc.
- (Optional) Gather annotation *according to the task of interest*: e.g. manually annotated POS tags, document classes, etc.
- Split the annotated data into training, development and test set.
    - The training data is to ‘learn the rules’.
    - The development data is to tune parameters (if needed).
    - The test data is the unknown set which gives system performance ‘in the real world’.
- Choose appropriate features and extract them from the data.
- Learn, evaluate.

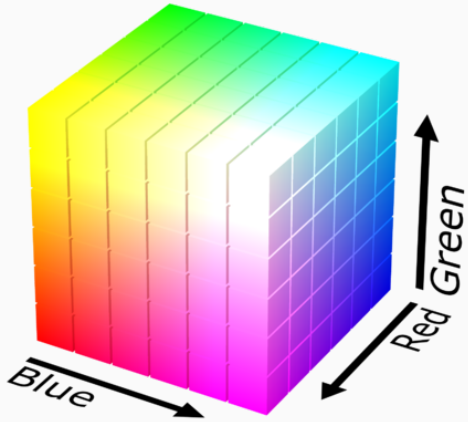
Why is my system not working?

- **Bad data:** the data we are learning from is not the right one for the task.
- **Bad humans:** the quality of the annotation is insufficient.
- **Bad features:** we didn't choose the right features for the task.
- Bad hyperparameters: we didn't tune the learning regime of the algorithm.
- Bad model: the learning algorithm / architecture itself is too dumb.

Bad data

What is data?

- **What is data?** Anything that can be converted into a mathematical representation.
- **What can be converted into maths?** Pretty much anything, as long a there exists a theory specifying how to convert reality into numbers.



By SharkD - CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=3375025

Bad data

- It is not always very clear which data should be used for producing *general* language understanding systems, let alone *domain-specific* applications.
- Some possible issues with data:
  - Domain dependence.
  - Wrong data split.
  - Data size.

The domain dependence issue

- In NLP, the word *domain* usually refers to the kind of data a system is trained/test on (e.g. news, biomedical, novels, tweets, etc).
- When the distribution of the data in the test set is different from that in the training set, our model may fail to perform.
- Domain-specific distributions are sometimes hard to spot. E.g. the English Wikipedia was written by a (mostly) white and (mostly) male population, with regular access to a computer and time to spare. *A general purpose corpus?*

Domain dependence: NER example

- Named Entity Recognition (NER) is the task of recognising and classifying proper names in text:

  [PER] *Trump owns* [LOC] *Mar-a-Lago*.
- NER on specific domains is close to human performance for the task.
- But it is not necessarily easy to port a NER system to a new domain:

  [PER] *Trump cards had been played on both sides.*
  Oops...

Domain dependence: possible solutions

- Annotate more data:
  - training a supervised algorithm necessitates appropriate data;
  - often, such data is obtained via human annotation;
  - so we need new data and new annotations for each new domain.
- Use a **domain adaptation** algorithm. (Not today!)
  Survey on domain adaptation at
  `http://sifaka.cs.uiuc.edu/jiang4/domain_adaptation/survey/da_survey.pdf`

Data presentation issue: ordering

- The ordering of the data will matter when you split it into training and test set.
- Example: you process a corpus of authors' novels. Novels are neatly clustered by authors:
  - 10 novels by Jane Austen, followed by...
  - 10 novels by Agatha Christie, followed by...
  - 10 novels by Steven King.
- If you train on Austen and Christie, and test on King, you end up back with a domain adaptation problem.
- **Always shuffle your data!**

K-fold cross-validation

- A good way to find out whether your data was balanced across splits.
- A good way to know whether you might have just got lucky / unlucky with your test set.
- Let's split our data into k equal $folds = \{D_1, D_2 ... D_k\}$.
- Now train k times on $k - 1$ folds and test on the k^{th} remaining fold.
- Average results.

K-fold cross-validation example

- We have 2000 data points: $\{i_1 \ldots i_{2000}\}$. We decide to split them into 5 folds:
  - Fold 1: $\{i_1 \ldots i_{400}\}$
  - Fold 2: $\{i_{401} \ldots i_{800}\}$
  - ...
  - Fold 5: $\{i_{1601} \ldots i_{2000}\}$
- We train/test 5 times:
  - Train on 2+3+4+5, test on 1. Score: S_1
  - Train on 1+3+4+5, test on 2. Score: S_2
  - ...
  - Train on 1+2+3+4, test on 5. Score: S_5
- Check variance in $\{S_1, S_2, S_3, S_4, S_5\}$, report average.

Leave-one-out

- What to do when the data is too small for K-fold cross-validation, or when you need as much training data as possible?
- **Leave-one-out:** special case of K-fold cross-validation, where the test fold only has one data point in it.

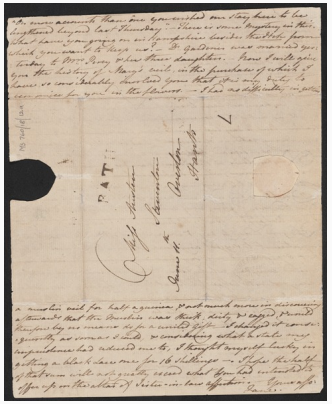
The small data issue

ML algorithms don't work well on small data.
It is a challenge for both science and applications.

Nonce learning

"We found a cute, hairy wampimuk sleeping be- hind the tree."
(Lazaridou et al, 2014)

Digital humanities



Low-resource languages

https://smc.org.in/

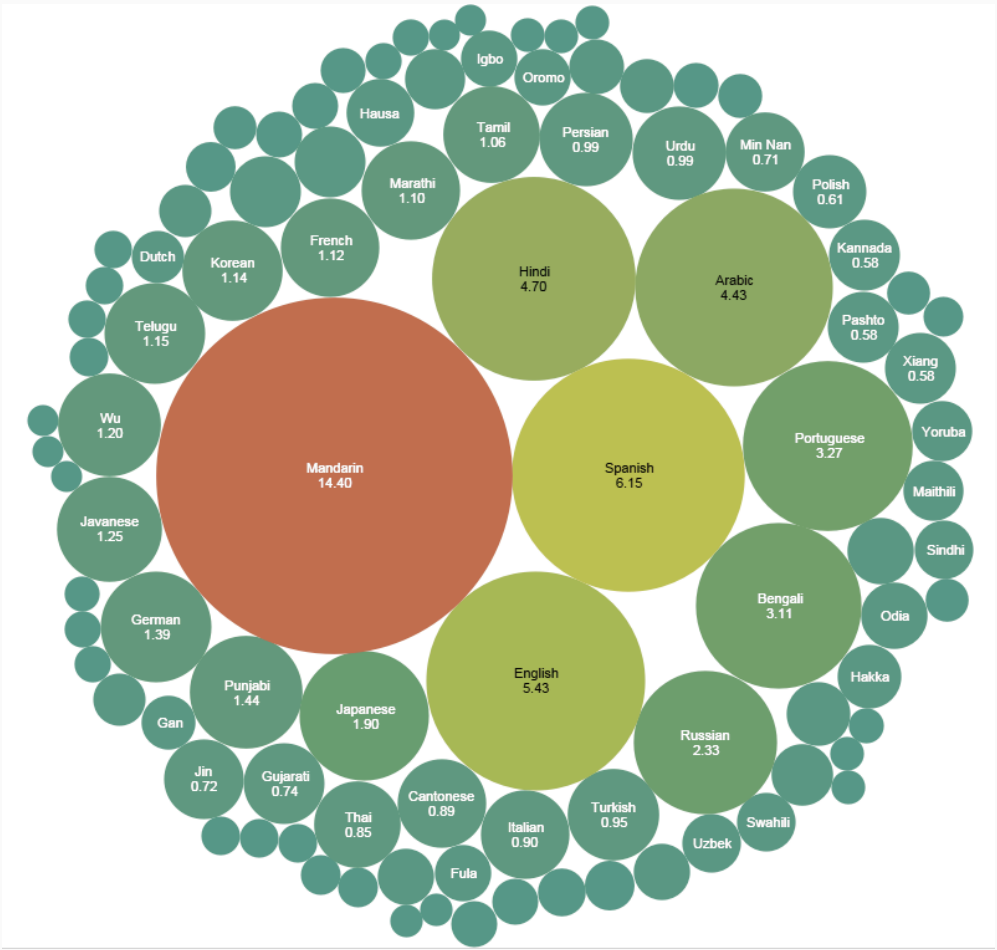
The small data issue

Table 2. Distribution of world languages by number of first-language speakers

Population range	Living languages			Number of speakers		
	Count	*Percent*	*Cumulative*	*Total*	*Percent*	*Cumulative*
100,000,000 to 999,999,999	8	0.1	0.1%	2,709,546,730	40.78777	40.78777%
10,000,000 to 99,999,999	82	1.2	1.3%	2,609,446,190	39.28092	80.06869%
1,000,000 to 9,999,999	307	4.3	5.6%	948,917,508	14.28439	94.35308%
100,000 to 999,999	956	13.5	19.1%	305,209,791	4.59443	98.94751%
10,000 to 99,999	1,811	25.5	44.6%	61,803,881	0.93036	99.87787%
1,000 to 9,999	1,980	27.9	72.5%	7,630,091	0.11486	99.99272%
100 to 999	1,064	15.0	87.4%	470,472	0.00708	99.99981%
10 to 99	329	4.6	92.1%	12,268	0.00018	99.99999%
1 to 9	144	2.0	94.1%	584	0.00001	100.00000%
0	219	3.1	97.2%	0	0.00000	100.00000%
Unknown	199	2.8	100.0%			
Totals	7,099	100.0		6,643,037,515	100.00000	

https://www.ethnologue.com/statistics/size

The small data issue

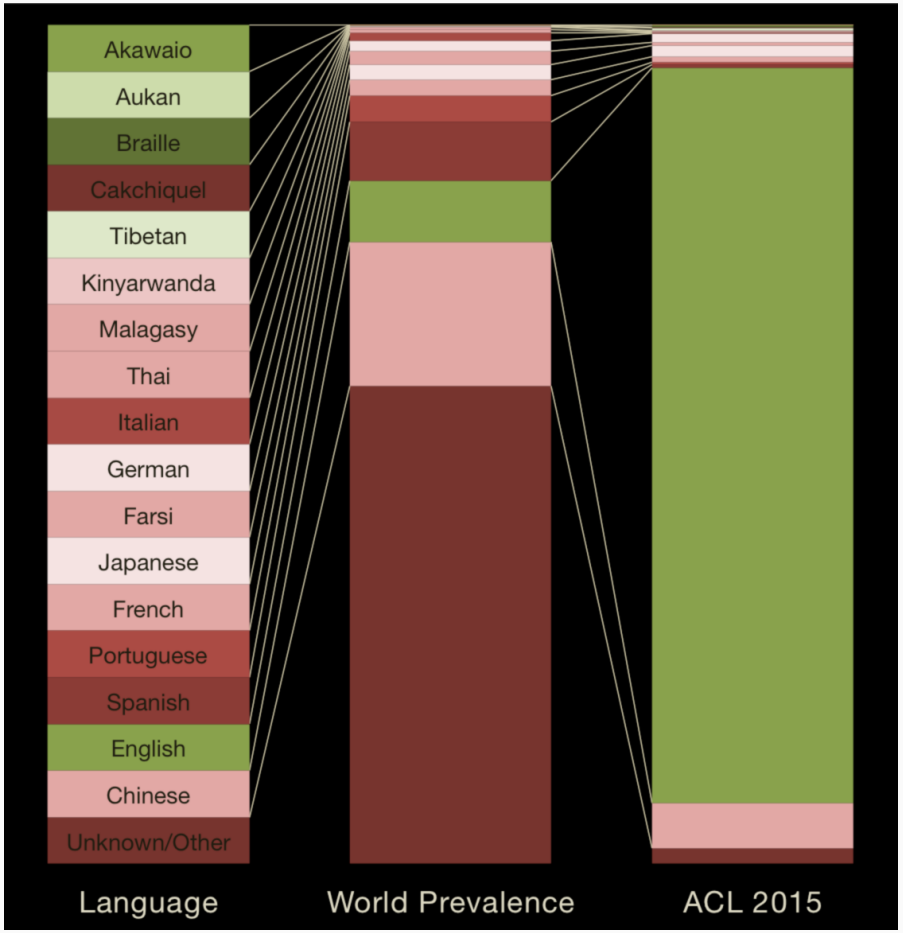


Languages by proportion of native speakers,

https://commons.wikimedia.org/w/index.php?curid=41715483

NLP for the languages of the world

- The ACL is the most prestigious computational linguistic conference, reporting on the latest developments in the field.
- How does it cater for the languages of the world?

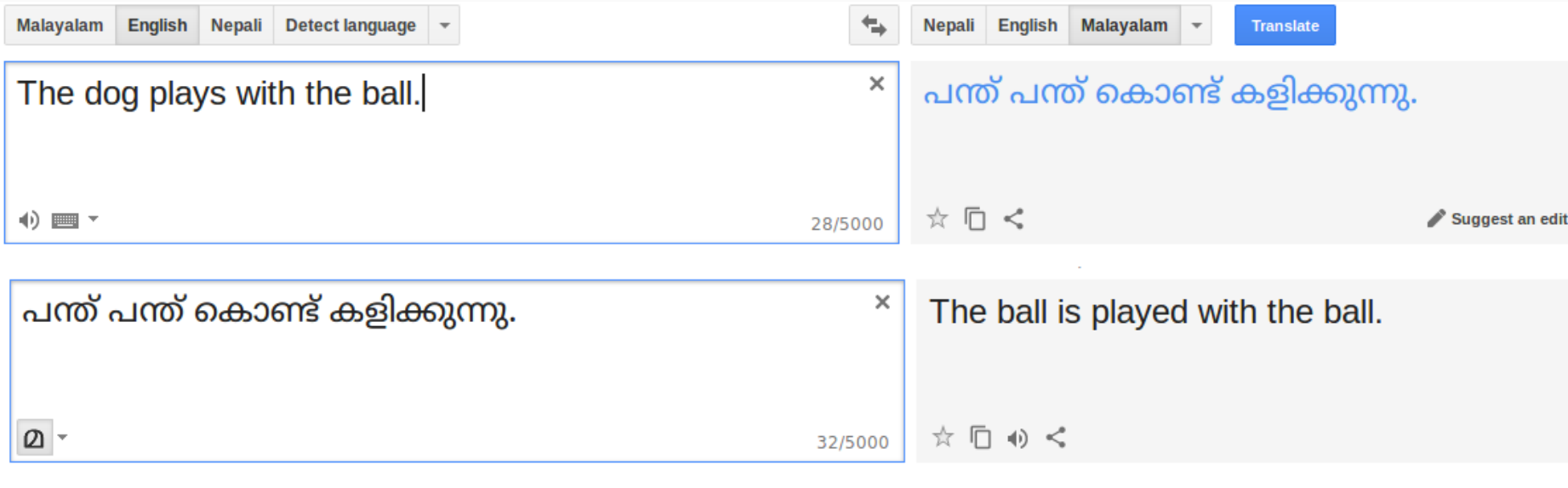


http://www.junglelightspeed.com/languages-at-acl-this-year/

NLP research and low-resource languages (Robert Munro)

- ‘In order to understand how computational linguistics applies to the full breadth of human communication, we need to test the technology across a representative diversity of languages.’
- ‘For vocabulary, word-order, morphology, standardized spelling, and more, English is an outlier, telling little about how well a result applies to the 95% of the world’s communications that are in other languages.’
- Check out the #BenderRule: https://thegradient.pub/the-benderrule-on-naming-the-languages-we-study-and-why-it-matters/

Google translate: English <–> Malayalam



Solutions?

- The ‘small data’ issue is one of the least understood problems in AI.
- Just showing that AI is not that ‘intelligent’ yet.
- Solutions to the small data issue:
    - **Projection methods:** transfer knowledge from a well-known language to a low-resource one.
    - **One-shot learning:** if the new data can be expressed in terms of the features of the old data, use similarity to find best match on the old data.
    - **Transfer learning:** more generically, use an already trained model to start off training on a new task.

Bad humans

Annotation

- The process of obtaining a gold standard from human subjects, for a system to be trained and tested on.
- An *annotation scheme* is used to tell humans what their exact task is.
- A good annotation scheme will:
  - remove any possible ambiguity in the task description;
  - be easy to follow.

Bad humans

- The annotation process should be followed by a validation of the quality of the annotation.
- The assumption is that the more *agreement* we have, the better the data is.
- *The* reference on human agreement measures for NLP: `http://dces.essex.ac.uk/technical-reports/2005/csm-437.pdf`.

Bad measures of agreement

- We have seen that when evaluating a system, not every performance metric is suitable.
- Remember: if the data is biased and a system can achieve reasonable performance by always predicting the most frequent class, we should not report accuracy.
- This is the same for the evaluation of human agreement.

Percentage of agreement

- The simplest measure: the percentage of data points on which two coders agree.
- The agreement value agr_i for datapoint i is:
  - 1 is the two coders assign i to the same class;
  - 0 otherwise;
- The overall agreement figure is then simply the mean of all agreement values:

$$A_o = \frac{1}{i} \sum_{i \in I} agr_i$$

Percentage of agreement - example

		Coder A		
		Stat	IReq	Total
Coder B	Stat	20 (.2)	20 (.2)	40 (.4)
	IReq	10 (.1)	50 (.5)	60 (.6)
	Total	30 (.3)	70 (.7)	100 (1)

Table 1: A simple example of agreement on dialogue act tagging

The percentage agreement here is:

$$A_o = (20 + 50)/100 = 0.7$$

Percentage of agreement - problems

- If the classes are imbalanced, chance agreement will be inflated.
- Example:
  - 95% of utterances in a domain are of class A and 5% of class B.
  - By chance, the agreement will be $0.95 \times 0.95 + 0.05 \times 0.05$, i.e. 90.5%.
    (The chance of class A being chosen by both annotators is 0.95×0.95, and the chance of class B being chosen by both annotators is 0.05×0.05.)

Percentage of agreement - problems

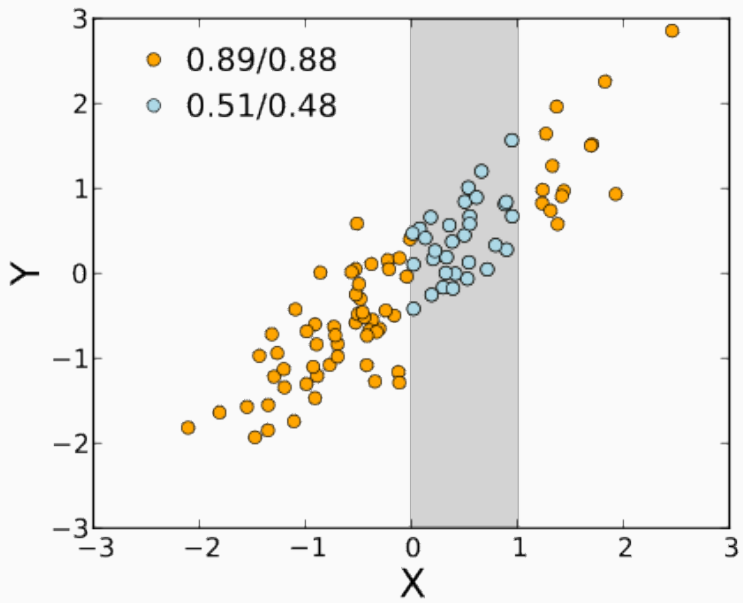
- Given two coding schemes, the one with fewer categories will have a higher percentage of agreement just by chance.
- Example:
  - 2 categories: the percentage of agreement *by chance* will be $(\frac{1}{2} \times \frac{1}{2} + \frac{1}{2} \times \frac{1}{2}) = 0.5$.
  - 3 categories: the percentage of agreement *by chance* will be $(\frac{1}{3} \times \frac{1}{3} + \frac{1}{3} \times \frac{1}{3} + \frac{1}{3} \times \frac{1}{3}) = 0.33$.

Correlation

- Correlation measures the dependence of one variable's values upon another.
- E.g. Pearson correlation:

$$\rho_{X,Y} = \frac{\mathrm{cov}(X, Y)}{\sigma_X \sigma_Y}$$

with σ the standard deviation and cov the covariance.



By Skbkekas - Own work, CC BY 3.0, https://commons.wikimedia.org/w/index.php?curid=9362598

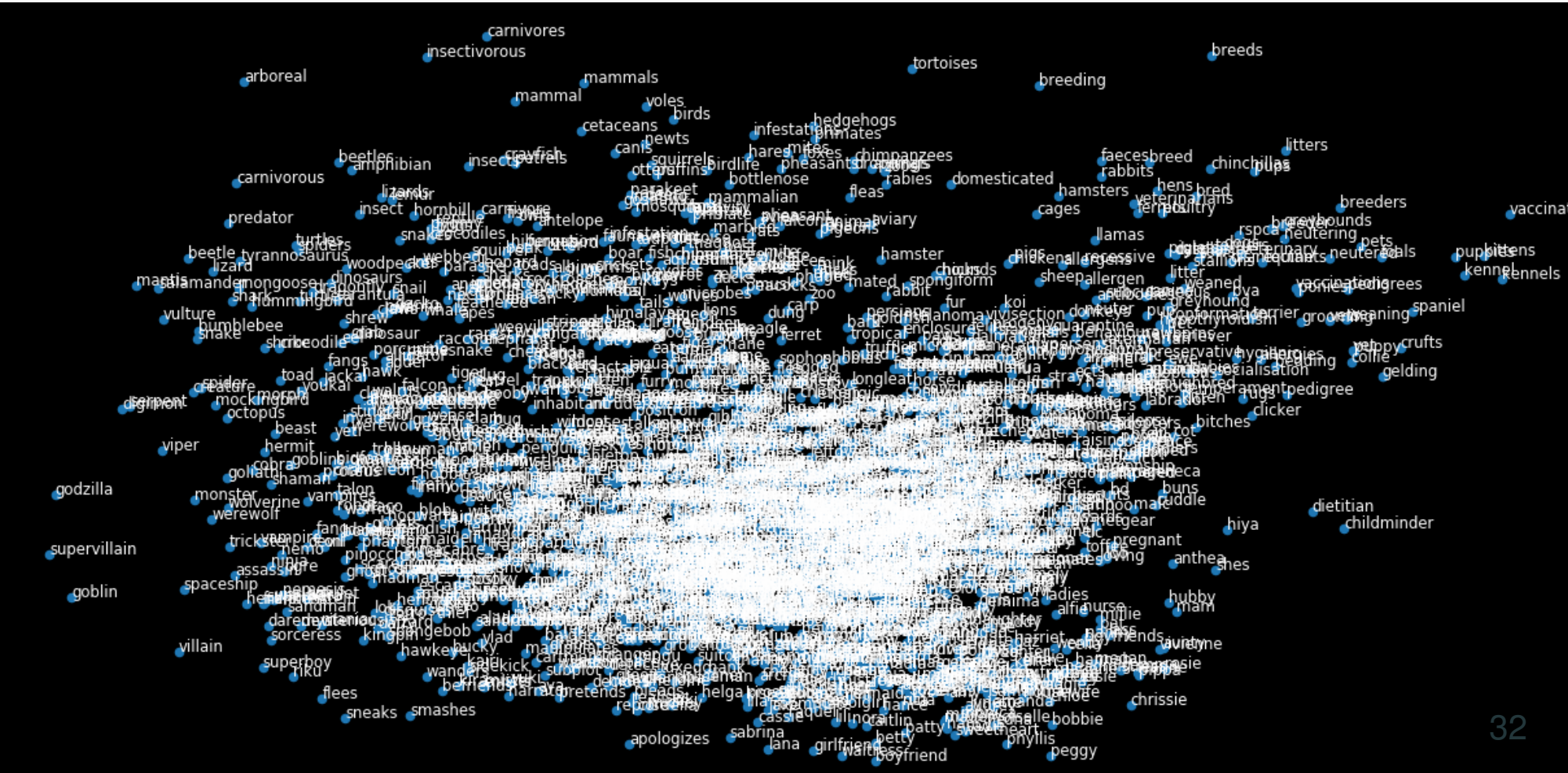
Correlation - problem

- Two sets of annotations can be correlated without there being agreement between the coders.
- Suppose a marking scheme where two coders must give a mark between 1 and 10 to student essays.

Item	Exp 1		Exp 2	
	A	B	C	D
a	1	1	1	2
b	2	2	2	4
c	3	3	3	6
d	4	4	4	8
e	5	5	5	10
	$r = 1.0$		$r = 1.0$	

Correlation - okay

- Correlation is however fine to use if only rank matters.
- **Example:** can we produce a distributional semantics system that models human similarity judgments?



Similarity-based evaluation with correlation

Human output

```
sun sunlight 50.000000
automobile car 50.000000
river water 49.000000
stair staircase 49.000000
...
green lantern 18.000000
painting work 18.000000
pigeon round 18.000000
...
muscle tulip 1.000000
bikini pizza 1.000000
bakery zebra 0.000000
```

System output

```
stair staircase 0.913251552368
sun sunlight 0.727390960465
automobile car 0.740681924959
river water 0.501849324363
...
painting work 0.448091435945
green lantern 0.383044261062
...
bakery zebra 0.061804313745
bikini pizza 0.056135605632 3
pigeon round 0.028243620524
muscle tulip 0.014257083536 7
```

Cohen's Kappa

- Cohen's Kappa defines a measure of agreement *above chance*:

$$\kappa = \frac{p_o - p_e}{1 - p_e}$$

- It is used for *nominal scales* rather than *numerical scales* (i.e. classification problems rather than problems where real values are elicited from annotators).

Cohen's Kappa example

	Class A	Class B	Totals
Class A	15	5	**20**
Class B	10	70	**80**
Totals	**25**	**75**	**100**

$$p_e = [(20/100) * (25/100)] + [(75/100) * (80/100)] = 0.05 + 0.60 = 0.65$$

Cohen's Kappa example

	Class A	Class B	Totals
Class A	**15**	5	20
Class B	10	**70**	80
Totals	25	75	**100**

$$p_o = (15 + 70)/100 = 0.85$$

$$\kappa = \frac{p_o - p_e}{1 - p_e} = \frac{0.85 - 0.65}{1 - 0.65} = 0.57$$

Kappa's interpretation

- What is a good enough kappa?
- Very unclear. Interpretation schemes have been proposed but don't take into account various properties of kappa.
- For instance, κ becomes higher when more classes are considered.

Interpretation of Kappa

	Poor	Slight	Fair	Moderate	Substantial	Almost perfect
Kappa	0.0	.20	.40	.60	.80	1.0

Kappa	Agreement
< 0	Less than chance agreement
0.01–0.20	Slight agreement
0.21– 0.40	Fair agreement
0.41–0.60	Moderate agreement
0.61–0.80	Substantial agreement
0.81–0.99	Almost perfect agreement

Landis & Koch (1977). Figure from Viera & Garrett (2005).

Agreement for several coders

- What to do when we have more than two coders?
- We can simply report the mean and variance of all pairs of coders. For instance, with three coders A_1, A_2 and A_3:
  - $\bar{\kappa} = \frac{\kappa(A_1,A_2)+\kappa(A_1,A_3)+\kappa(A_2,A_3)}{3}$
- There are also measures specific to multi-coder cases (see Fleiss' Kappa).

Intra-annotator agreement

- Sometimes useful to measure *intra*-annotator agreement!
- Ask the same coder to perform the annotation twice, at a few weeks' interval. How likely is the coder to significantly agree with themselves?

How many coders should I have?

- As for any data, the more the better...
- At least three!
- Unfortunately, human annotation is very expensive. So a trade-off is usually needed.

Where does low agreement come from?

- The guidelines were bad. Compare:
  - How similar are *cat* and *dog*? (1-7)
  - Is *cat* more similar to *dog* or to *horse?*
- The task is hard: it requires access to knowledge that is normally unconscious, or too much interpretation.
  - Quantify the following with *no, few, some, most, all*:
    - ___ bathtubs are white
    - ___ trumpets are loud

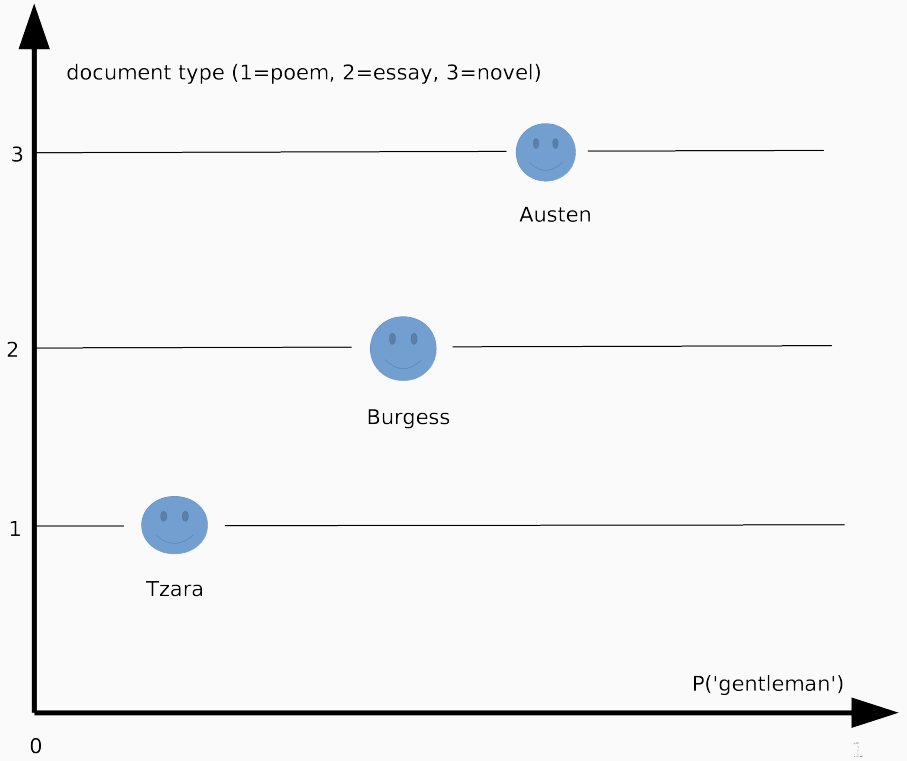
Never trust humans to do what you want...

Predication type	Example	Prevalence
Principled	Dogs have tails	92%
Quasi-definitional	Triangles have three sides	92%
Majority	Cars have radios	70%
Minority characteristic	Lions have manes	64%
High-prevalence	Canadians are right-handed	60%
Striking	Pit bulls maul children	33%
Low-prevalence	Rooms are round	17%
False-as-existentials	Sharks have wings	5%

Table 1: Classes of generic statements with associated prevalence, as per Khemlani et al (2009).

Bad features

Explicit vs latent features



Here is a hypothetical feature space to distinguish between authors (simplified to only show two dimensions!)
The features in this space are explicit: *P*(*gentleman*), document type.
Do we always have such explicit features?

Features in cognition

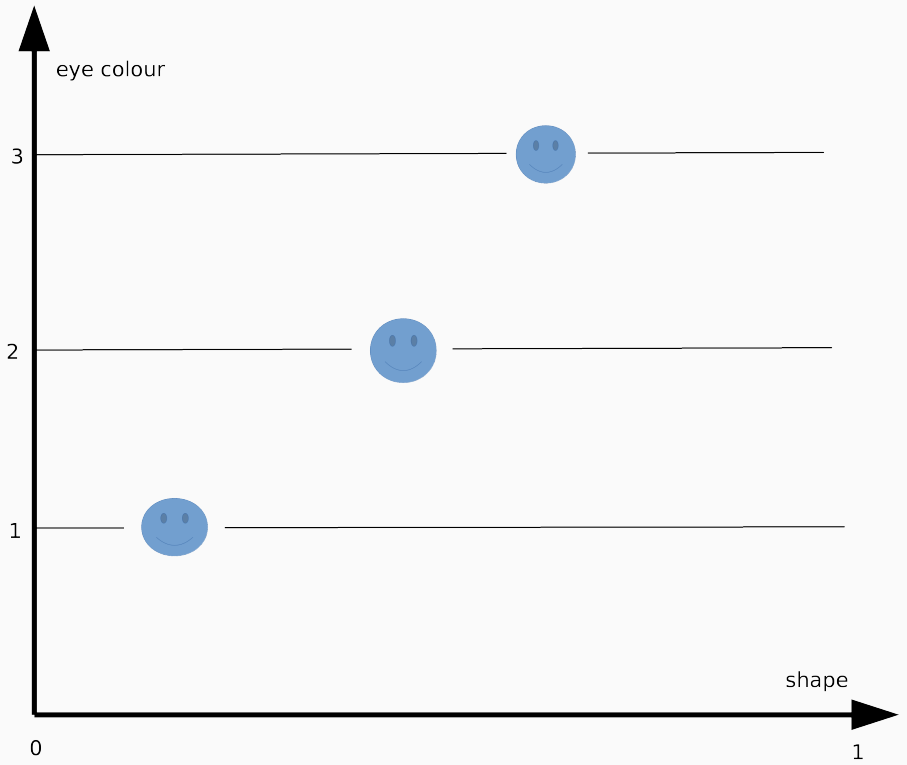
- We said that features are aspects of the data that may be relevant for a task.
- Which features do you use to recognise a face?

Gao et al (2017)

Features in cognition

- We could say that recognising a face involves classifying the shape and colour of someone's eyes.
- That is two features. We need to define them a little more:
  - The shape of the eyes will be given by the average (you have two eyes!) of the ratios of width and height. This is a number between 0 and 1.
  - For the eyes' colour, we will simplify to a number of classes: blue (class 1), green (class 2), brown (class 3) eyes. That is a number in the set $\{1, 2, 3\}$.

Features in cognition

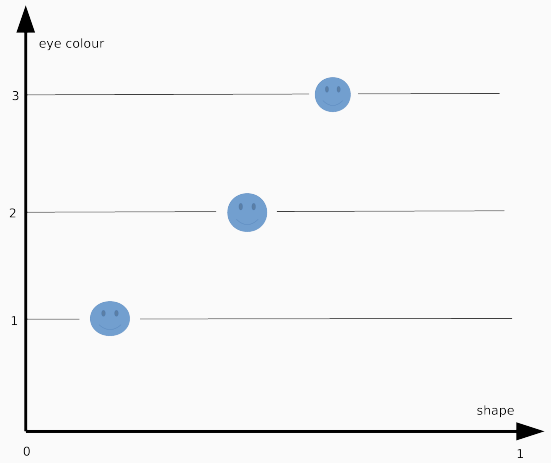


Eye shape and colour are *dimensions* (axes) in a vector space.

Each point in the space represents a potential person.

Latent features

- This is not very realistic. In fact, we do not know what features our perceptual system selects for face recognition.
- Neural net simulations of the human visual system suggest that individual neurons specialise in recognising low-level features such as curves of a certain shape, as well as combinations thereof.
- Such features, obtained by manipulation of the raw signal, are called *latent* features.



Explicit feature selection

- The process of automatically selecting a subset of the possible features occurring before training, to weed out irrelevant or redundant properties of the data.
- **Feature selection** avoids two common problems which can impact on system performance:
  - the curse of dimensionality;
  - overfitting.
- Typically, it also speeds up the system. (Less features means less training time.)

The curse of dimensionality

Kevin Lacker, 2x Putnam fellow

Updated May 25, 2011 · Upvoted by Peter Norvig, Research Director at Google and Jay Verkuilen, PhD Psychometrics, MS Mathematical Statistics, UIUC · Author has **185** answers and **405.7k** answer views

Let's say you have a straight line 100 yards long and you dropped a penny somewhere on it. It wouldn't be too hard to find. You walk along the line and it takes two minutes.

Now let's say you have a square 100 yards on each side and you dropped a penny somewhere on it. It would be pretty hard, like searching across two football fields stuck together. It could take days.

Now a cube 100 yards across. That's like searching a 30-story building the size of a football stadium. Ugh.

The curse of dimensionality

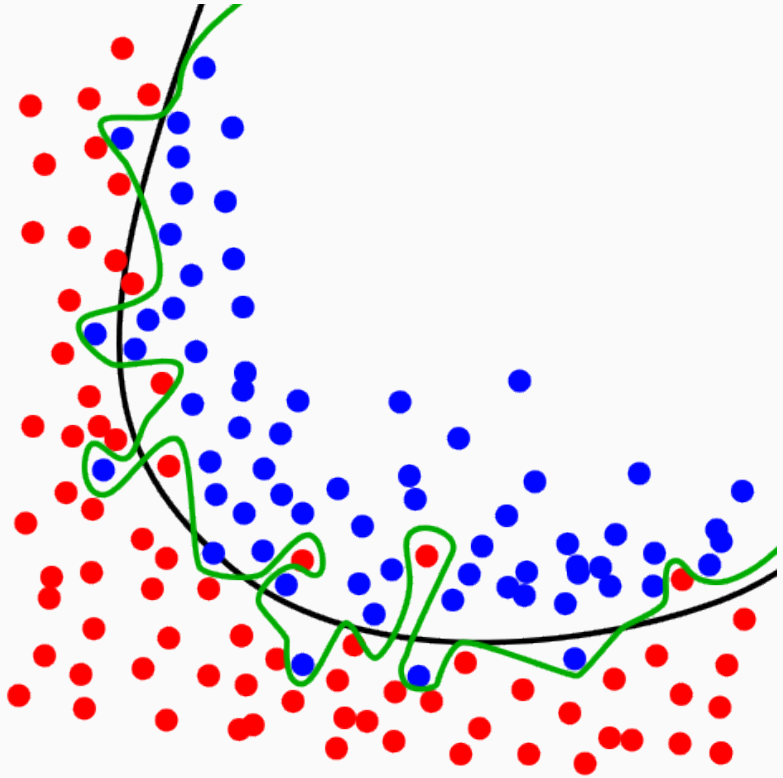
- Say we are training a classifier that needs to distinguish between 10 different possible values of 1000 features to perform well.
- Each class is represented by a set of feature combinations to be found out of 10^{1000} such combinations...

The curse of dimensionality

- This said... dimensionality is not the only issue. In fact, a high-dimensional space may be okay if datapoints are nicely clustered in it.
- The number of training examples needed is a function of the number of regions that must be distinguished in space.

Overfitting

- Overfitting is producing a model that is too close to the data and won't generalise well on new data.
- Typically, an overfitted model has too many parameters / features than is justified given the training data.



By Chabacano - Own work, GFDL, https://commons.wikimedia.org/w/index.php?curid=3610704

Overfitting example

- An authorship attribution system has learnt a ‘signature’ for recognising J.K. Rowling’s work from training on Harry Potter.
- Particularly important features ended up being:
  - the proper nouns *Hermione* and *Hagrid*;
  - the bigram *quidditch game*.
- It now fails to recognise *The Casual Vacancy*, a novel by J.K. Rowling with no relation to the Harry Potter universe.

Feature selection in NLP

- In machine learning overall, feature selection can be achieved using many different techniques.
- In NLP specifically, three methods are commonly used:
  - Raw frequency.
  - TF-IDF.
  - Mutual Information.

Feature selection by frequency

- A very simple method when we anyway have many features.
- Just select the *n* most frequent words of a class / of the entire vocabulary.
- Often, the selected words won't be directly related to the nature of the class:
  - *Monday, Tuesday...* for news text;
  - *x, y* for maths texts.

Feature selection with TF-IDF

- The measure that powered search engines *for years*. Used to measure the importance of a term to a document (or to a class).
- **TF:** term frequency. This is simply the relative frequency of the term in a given document.
- **IDF:** inverse document frequency. This is the log of the inverse fraction of the documents containing the term.

$$tf(t, d) = \frac{f(t)}{N_d}$$

with N_d the number of words in d.

$$idf(t) = log \frac{|D|}{|d \in D : t \in d|}$$

with D the number of documents in the collection.

Feature selection by Mutual information

- The *mutual information (MI)* of word w and class c.
- MI measures how much information the presence or absence of a word contributes to correctly classifying a text in class c.
- MI is calculated as:

$$\sum_{c \in \{0,1\}} \sum_{w \in \{0,1\}} p(w,c) \log \left(\frac{p(w,c)}{p(w)\,p(c)} \right)$$

We are summing over documents that are or are not in class c, and over words w that are or are not in the document.

Mutual information example

	$w \in$ Doc	$w \notin$ Doc	Totals
#Doc $\in$ class A	49	27652	27701
#Doc $\notin$ class A	141	774106	774247
Totals	190	801758	801948

$$
\begin{aligned}
I(U;C) &= \frac{49}{801{,}948}\log_2\frac{801{,}948\cdot 49}{(49+27{,}652)(49+141)} \\
&\quad + \frac{141}{801{,}948}\log_2\frac{801{,}948\cdot 141}{(141+774{,}106)(49+141)} \\
&\quad + \frac{27{,}652}{801{,}948}\log_2\frac{801{,}948\cdot 27{,}652}{(49+27{,}652)(27{,}652+774{,}106)} \\
&\quad + \frac{774{,}106}{801{,}948}\log_2\frac{801{,}948\cdot 774{,}106}{(141+774{,}106)(27{,}652+774{,}106)} \\
&\approx 0.0001105
\end{aligned}
$$

https://nlp.stanford.edu/IR-book/html/htmledition/mutual-information-1.html

Mutual information example

UK	
london	0.1925
uk	0.0755
british	0.0596
stg	0.0555
britain	0.0469
plc	0.0357
england	0.0238
pence	0.0212
pounds	0.0149
english	0.0126

China	
china	0.0997
chinese	0.0523
beijing	0.0444
yuan	0.0344
shanghai	0.0292
hong	0.0198
kong	0.0195
xinhua	0.0155
province	0.0117
taiwan	0.0108

poultry	
poultry	0.0013
meat	0.0008
chicken	0.0006
agriculture	0.0005
avian	0.0004
broiler	0.0003
veterinary	0.0003
birds	0.0003
inspection	0.0003
pathogenic	0.0003

coffee	
coffee	0.0111
bags	0.0042
growers	0.0025
kg	0.0019
colombia	0.0018
brazil	0.0016
export	0.0014
exporters	0.0013
exports	0.0013
crop	0.0012

elections	
election	0.0519
elections	0.0342
polls	0.0339
voters	0.0315
party	0.0303
vote	0.0299
poll	0.0225
candidate	0.0202
campaign	0.0202
democratic	0.0198

sports	
soccer	0.0681
cup	0.0515
match	0.0441
matches	0.0408
played	0.0388
league	0.0386
beat	0.0301
game	0.0299
games	0.0284
team	0.0264

► **Figure 13.7** Features with high mutual information scores for six Reuters-RCV1 classes.

https://nlp.stanford.edu/IR-book/html/htmledition/mutual-information-1.html