# Computational Methods for Linguists

## Ling 471

Olga Zamaraeva (Instructor)
Yuanhe Tian (TA)

05/27/21

# Reminders
## and announcements

- Please fill out the **presentations survey**
- Please **don’t miss** presentation deadlines even if they are “ungraded”
  - be mindful of the **timezones**
- **Last Blog** due today!
- **Assignment 5** published

# Reminders
## and announcements

- Assignment 5 published
  - I ask you to figure out how to sort a dict **by value** in descending order
    - using **built-in** methods
    - what's "value" in a dict?
  - As before, you will be **adapting** code
    - That's valuable skill
      - ...**more** valuable in **practice** than writing code from scratch

# Plan for today

- Some notes on lists and dicts
  - for HW5
- Visualizing results
  - = **communicating** results!
- Activity with Matplotlib
- Bonus!
  - Entropy and info gain
  - (if there's time)

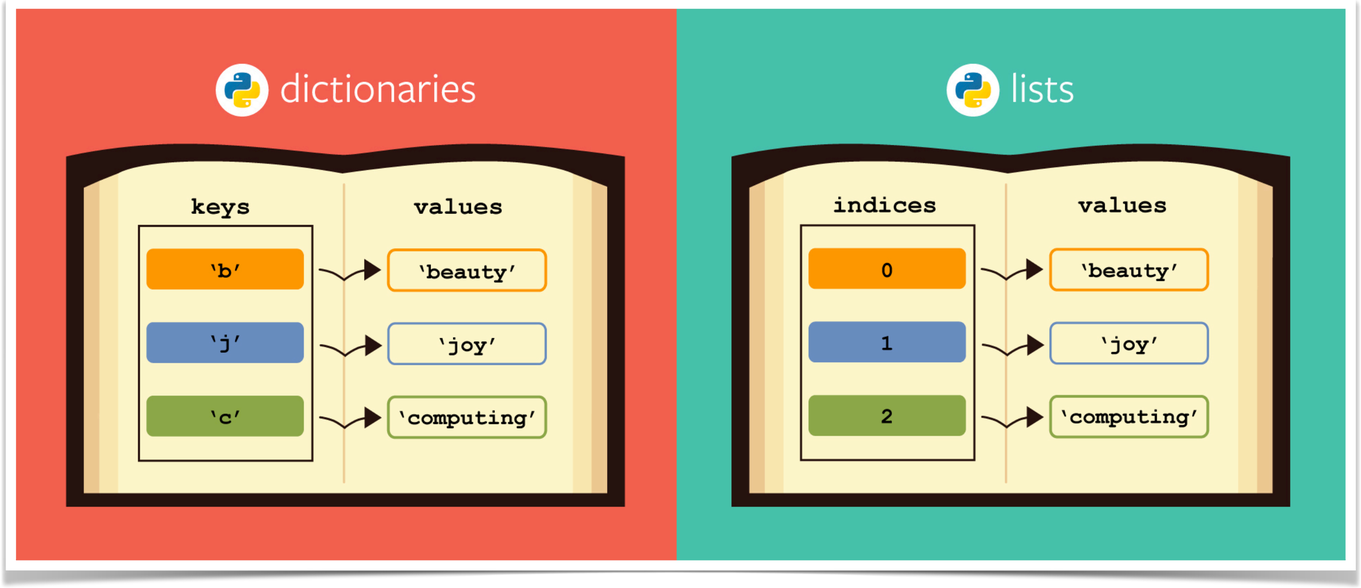
# Lists, dicts, and sorting

# Lists, dicts, and sorting
## data structures

- Lists are ordered
- Dicts are unordered
- Can a dict be sorted?
  - No, in a sense that there is no direct method and no properties associated with it
  - Yes, in a sense that in practice, a dict can contain key-value pairs in a sorted order
  - In practice, we often need that

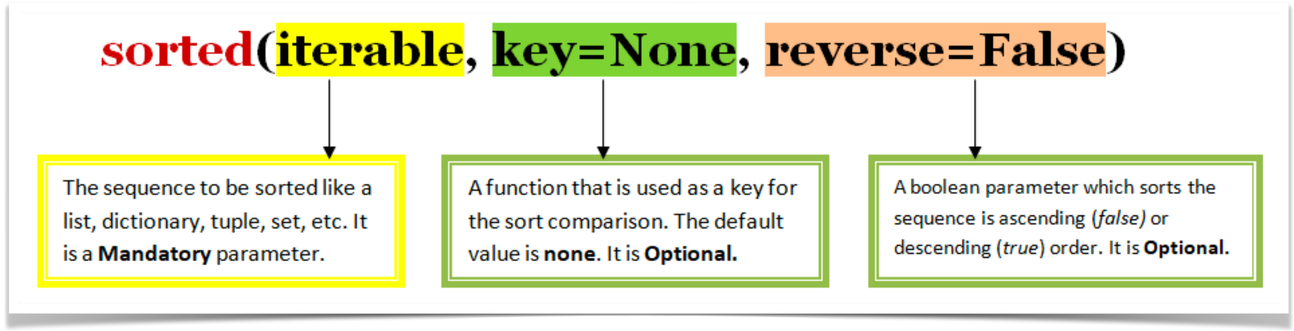
| | Mutable | Ordered | Indexing / Slicing | Duplicate Elements |
|---|---|---|---|---|
| List | ✓ | ✓ | ✓ | ✓ |
| Tuple | ✗ | ✓ | ✓ | ✓ |
| Set | ✓ | ✗ | ✗ | ✗ |

https://towardsdatascience.com/15-examples-to-master-python-lists-vs-sets-vs-tuples-d4ffb291cf07



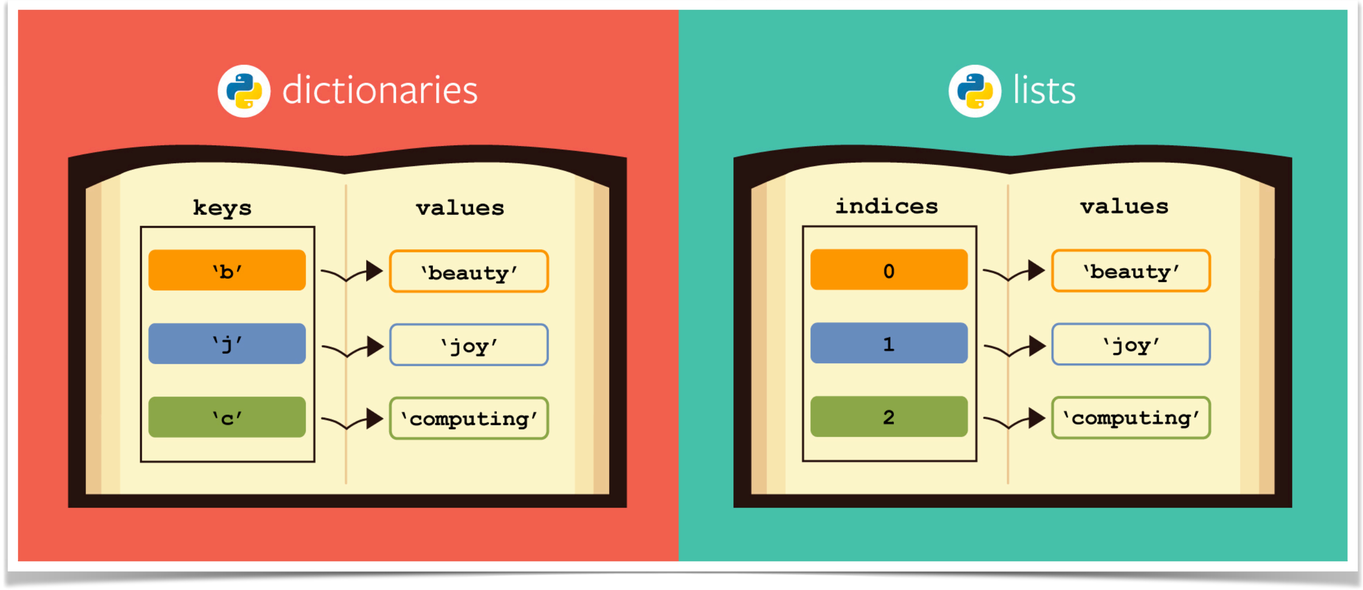

https://bjc.edc.org/bjc-r/cur/programming/old-labs/python/comparing_dicts_lists.html?topic=nyc_bjc%2FNA-python.topic

# Lists, dicts, and sorting




https://laptrinhx.com/how-to-sort-a-dictionary-by-value-in-python-2183312668/

- How to sort a dict?
- **Conceptually:**
  - treat it as a list of key-value pairs
    - items()
  - sort the list (using **sorted()** )
  - then convert back to dict
    - **if needed!**
- **Technically:**
  - Try to figure out on your own for HW5!




https://bjc.edc.org/bjc-r/cur/programming/old-labs/python/comparing_dicts_lists.html?topic=nyc_bjc%2FNA-python.topic

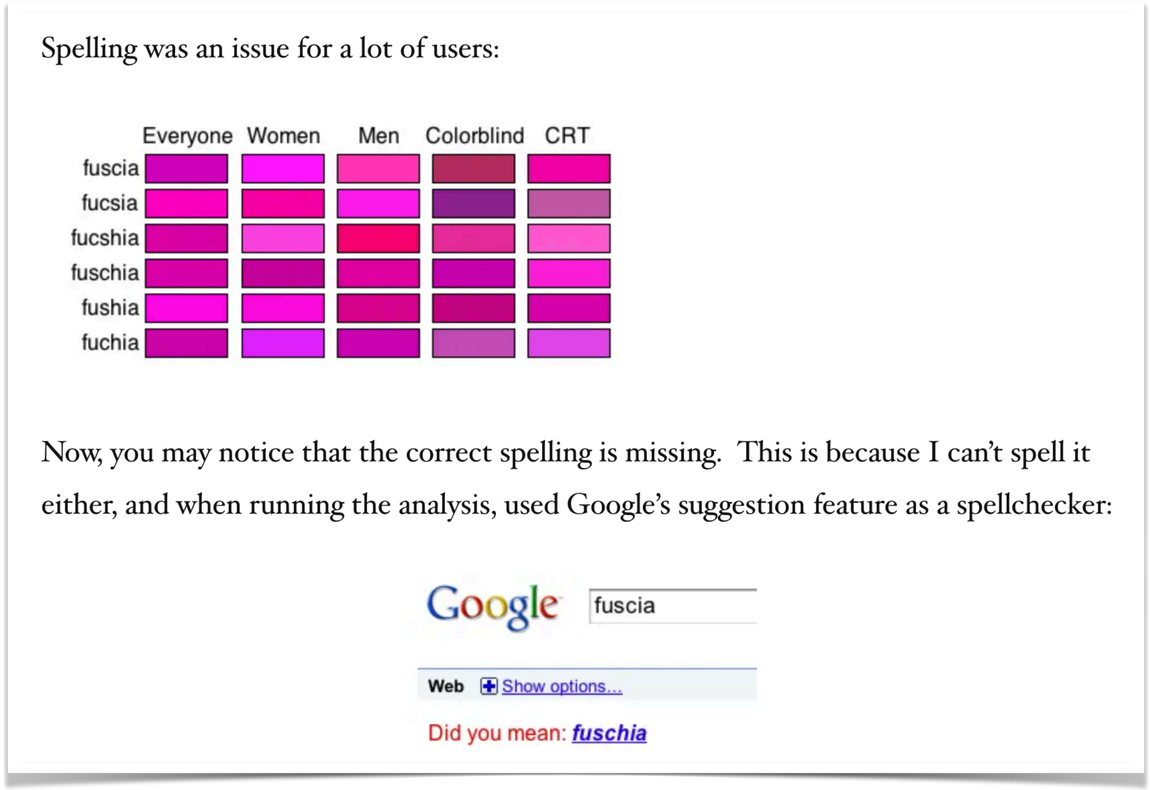
# Adapting code
## in your HW

- We had to modify the same functions multiple times wrt their “signatures”
- ...annoying
- ...but often needed in practice
- ...that being said, part of it is because assignments are new :)
  - Thanks for your patience!

# Visualizing results

# Matplotlib

- https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.html
- line **styles**:
  - https://matplotlib.org/2.0.2/api/lines_api.html
- **colors**:
  - https://matplotlib.org/stable/gallery/color/named_colors.html




https://blog.xkcd.com/2010/05/03/color-survey-results/

# Pandas
## visualization

- Uses **matplotlib** under the hood
- https://pandas.pydata.org/pandas-docs/stable/user_guide/visualization.html

Tons of resources/tutorials of all sorts!

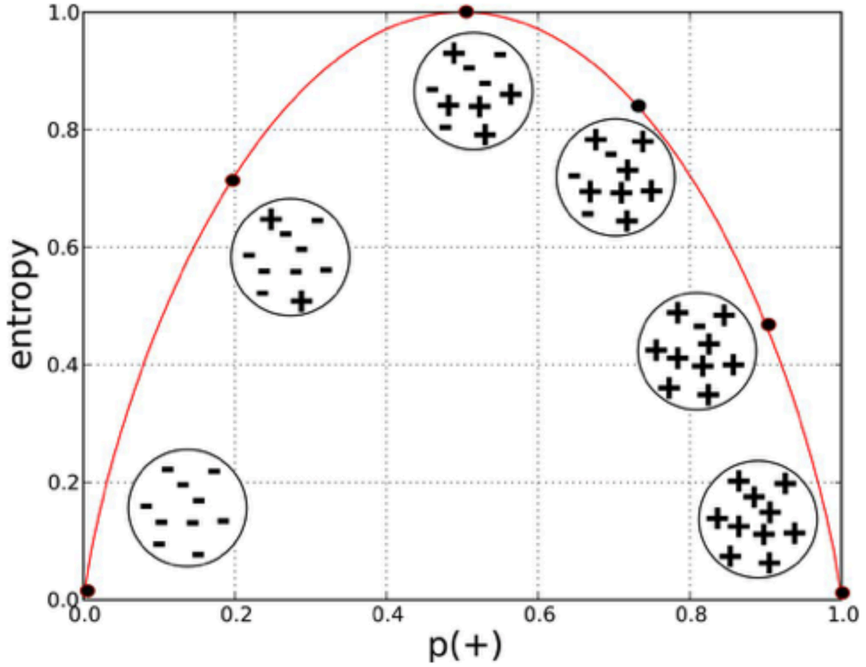
I will demo a few things!

After that, go and explore for your HW5 :)

# Bonus:
# Entropy and Information Gain
# Decision Trees

# Entropy and information gain

- Entropy:
  - measure of **unpredictability**
    - measured in "**bits**"
  - P=**1/2** (50-50) -> **most** entropy
- Information gain:
  - IF(Y,X) = E(Y) - E(Y|X)
- Some learning algorithms:
  - follow the **biggest IG**!
  - e.g. Decision Tree




https://towardsdatascience.com/entropy-how-decision-trees-make-decisions-2946b9c18c8

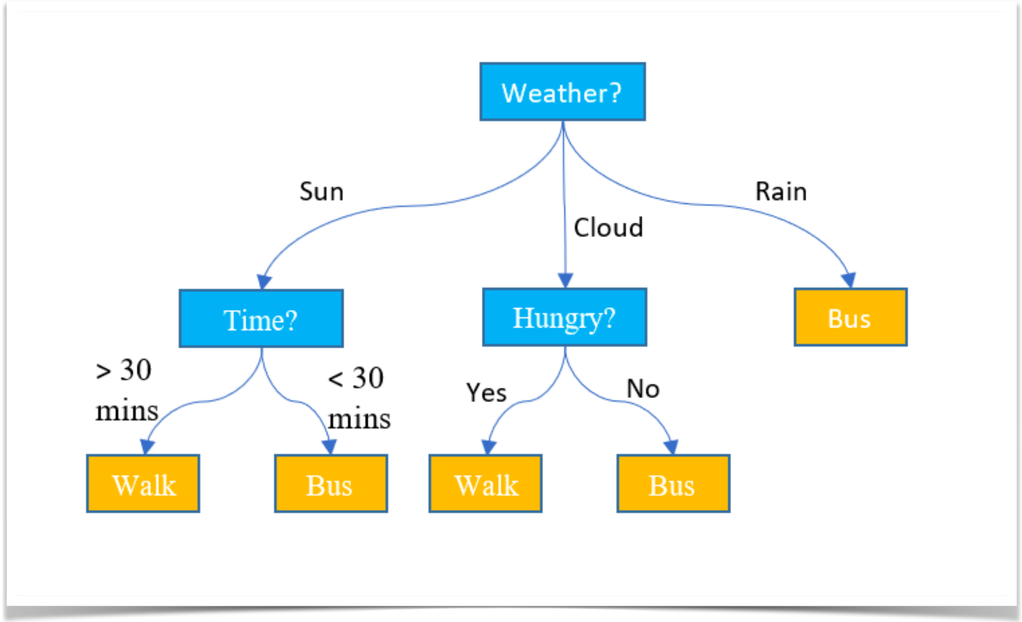
$$E(S) = \sum_{i=1}^{c} - p_i \log_2 p_i$$

Entropy. Sometimes also denoted using the letter 'H'

$$-\frac{3}{10} \times \log_2\left(\frac{3}{10}\right) - \frac{7}{10} \times \log_2\left(\frac{7}{10}\right) \approx 0.88$$

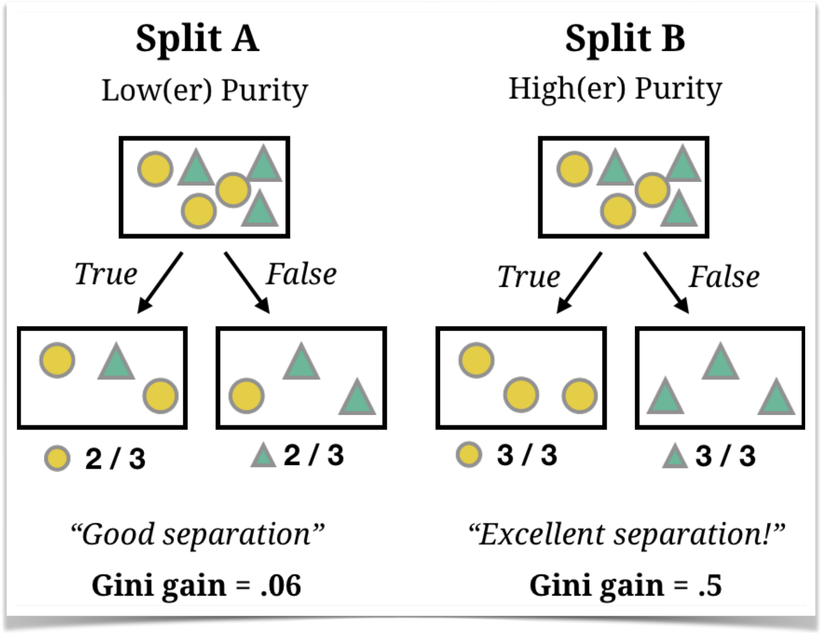
# Example
## Decision trees

- Classification algorithm
- Data is classic feature vectors
- Data organized into tree nodes
- Edges are feature values
  - “decisions”
- Picture:
  - an unnamed feature with values T/F
- Goal: Learn Tree A, not B
  - ...then simply use it deterministically to classify new data which as same feature representation!



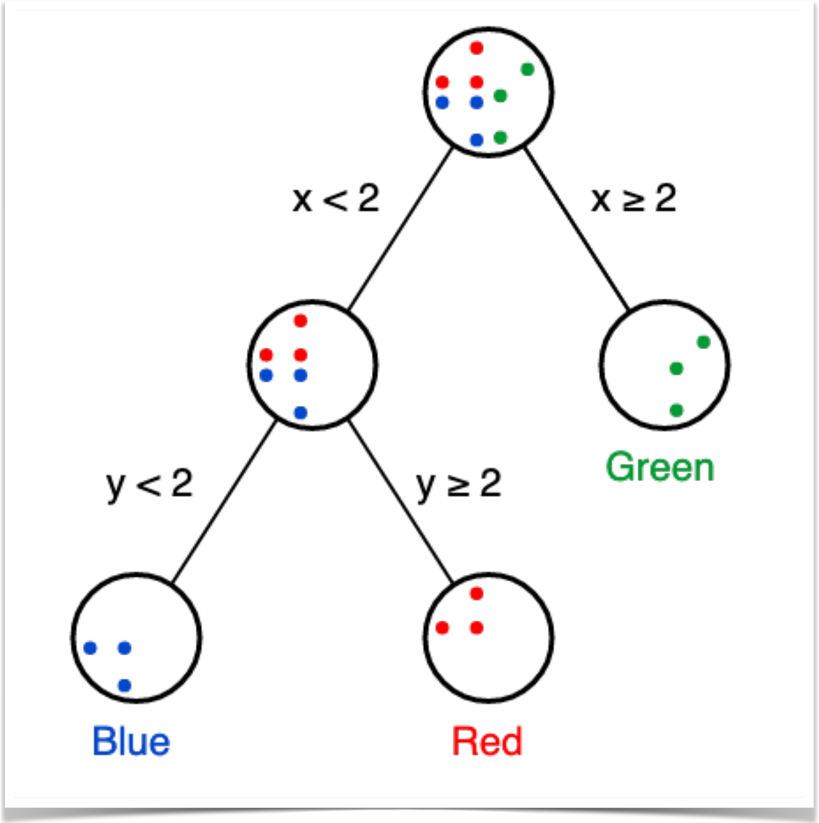

http://www.niser.ac.in/~smishra/teach/cs460/lectures/lec11/

https://therbootcamp.github.io/ML_2019May/_sessions/Prediction/Prediction.html#1



# Example
## Decision trees

- Classification algorithm
- Data is classic feature vectors
- Data organized into tree nodes
- Edges are feature values
  - “decisions”
- Picture:
  - Data has form [x,y]
  - E.g. one of blue points:
    - [x=0,y=0]
  - E.g. one of red points:
    - [x=0,y=3]



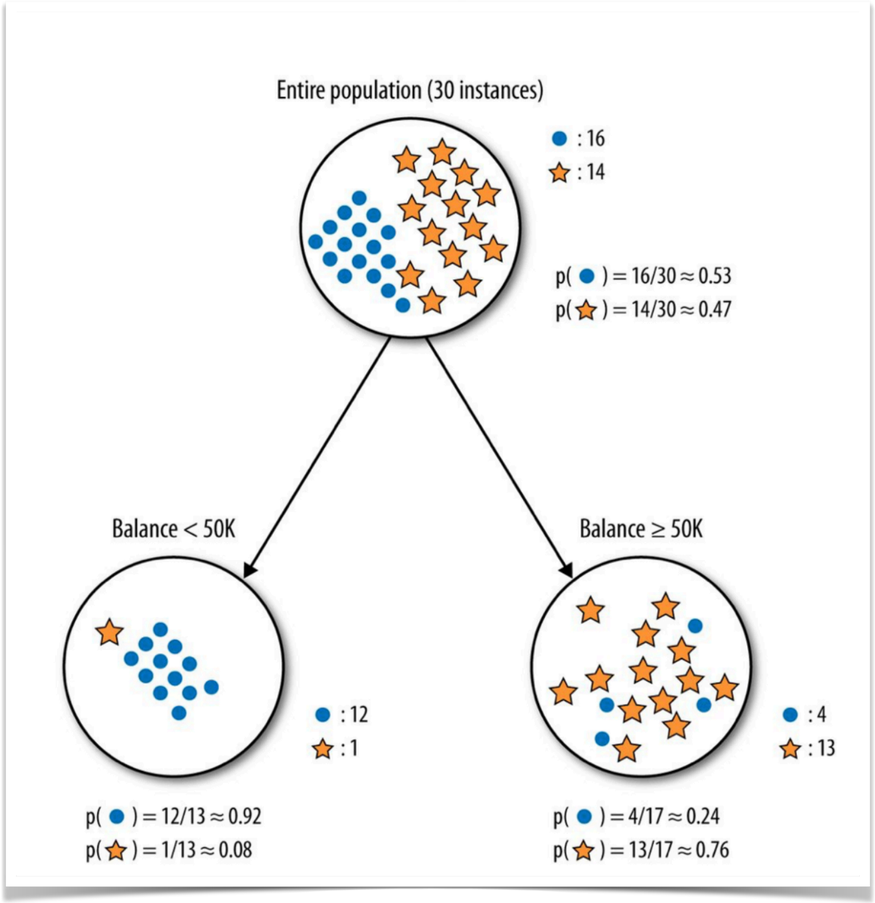

https://victorzhou.com/blog/intro-to-random-forests/

# Entropy and information gain

$$E(Parent) = -\frac{16}{30}\log_2\left(\frac{16}{30}\right) - \frac{14}{30}\log_2\left(\frac{14}{30}\right) \approx 0.99$$

$$E(Balance < 50K) = -\frac{12}{13}\log_2\left(\frac{12}{13}\right) - \frac{1}{13}\log_2\left(\frac{1}{13}\right) \approx 0.39$$

$$E(Balance > 50K) = -\frac{4}{17}\log_2\left(\frac{4}{17}\right) - \frac{13}{17}\log_2\left(\frac{13}{17}\right) \approx 0.79$$



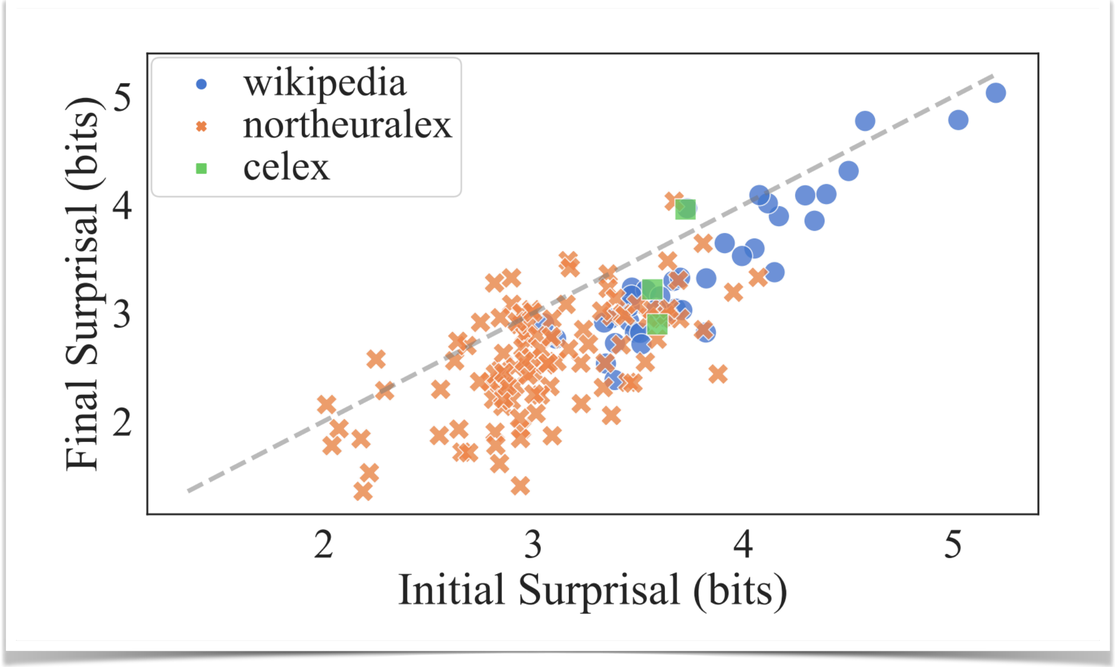

$$E(S) = \sum_{i=1}^{c} -p_i \log_2 p_i$$

Entropy. Sometimes also denoted using the letter 'H'

https://towardsdatascience.com/entropy-how-decision-trees-make-decisions-2946b9c18c8

# Case study: Disambiguatory Signals
## Pimentel et al. 2021

- Conjecture:
  - **More** information at the **beginning** of words than at the end
- **Theoretical** evidence:
  - **Information theory**
    - Info “gain”
    - Based on a **mathematical** notion of “**surprise**” (how easily **predictable**?)
    - Related concept: “**entropy**”
  - **This** paper: Probability distributions over phonological possibilities; DL networks




https://www.aclweb.org/anthology/2021.eacl-main.3.pdf

| Dataset | # Languages | Surprisal | | | | |
|---|---|---|---|---|---|---|
| | | Forward | Backward | Unigram | Position-Specific | Cloze |
| CELEX | 3 | 3 \| 0 | 0 \| 3 | 2 \| 0 | 2 \| 1 | 2 \| 1 |
| NorthEuraLex | 107 | 106 \| 0 | 11 \| 31 | 71 \| 1 | 24 \| 4 | 45 \| 1 |
| Wikipedia | 41 | 41 \| 0 | 0 \| 39 | 39 \| 1 | 31 \| 1 | 35 \| 2 |

Table 1: Number of languages in the analysed datasets with significantly larger surprisals in initial | final positions.

# Lecture survey in the chat!