# Naive Bayes

Abram Hindle

University of California, Davis
Davis, California
http://softwareprocess.es/

abram.hindle@softwareprocess.es

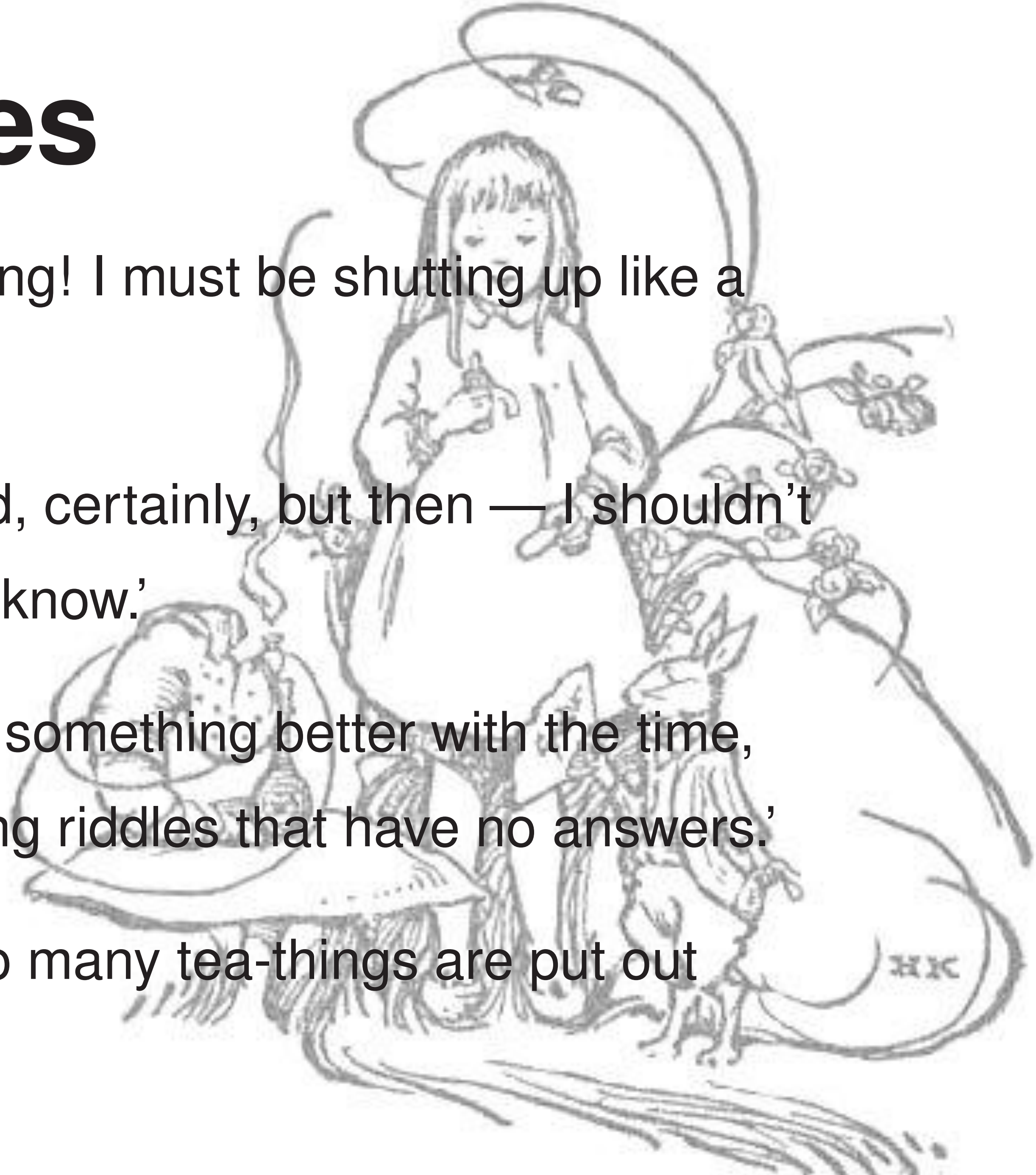
# Machine Learning

- Teaching a computer to make decisions
- Teaching a computer to learn about data
- Like AI except specific to data-mining
- Usually based on analyzing data
- Supervised learning
  - Manually annotated training data often to classify.
- Unsupervised
  - Automatic discovery of properties used to describe data.

# Supervised Learning

- We'll focus on classification
- We want to say an entity belongs to a class
  - Spam / Ham or On-topic / off-topic
- We need:
  - A learning algorithm to learn properties associated with labels
  - A training set – manually annotated examples used to learn from
  - A test set – manually annotated examples used to validate performance

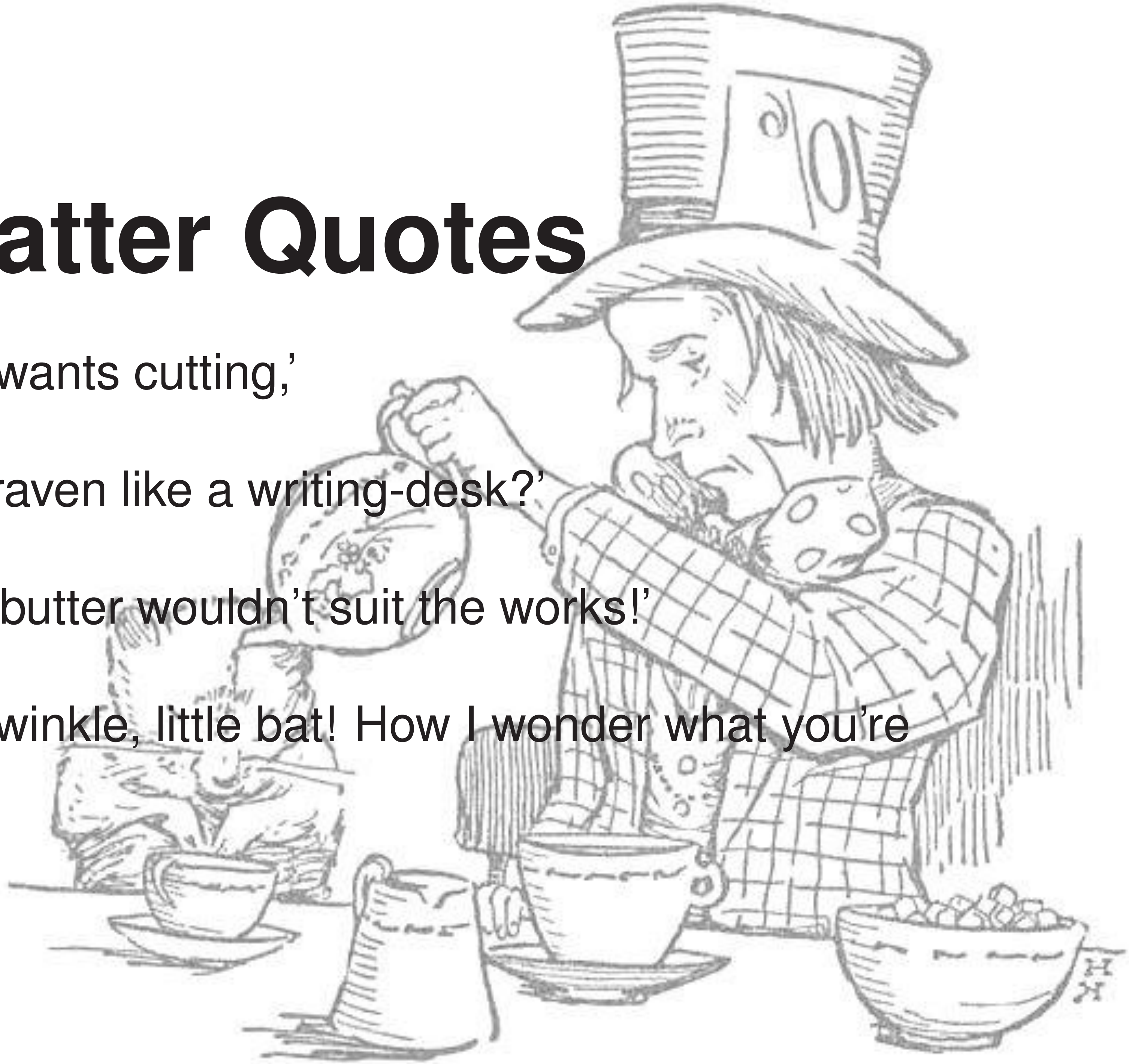
# Alice Quotes

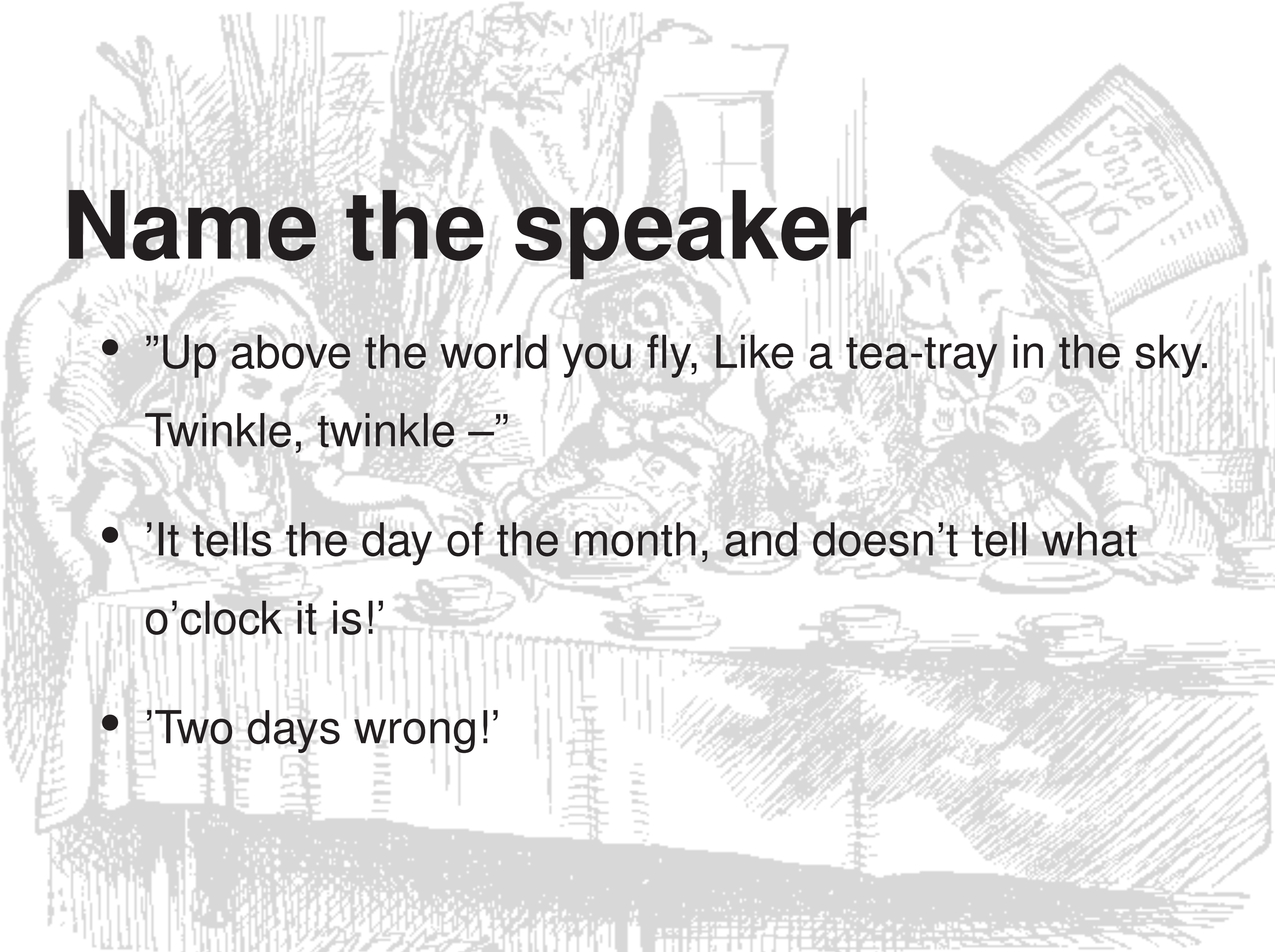


- ’What a curious feeling! I must be shutting up like a telescope.’
- ’That would be grand, certainly, but then — I shouldn’t be hungry for it, you know.’
- ’I think you might do something better with the time, than waste it in asking riddles that have no answers.’
- ’Is that the reason so many tea-things are put out here?’

# Mad Hatter Quotes

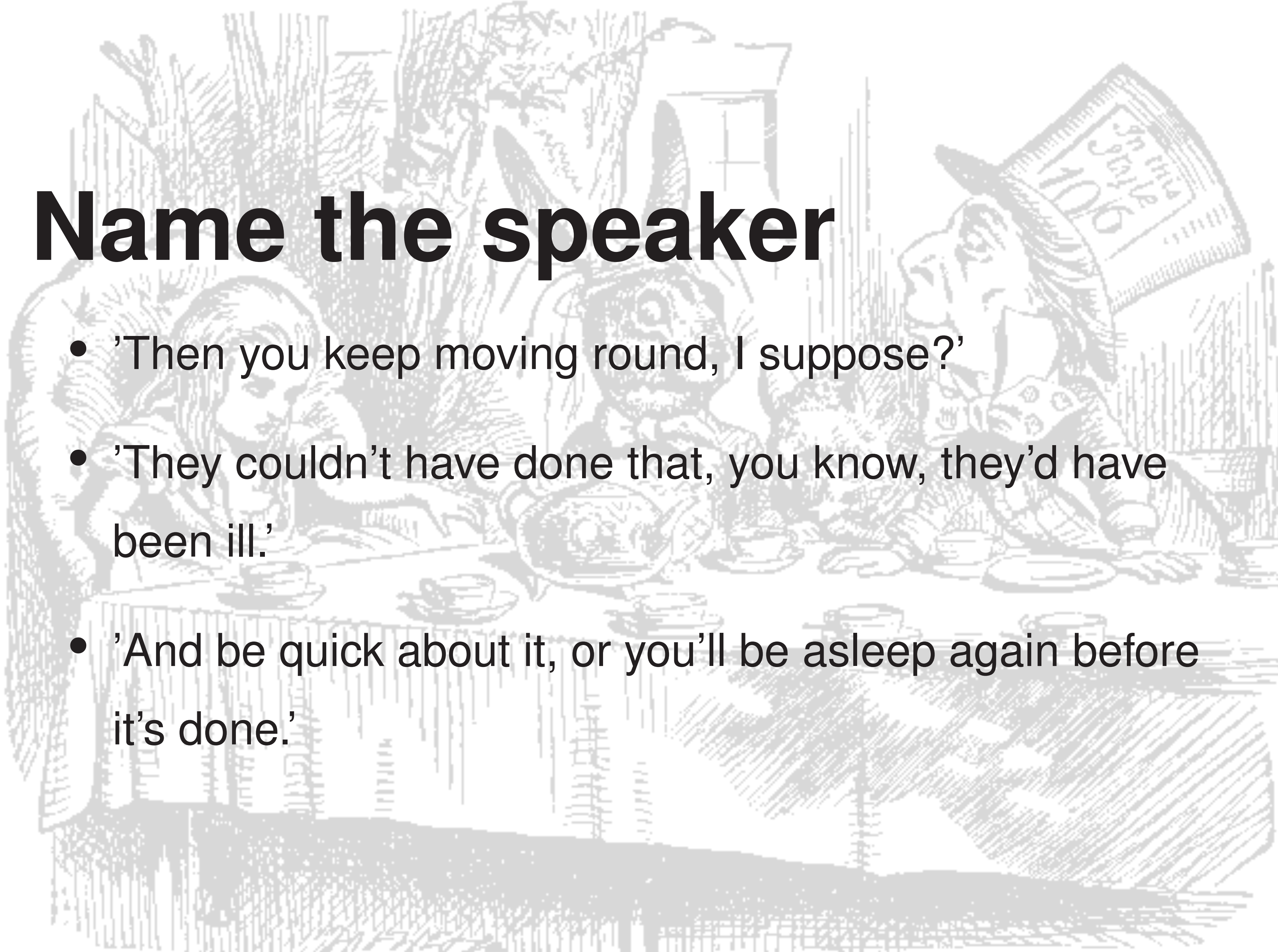
- ’Your hair wants cutting,’
- ’Why is a raven like a writing-desk?’
- ’I told you butter wouldn’t suit the works!’
- ”Twinkle, twinkle, little bat! How I wonder what you’re at!”





# Name the speaker

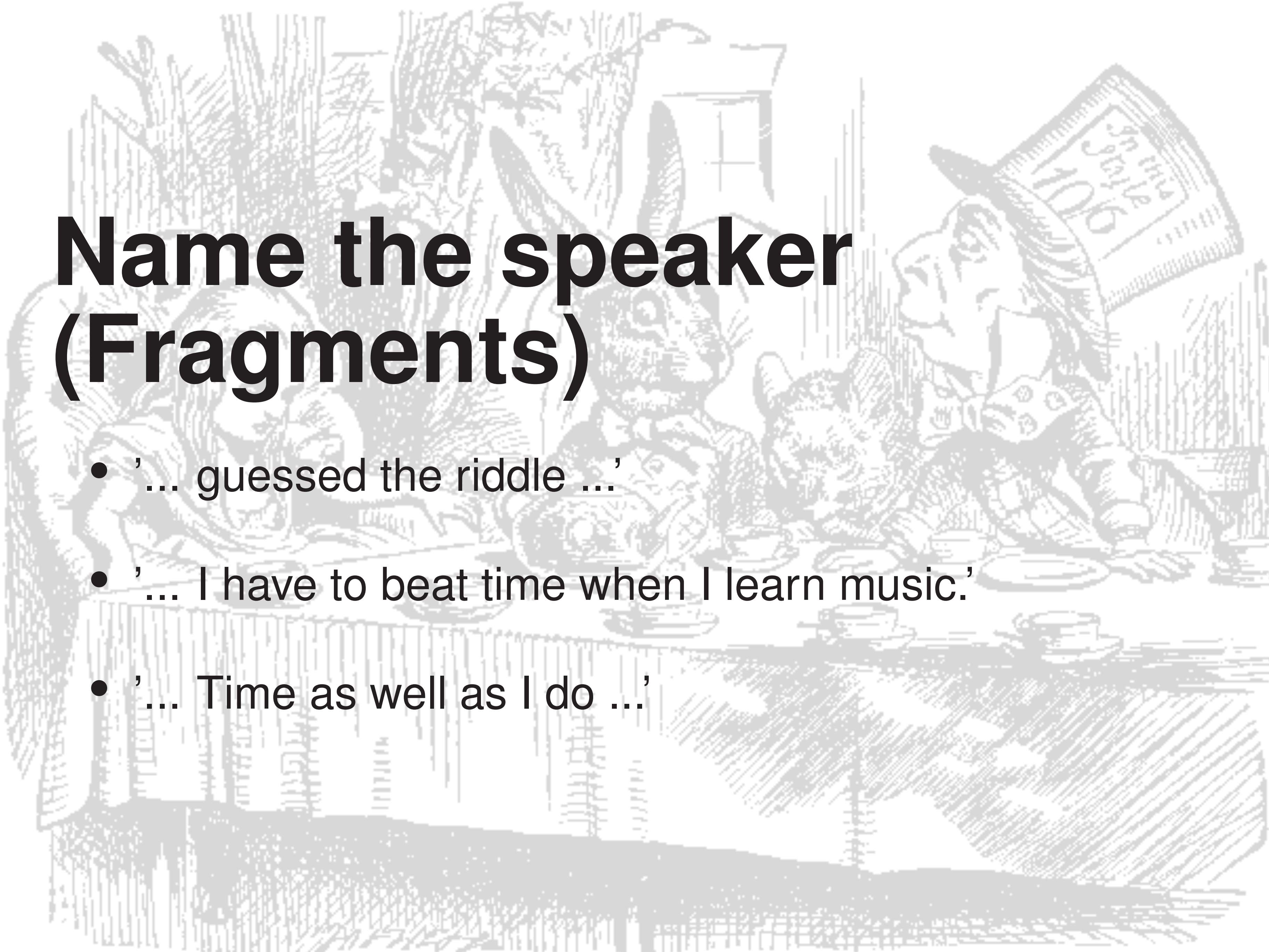
- ”Up above the world you fly, Like a tea-tray in the sky. Twinkle, twinkle –”
- ’It tells the day of the month, and doesn’t tell what o’clock it is!’
- ’Two days wrong!’



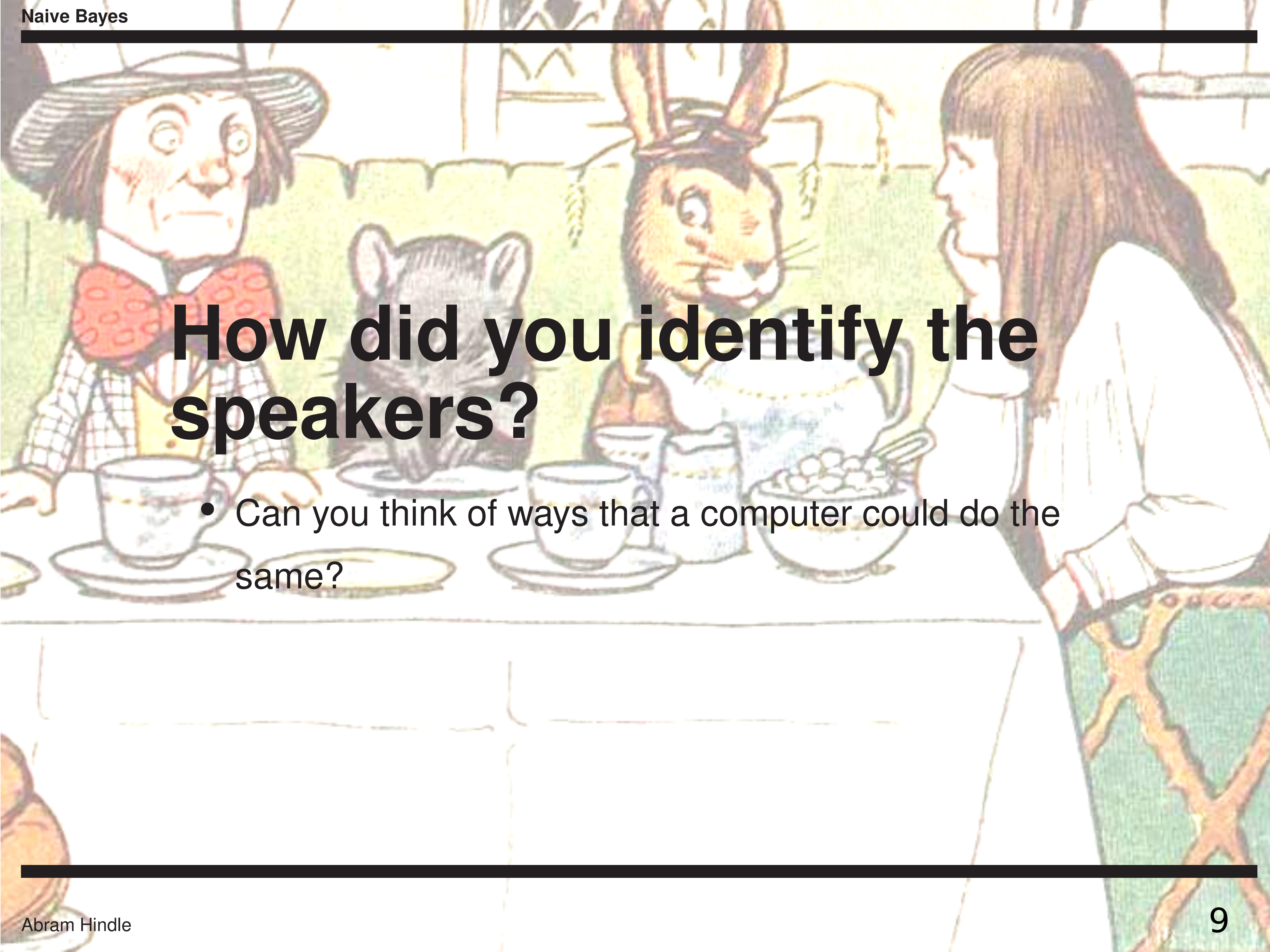
# Name the speaker

- 'Then you keep moving round, I suppose?'
- 'They couldn't have done that, you know, they'd have been ill.'
- 'And be quick about it, or you'll be asleep again before it's done.'

# Name the speaker (Fragments)



- '... guessed the riddle ...'
- '... I have to beat time when I learn music.'
- '... Time as well as I do ...'



# How did you identify the speakers?

- Can you think of ways that a computer could do the same?

# How can we describe these quotations to a program?

- Features and Feature vectors
  - Features - measurable aspects of a sample
  - Feature vector - features normalized into a vector form
    - Easy to summarize a single entity

# Discussion

- What kinds of features can we tease from text?

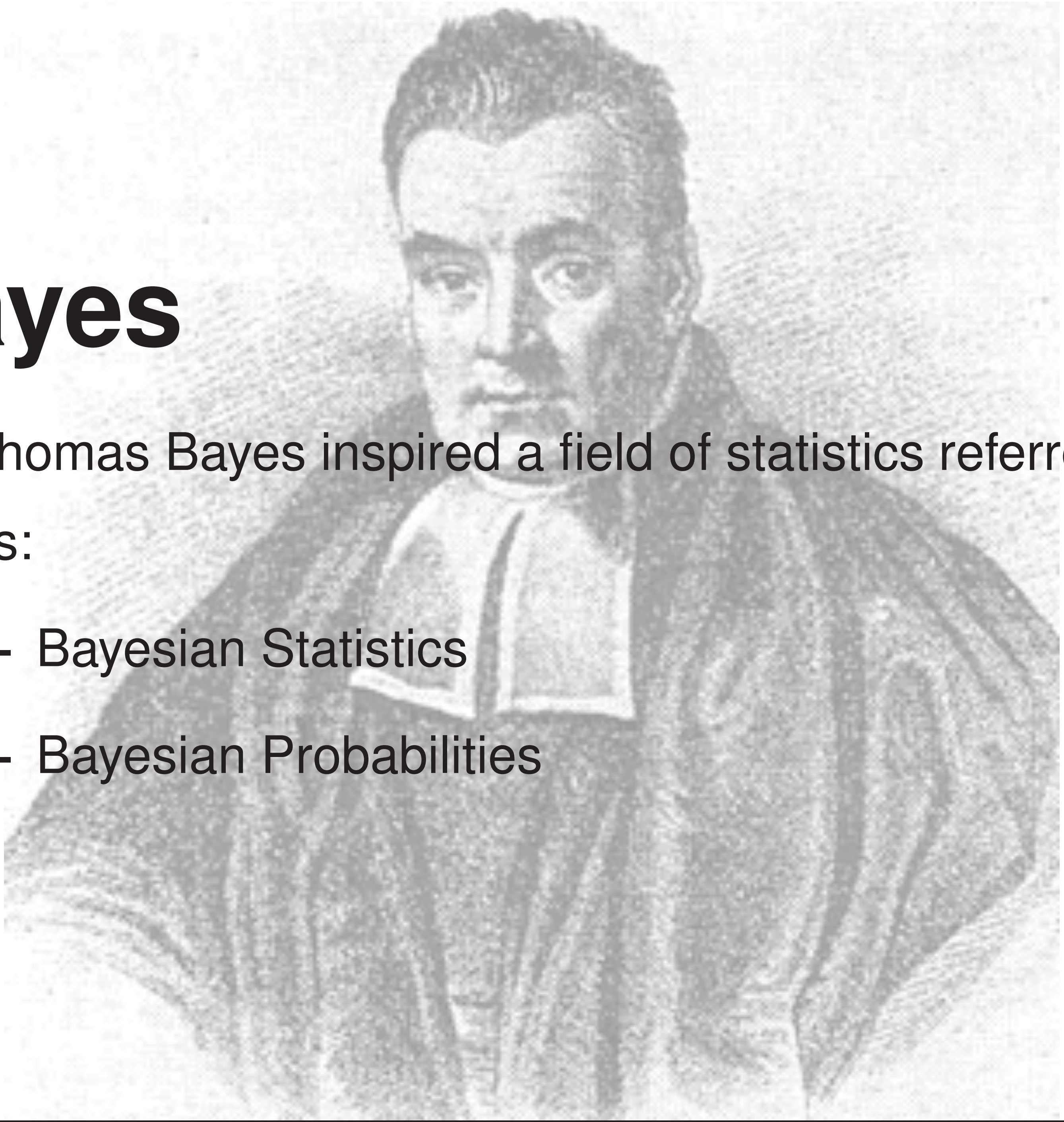
# Discussion

- What kinds of features can we tease from text?
  - Words
  - Characters
  - substrings
  - n-grams ( strings of tokens )
  - typography

# Word Counts

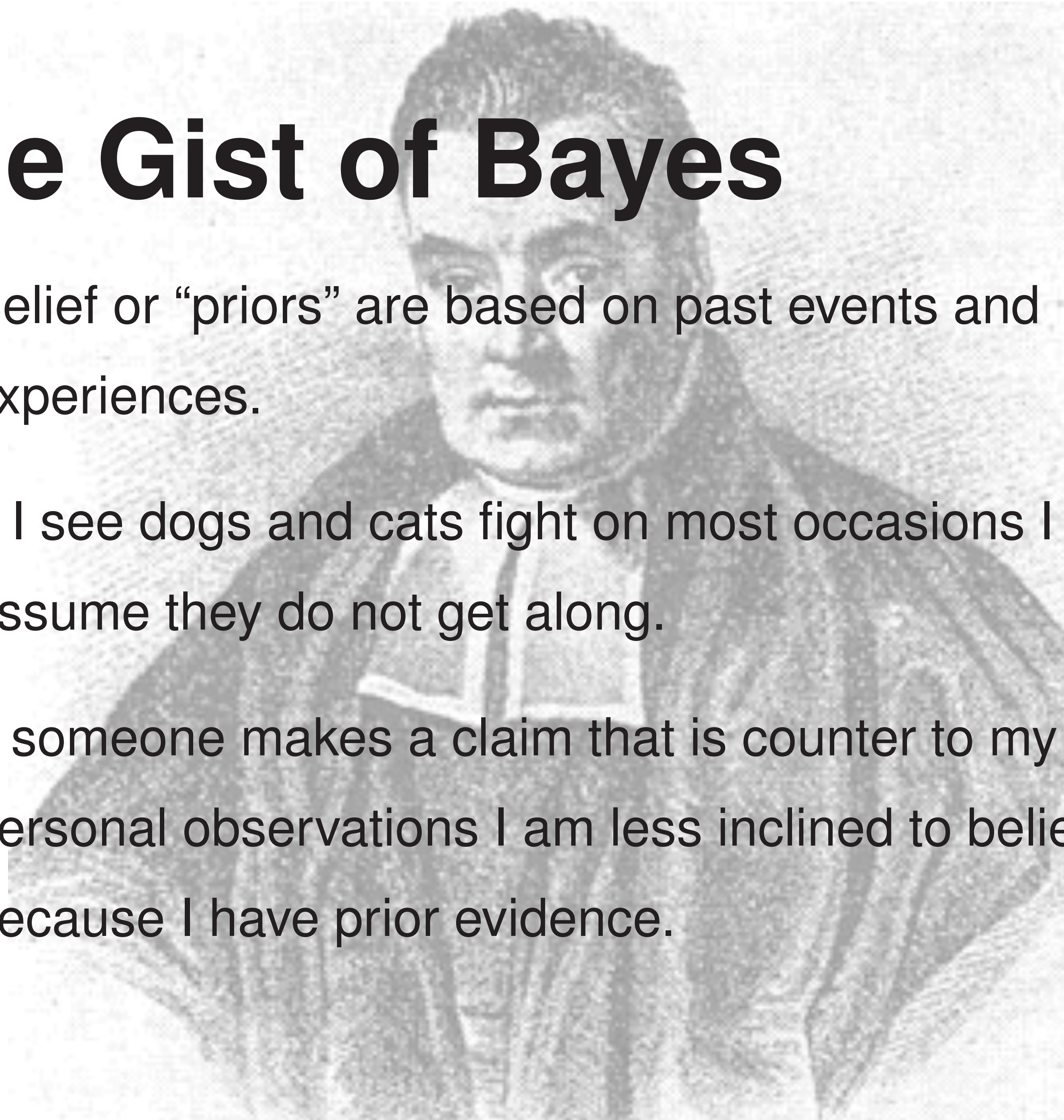
- Word counts of the Mad Hatter and Alice

| Word | Mad Hatter | Alice |
|---|---|---|
| I | 9 | 14 |
| you | 16 | 5 |
| twinkle | 4 | 0 |
| clock | 3 | 1 |
| now | 1 | 3 |
| in | 3 | 3 |
| Total | 36 | 26 |



# Bayes

- Thomas Bayes inspired a field of statistics referred to as:
  - Bayesian Statistics
  - Bayesian Probabilities



# The Gist of Bayes

- Belief or “priors” are based on past events and experiences.
- If I see dogs and cats fight on most occasions I will assume they do not get along.
- If someone makes a claim that is counter to my personal observations I am less inclined to believe it because I have prior evidence.

# Naive Bayes for documents

- We assume all features are independent
  - no dependant probabilities between features.
- Classifier:
  $classify(D) = \arg\max P(C) \prod_i P(w_i|C)$
- 
  - Return the class $C$ whose product of word ($w_i$) probabilities is the greatest.
  - Note lack of dependence between the words!
  - Also considered to be Maximum Likelihood Estimation (MLE)

# Documents!

- We model each document $D$ as a set of words from $w_0$ to $w_n$.
- $P(w_i|C)$ means $count(w_i, C)/count(C)$
  - $count(w_i, C)$ how many training samples of class $C$ include $w_i$ where $w_i \in D$.
  - $count(C)$ how many training samples are of class $C$

# Classify Hatter or Alice

- Word appearance per class

| Word | Mad Hatter | Alice |
|---|---|---|
| I | 9 | 14 |
| you | 16 | 5 |
| twinkle | 4 | 0 |
| clock | 3 | 1 |
| now | 1 | 3 |
| in | 3 | 3 |
| # Docs | 40 | 30 |

- $D =$ "I now clock"

| Word | Mad Hatter | Alice |
|---|---|---|
| I | 9 | 14 |
| now | 1 | 3 |
| clock | 3 | 1 |
| # Docs | 40 | 30 |

- $P(Mh|D) =$ $P(Mh)P(I|Mh)P(now|Mh)P(clock|Mh)$
- $0.00024 = \frac{40}{70}\frac{9}{40}\frac{1}{40}\frac{3}{40}$
- $P(A|D) = P(A)P(I|A)P(now|A)P(clock|A)$
- $0.000\bar{6} = \frac{30}{70}\frac{14}{30}\frac{3}{30}\frac{1}{30}$

# Implementation issues

- Floating point underflow:
  - The product of many features quickly race to zero
    * Iff $P(A|D) > P(Mh|D)$ then $log(P(A|D)) > log(P(Mh|D))$
    * $classify(D) = \arg\max P(C) \prod_i P(w_i|C)$ becomes $classify(D) = \arg\max log(P(C)) + \sum_i log(P(w_i|C))$

# Implementation issues

- Zero probability
  - Dangerous when multiplying and is negative infinity in log form.
  - Smoothing is a solution
    * Simplest smoothing is to use the equation $P(w_i|C) = \frac{count(w_i,C)+1}{count(C)+n}$ where $n$ is the number of training samples.

# Bayes Theorm

- Useful to convert 1 conditional probability into another.
- $P(A|B) = \frac{P(B|A)P(A)}{P(B)}$
- $P(B|A) = \frac{P(A|B)P(B)}{P(A)}$

# Bayes on Documents

- Given words $w_0$ to $w_n$
  - The probability of $w_i$ appearing in class $C$ is $P(w_i|C)$.
  - $*$ If there are 1000 total documents in $C$ and $w_i$ in in 100 of them then $P(w_i|C) = 0.1$

# Bayes On Documents

- Given class $C$ what's the probability of $D$?
  - $P(D|C) = \prod_i P(w_i|C)$
  - $P(D|C) = P(D \cap C)/P(C)$
  - And thus $P(C|D) = P(D \cap C)/P(D)$
- We want $P(C|D)$ but we have $P(D|C)$
  - By Bayes Theorm:
  - $P(C|D) = \frac{P(D|C)P(C)}{P(D)}$
  - $P(C|D) = \frac{P(C)}{P(D)} \prod P(w_i|C)$

# Bayes On Documents

- If we want to classify 1 document, what is constant in this equation:
- – $P(C|D) = \frac{P(C)}{P(D)} \prod P(w_i|C)$
  – If we're comparing probabilities We don't actually need $P(D)$, it's a constant.
  – $\frac{P(A|D)}{P(Mh|D)} = \frac{P(A)}{P(Mh)} \frac{\prod P(w_i|A)}{\prod P(w_i|Mh)}$

# Conclusions

- Naive Bayes algorithm can be used to classify known and unknown examples using examples that have been previously annotated by a class.
- Naive Bayes is naive because it assumes no relationship between the features of a class, thus each feature is evaluated independently per class.
- Efficient and effective, it is often used in Spam classification.

# Get Help!

- My page on Naive Bayes:

  http://softwareprocess.es/wiki/Naive_Bayes

- Wikipedia: http://en.wikipedia.org/wiki/Naive_Bayes_classifier
- Another concrete example:

  http://www.cs.rpi.edu/academics/courses/fall03/ai/misc/naive-example.pdf

- Python implementation

  http://ebiquity.umbc.edu/blogger/2010/12/07/naive-bayes-classifier-in-50-lines

- Alice in wonderland http://www.gutenberg.org/ebooks/11

# Copyright