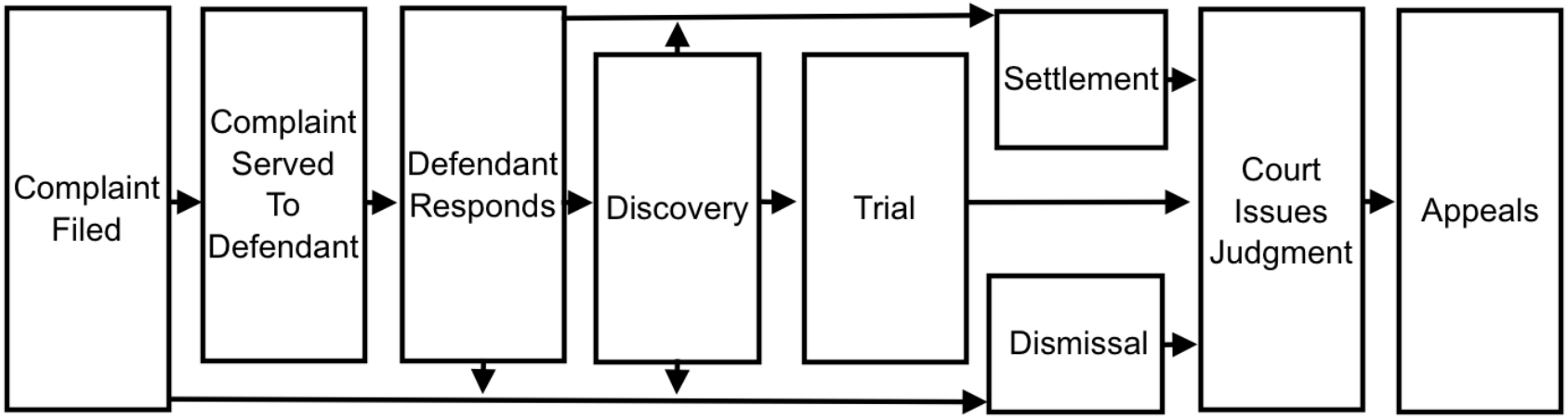
PLSC 476: Empirical Legal Studies

Christopher Zorn

April 20, 2021

The Civil Litigation Process



Civil Litigation: Empirical Moments

- Document Review (pre-litigation; for contracts, etc.)
- Legal Research (finding relevant laws, precedents, etc.)
- Predictive Tools (litigation costs, likely trial outcomes, settlement probabilities, etc.)
- Settlement Calculators
- Electronic Discovery
- Many others...

Relevant Federal Law

- The *Stored Communication Act* ("SCA") (1986)
  - Foundational law governing access to electronically stored information held by third parties (especially ISPs)
  - Both protects users from unlawful access to their data *and* defines the terms under which compelled disclosure of that data may take place
  - Also governs data preservation
- The *CLOUD* Act ("Clarifying Lawful Overseas Use of Data") (2018)
  - Extends the SCA to data held outside the U.S.
- The *Federal Rules of Civil Procedure* ("FRCP")
  - The "First Rule" of civil procedure: The FRCP are "construed, administered, and employed by the court and the parties to secure the just, speedy, and inexpensive determination of every action and proceeding."
  - Amended 2006, 2009 & 2015 to cope with electronically stored information ("ESI") and e-discovery

How Discovery Works: A Brief History

???-1960s: Paper-Based Discovery

1. Attorneys request (paper) copies of relevant documents
2. Attorneys review documents manually for relevance

1970s-2010s: eDiscovery 1.0

1. Attorneys request electronic (image or OCR-ready) copies of relevant documents
2. Attorneys review documents for relevance
    a. 1970s-1990s: Manually
    b. 1990s-2010s: Digitally (via search tools)

2010s-Present: eDiscovery 2.0 ("Technology-Assisted Review")

1. Attorneys request machine-readable electronic documents
2. Legal services companies (LSCs) use machine learning / text analysis tools to review documents
3. Attorneys review LSC findings

Text As Data: Concepts

Machine Learning ("ML")

- Teaching computers to "think" and learn like humans
- Includes a wide range of classification and prediction models (including simple multivariate ones like we used last week)
- Uses: Search + recommender engines; image/voice recognition; forecasting; others

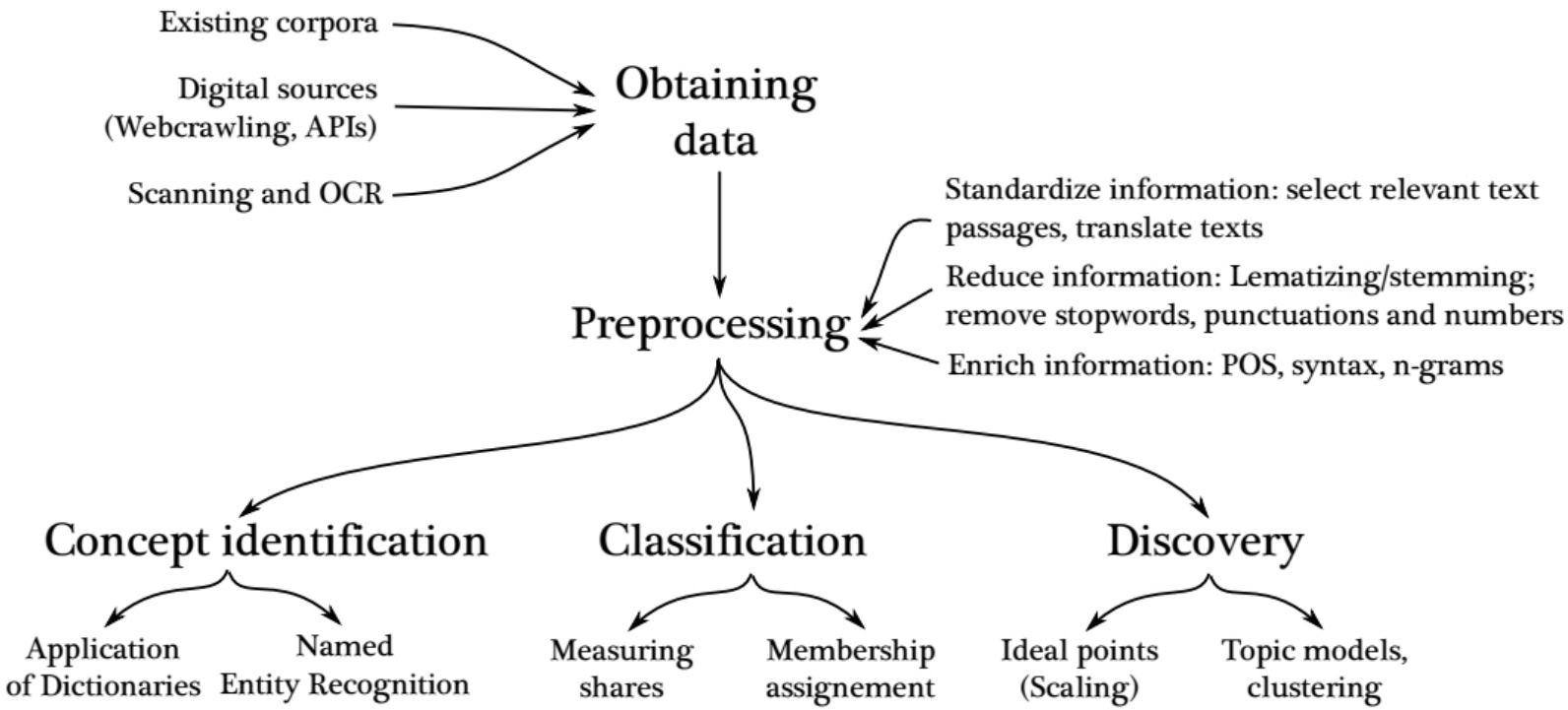
Natural Language Processing ("NLP")

- How computers "understand" human language
- Generally requires text to be represented numerically
- Uses: Predictive text (spell checking, mail filters, etc.); machine translation; attribution/plagiarism detection; others

Flavors:

- Supervised:
  - Train a computer to recognize / predict patterns in data **A**
  - $\rightarrow$ use that training to identify patterns in data **B**
- Unsupervised: Use patterns in the data to group/classify text/documents

Text As Data: Processes



Source

Stupid Text Tricks: Semantic Analysis

Variations ("S" = supervised, "U" = unsupervised):

- Part-Of-Speech Tagging (S + U)
- Word Sense Disambiguation (U)
- Named Entity Recognition (S + U)
- Sentiment Analysis (S + U)
- Topic Modeling (S + U)
- Terminology Extraction (U)

How Do We Analyze Text?

General Idea:

- Represent text $\mathfrak{D}$ as a numerical array $\mathbf{C}$
- Analyze $\mathbf{C} \rightarrow$ generate predictions / classifications $\hat{\mathbf{P}}$
- Map $\hat{\mathbf{P}}$ back to $\mathfrak{D}$ or $\mathfrak{D}'$

Key Steps / Concepts:

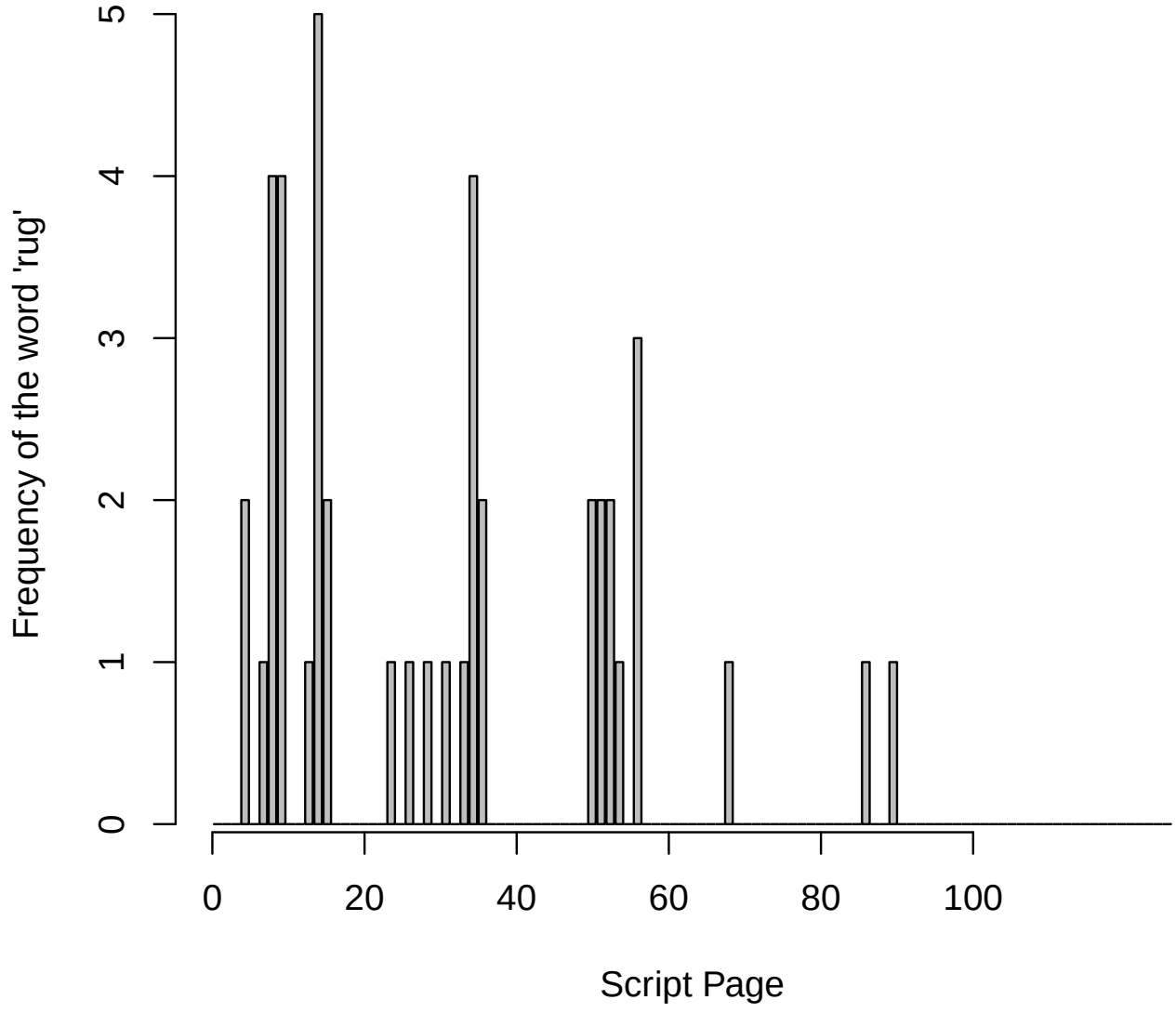
1. Text *Preprocessing*:
    - Removing capitalization, punctuation, "stop words," etc.
    - Stemming / lemmatization (e.g., `traveler`, `traveling` $\rightarrow$ `travel*`)
2. *Document-Term Matrix* ("DTM")
    - Rows = *documents* (sentences, speeches, tweets, etc.)
    - Columns = *terms* (words / phrases)
    - Cells = counts (or weighted counts)

An Example...

Creating a DTM

```
> TBL<-pdf_text("https://www.raindance.org/scripts/The%20Big%20Lebowski%20script.pdf")

> # Turn that into a "corpus":

> TBL.corp<-SimpleCorpus(VectorSource(TBL))

> TBL.dtm <- DocumentTermMatrix(TBL.corp,
+                 control=list(removePunctuation=TRUE,
+                              stopwords=TRUE,
+                              tolower=TRUE,
+                              stemming=TRUE,
+                              removeNumbers=FALSE,
+                              weight=weightTfIdf))

> dim(TBL.dtm)
[1]  105 2863

> inspect(TBL.dtm)

<<DocumentTermMatrix (documents: 105, terms: 2863)>>
Non-/sparse entries: 8556/292059
Sparsity           : 97%
Maximal term length: 23
Weighting          : term frequency (tf)
Sample             :
    Terms
Docs car donni dude fuck know lebowski look man maud walter
  1    0     0    4    0    0        3    1   2    0      0
  2    1     0    9    0    0        0    0   6    0      0
  29   0     0    9    0    0        0    3   2    0      0
  30   0     0    6    0    1        0    2   1    0      0
  43   2     0    9    0    0        3    1   0    1      0
  54   0     0    9    0    0        2    3   6    0      0
  64   6     0   10    1    0        0    2   2    0      0
  72   0     0    1    1    1        4    0   5    0      0
  76   4     0    7    0    0        1    2   0    1      0
  87   1     0    3    4    1        1    1   2    0      3
```

Frequency of the Word “rug”



Terms Correlated with the Word "rug"