Text Data in Business and Economics

Basel University - Autumn 2023

1. Overview

Welcome

- This course focuses on applications of **natural language processing** in **applied economics**.

Welcome

- This course focuses on applications of **natural language processing** in **applied economics**.
- Methods:
  - Develop skills in applied natural language processing
  - Convert natural language texts – e.g. legal and political documents – to data.

Welcome

- This course focuses on applications of **natural language processing** in **applied economics**.
- Methods:
  - Develop skills in applied natural language processing
  - Convert natural language texts – e.g. legal and political documents – to data.
- Economics:
  - Relate text data to metadata to understand economic/political/social forces.
  - e.g., analyze the motivations and decisions of public officials through their writings and speeches.
  - Assess the real-world impacts of language on government and the economy.

Logistics

Learning Materials

Course Content Overview

Schedule

- See syllabus for the schedule.
- 10 lectures (5 meetings, with 2 lectures each):
  - up to 40 minutes of lecturing
  - ~20 minutes for student presentations/discussions of papers
  - remaining time: going over the example notebooks

Course Learning Materials

- Course Syllabus:
  - https://docs.google.com/document/d/1EO8q8iBK0LvDBp-LTLqnyXS2P2OCoRwnt1B7-hUpTPs/edit
- Course Repo:
  - https://github.com/BenjaminArold/Course-Text-Data-23

Course Communication

- Course announcements will be done via email (if you have not been getting emails from me already, let me know).

Logistics

Learning Materials

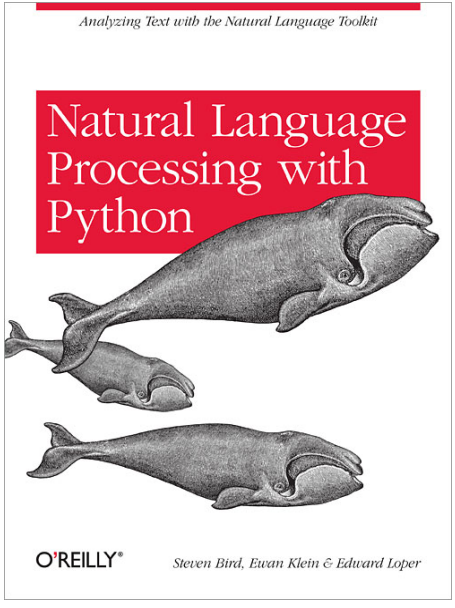
Course Content Overview

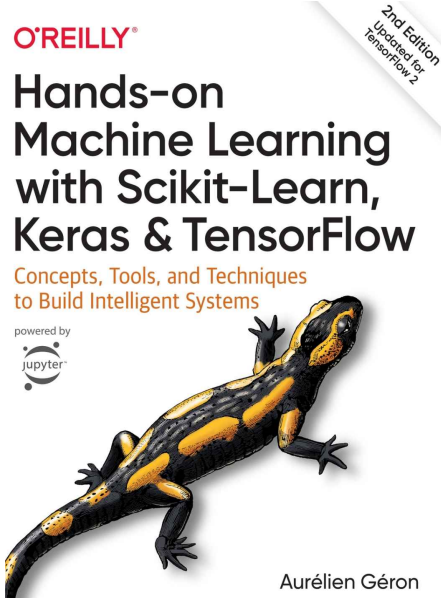
Course Bibliographies

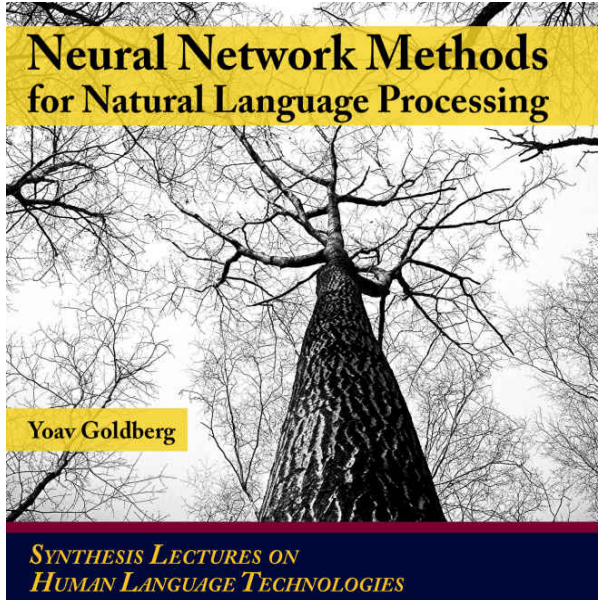
- Bibliography of references:
  - reference readings on tools/methods
  - not required, but useful to complement the slides

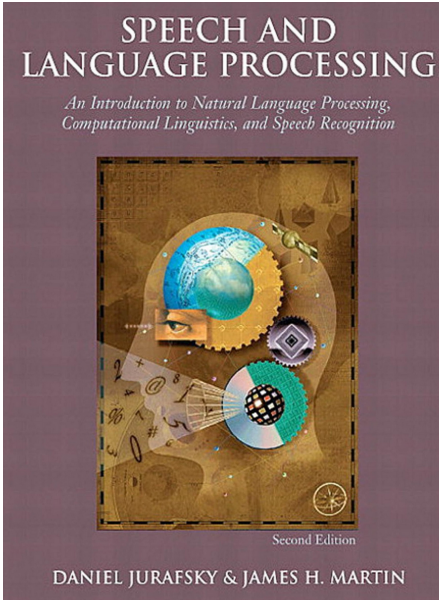
Course Bibliographies

- Bibliography of references:
  - reference readings on tools/methods
  - not required, but useful to complement the slides
- Bibliography of applications:
  - economics application papers, for class presentations.


Analyzing Text with the Natural Language Toolkit
Natural Language Processing with Python
O'REILLY®
Steven Bird, Ewan Klein & Edward Loper


O'REILLY
2nd Edition
Updated for TensorFlow 2
Hands-on Machine Learning with Scikit-Learn, Keras & TensorFlow
Concepts, Tools, and Techniques to Build Intelligent Systems
powered by
jupyter
Aurélien Géron


Neural Network Methods
for Natural Language Processing
Yoav Goldberg
Synthesis Lectures on
Human Language Technologies


SPEECH AND
LANGUAGE PROCESSING
An Introduction to Natural Language Processing,
Computational Linguistics, and Speech Recognition
Second Edition
DANIEL JURAFSKY & JAMES H. MARTIN

Main Python packages for NLP

- Python 3 is ideal for text data and natural language processing.
  - Can use Anaconda or download the packages we need to a pip environment.
    - nltk – broad collection of pre-neural-nets NLP tools
    - scikit-learn – ML package with nice text vectorizers, clustering, and supervised learning
    - xgboost – gradient-boosted machines for supervised learning
    - gensim – topic models and embeddings
    - spaCy – tokenization, NER, parsing, pre-trained vectors
    - huggingface – source for pre-trained transformer models.

Coding Practice and Assignments

Main Coding Examples on GitHub (discussed in class):`https://github.com/BenjaminArold/Course-Text-Data-23/tree/main/notebooks`

Additional Assignments on GitHub (not discussed in class): `https://github.com/BenjaminArold/Course-Text-Data-23/tree/main/assignments`

Discussant Presentations

- At the end of most lectures, we will have one discussant presentations on one of the economics articles listed in the syllabus.
- Please sign up here:

  https://docs.google.com/spreadsheets/d/1ASbOxEPEwhZeDo6JefnZZGUxBGdYbMjRwdnESTCZUvM/edit#gid=0
- Critical presentations are up to 10 minutes, should present and critique:
  - research question
  - text-analysis methods
  - empirical methods
  - results
  - contribution

Logistics

Learning Materials

Course Content Overview

Text is high-dimensional

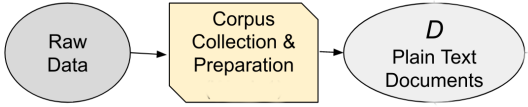
- sample of documents, each n_L words long, drawn from vocabulary of n_V words.
- The unique representation of each document has dimension $n_V^{n_L}$.
  - e.g., a sample of 30-word Twitter messages using only the one thousand most common words in the English language
    - $\rightarrow$ dimensionality $= 1000^{30} = 10^{32}$

"Text as Data", GKT 2017

Summarize analysis in three steps:

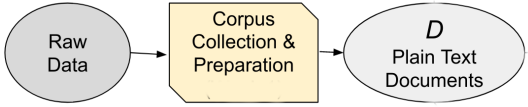
- convert raw text D to numerical array $\boldsymbol{C}$
  - The elements of $\boldsymbol{C}$ are counts over tokens (words or phrases)
- map $\boldsymbol{C}$ to predicted values $\hat{\boldsymbol{V}}$ of unknown outcomes $\boldsymbol{V}$
  - Learn $\hat{\boldsymbol{V}}(\boldsymbol{C})$ using machine learning
  - e.g. supervised learning for some labeled $\boldsymbol{C}_i$ and V_i
  - or unsupervised learning of topics/dimensions just from $\boldsymbol{C}$
- use $\hat{\boldsymbol{V}}$ for subsequent descriptive or causal analysis

Corpora



- Text data is a sequence of characters called documents.
- The set of documents is the corpus, which we will call D.

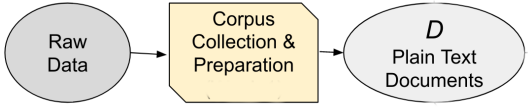
Corpora



- Text data is a sequence of characters called documents.
- The set of documents is the corpus, which we will call D.

- Text data is **unstructured**:
  - the information we want is mixed together with (lots of) information we don't.

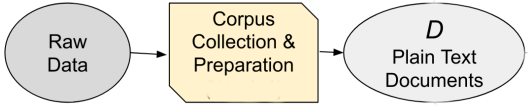
Corpora



- Text data is a sequence of characters called documents.
- The set of documents is the corpus, which we will call D.

- Text data is **unstructured**:
  - the information we want is mixed together with (lots of) information we don't.
- All text data approaches will throw away some information:
  - The trick is figuring out how to retain valuable information.

Corpora



- Text data is a sequence of characters called documents.
- The set of documents is the corpus, which we will call D.

- Text data is **unstructured**:
  - the information we want is mixed together with (lots of) information we don't.
- All text data approaches will throw away some information:
  - The trick is figuring out how to retain valuable information.
- The tools from Lectures 3 (Tokenization) and 5 (Dimension Reduction) are focused on this step:
  - transforming an unstructured corpus D to a usable matrix X.

This course is about relating documents to metadata

- This course is on NLP for **economics**:
  - the documents are not that meaningful by themselves.
  - we want to relate **text** data to **metadata**.

This course is about relating documents to metadata

- This course is on NLP for **economics**:
  - the documents are not that meaningful by themselves.
  - we want to relate **text** data to **metadata**.
- e.g., measuring positive-negative sentiment Y in political speeches.
  - not that meaningful by itself.

This course is about relating documents to metadata

- This course is on NLP for **economics**:
  - the documents are not that meaningful by themselves.
  - we want to relate **text** data to **metadata**.
- e.g., measuring positive-negative sentiment Y in political speeches.
  - not that meaningful by itself.
- but how about sentiment Y_{ijkt} in speech i by politician j on topic k at time t:
  - how does sentiment vary over time t?
  - does politician from party p_j express more negative sentiment toward topic k?

What counts as a document?

The unit of analysis (the "document") will vary depending on your question.

- needs to be fine enough to fit the relevant metadata variation
- should not be finer – would make dataset more high-dimensional without relevant empirical variation.

What counts as a document?

The unit of analysis (the "document") will vary depending on your question.

- needs to be fine enough to fit the relevant metadata variation
- should not be finer – would make dataset more high-dimensional without relevant empirical variation.

E.g., what should we use as the document in these contexts?

1. predicting whether a judge is right-wing or left-wing in partisan ideology, from their written opinions.
2. predicting whether parliamentary speeches become more emotive in the run-up to an election

Handling Corpora

- There is already a vast amount of data out there that has already been compiled (e.g. ParlSpeech, CourtListener, Twitter, New York Times, Google Trends, Wikipedia).

Handling Corpora

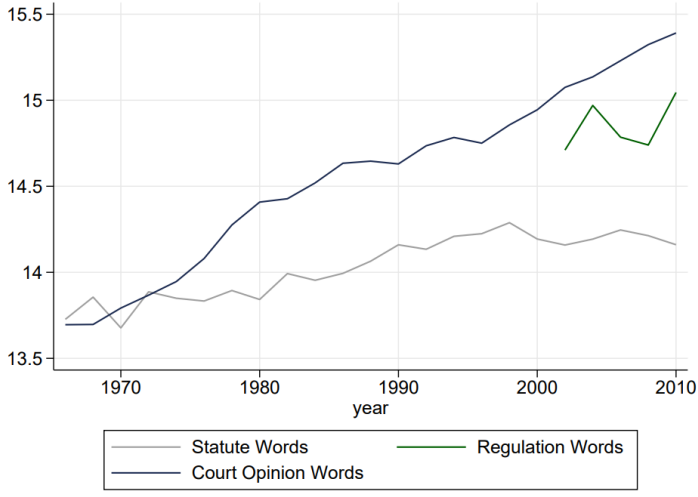
- There is already a vast amount of data out there that has already been compiled (e.g. ParlSpeech, CourtListener, Twitter, New York Times, Google Trends, Wikipedia).
- Everyone in this class should learn how to:
  1. query REST API's
  2. run a web scraper in selenium
  3. do pre-processing on corpora, e.g. to remove HTML markup, fix errors associated with OCR.

Handling Corpora

- There is already a vast amount of data out there that has already been compiled (e.g. ParlSpeech, CourtListener, Twitter, New York Times, Google Trends, Wikipedia).
- Everyone in this class should learn how to:
  1. query REST API's
  2. run a web scraper in selenium
  3. do pre-processing on corpora, e.g. to remove HTML markup, fix errors associated with OCR.
- All of the tools that we discuss in this class are available in many languages, and machine translation is now quite good and automatable (e.g. `huggingface.co/docs/transformers/master/en/model_doc/marian`).

Quantity of Text as Data

e.g., Ash, Morelli, and Vannoni (2022), "More laws, more growth?"



note log scale – per year we see:

- ~1.3M words in statutes
- ~3.3M words in regulations
- ~4.8M words in state court opinions

Dictionary Methods

e.g., Baker, Bloom, and Davis (QJE 2016), "Measuring Policy Uncertainty"

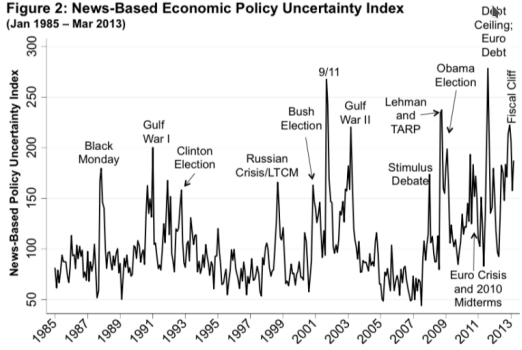
Dictionary Methods

e.g., Baker, Bloom, and Davis (QJE 2016), "Measuring Policy Uncertainty"

For each newspaper on each day since 1985, tag each article mentioning

1. uncertainty word
2. economy word
3. policy word (eg "legislation", "regulation")

Normalize resulting article counts by total newspaper articles that month.



Tokenization

- Input:
  - A set of documents (e.g. text files), D.

Tokenization

- Input:
  - A set of documents (e.g. text files), D.
- Pre-processing:
  - removing page numbers, capitalization, punctuation, etc.

Tokenization

- Input:
  - A set of documents (e.g. text files), D.
- Pre-processing:
  - removing page numbers, capitalization, punctuation, etc.
- Output:
  - Tokens: A sequence, $\boldsymbol{w}$, containing a list of tokens (words) in document i, for use in natural language processing
  - Counts/Frequencies: A **document-term matrix**, X, containing statistics about word/phrase frequencies in each document.

Relating Token Counts to Metadata

e.g., Loan Application Words Predicting Default (Netzer, Lemaire, and Herzenstein 2019)

Relating Token Counts to Metadata

e.g., Loan Application Words Predicting Default (Netzer, Lemaire, and Herzenstein 2019)

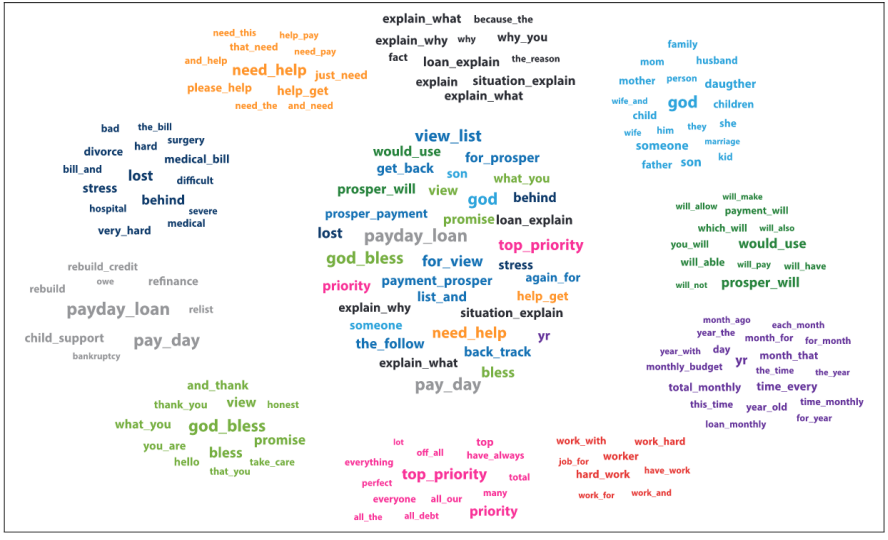


Figure 3. Words indicative of loan default.

Notes: The most common words appear in the middle cloud (cutoff = 1:1.5) and are then organized by themes. Starting on the top and moving clockwise: words related to explanations, external influence words and others, future-tense words, time-related words, work-related words, extremity words, words appealing to lenders, words relating to financial hardship, words relating to general hardship, and desperation/plea words.

Document Distance – e.g. Burgess et al, "Legislative Influence Detectors"

- Compare bill texts across states in two-step process:
  (1) find statutes with a high cosine similarity between the associated n-gram frequency vectors

$$\cos\_\text{sim}(x_1, x_2) = \frac{x_1 \cdot x_2}{||x_1|| ||x_2||}$$

  (2) compare candidates using text reuse score (plagiarism detection algorithm).

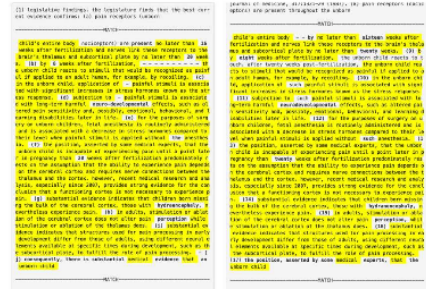


Figure 10: Match between Scott Walker's bill and a highly similar bill from Louisiana. For a detailed view, please visit http://dssg.uchicago.edu/lid/.

Document Distance – e.g. Burgess et al, "Legislative Influence Detectors"

- Compare bill texts across states in two-step process:
  (1) find statutes with a high cosine similarity between the associated n-gram frequency vectors

$$\cos\_\text{sim}(x_1, x_2) = \frac{x_1 \cdot x_2}{||x_1|| ||x_2||}$$

  (2) compare candidates using text reuse score (plagiarism detection algorithm).

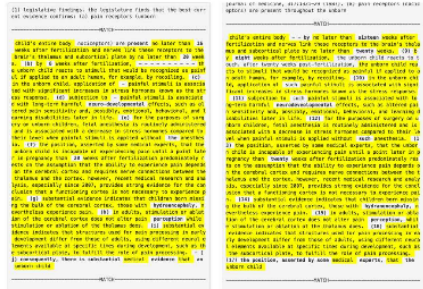


Figure 10: Match between Scott Walker's bill and a highly similar bill from Louisiana. For a detailed view, please visit http://dssg.uchicago.edu/lid/.

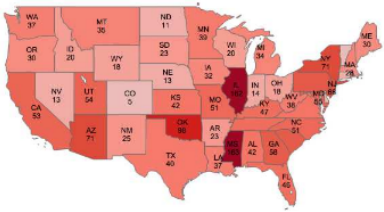


Figure 7: Introduced bills by state from ALEC model legislation

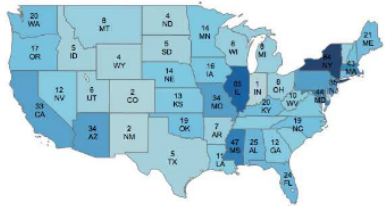


Figure 8: Introduced bills by state from ALICE model legislation

Dimensionality Reduction

- **Dimensonality reduction**
  - makes data more interpretable – for example by projecting down to two dimensions for visualization.
  - improves computational tractability.
  - can improve model performance.

Dimensionality Reduction

- **Dimensonality reduction**
  - makes data more interpretable – for example by projecting down to two dimensions for visualization.
  - improves computational tractability.
  - can improve model performance.

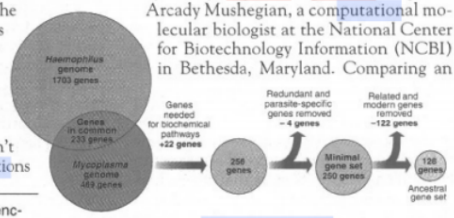
- Much of what we do in this class is some form of informative dimensionality reduction.
- More explicitly:
  - run PCA on the document-term matrix $\boldsymbol{X}$
  - use clustering to identify groups of similar documents
  - use topic models to identify relevant topics across documents.

LDA (Latent Dirichelt Allocation): A statistical highlighter

Seeking Life's Bare (Genetic) Necessities

COLD SPRING HARBOR, NEW YORK— How many genes does an organism need to survive? Last week at the genome meeting here,* two genome researchers with radically different approaches presented complementary views of the basic genes needed for life. One research team, using computer analyses to compare known genomes, concluded that today's organisms can be sustained with just 250 genes, and that the earliest life forms required a mere 128 genes. The other researcher mapped genes in a simple parasite and estimated that for this organism, 800 genes are plenty to do the job—but that anything short of 100 wouldn't be enough.

Although the numbers don't match precisely, those predictions "are not all that far apart," especially in comparison to the 75,000 genes in the human genome, notes Siv Andersson of Uppsala University in Sweden, who arrived at the 800 number. But coming up with a consensus answer may be more than just a genetic numbers game, particularly as more and more genomes are completely mapped and sequenced. "It may be a way of organizing any newly sequenced genome," explains Arcady Mushegian, a computational molecular biologist at the National Center for Biotechnology Information (NCBI) in Bethesda, Maryland. Comparing an



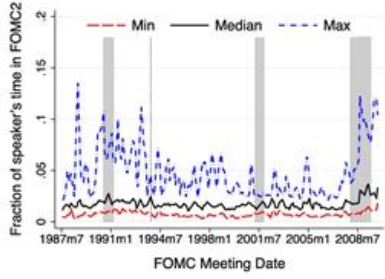
Stripping down. Computer analysis yields an estimate of the minimum modern and ancient genomes.

* Genome Mapping and Sequencing, Cold Spring Harbor, New York, May 8 to 12.

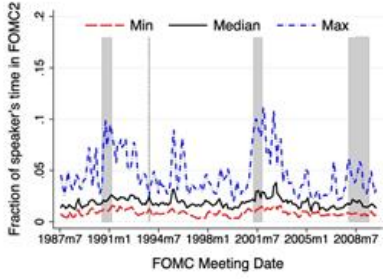
Image from Hanna Wallach

Counter-Cyclical Topics in Central Bank Discussions

Hansen, McMahon, and Prat (QJE 2017)



(A) Topic 38 'Financial sector'



(B) Topic 39 'Economic weakness'

Machine Learning with Text Data

Machine Learning with Text Data

- Say each document i has an associated outcome or label $\boldsymbol{y}_i$ with dimensions $n_y \geq 1$
- Some documents are labeled and some are unlabeled $\rightarrow$
  - we would like to learn a function $\hat{\boldsymbol{y}}(x_i)$ based on the labeled data ...
  - ... to machine-classify the unlabeled data.

e.g., OLS is Machine Learning

- Ordinary Least Squares Regression (OLS) assumes the functional form $\mathbb{E}(y|x) = f(x;\theta) = x_i'\theta$ and minimizes the mean squared error (MSE)

$$\min_{\hat{\theta}} \frac{1}{n_D} \sum_{i=1}^{n_D} (x_i'\hat{\theta} - y_i)^2$$

e.g., OLS is Machine Learning

- Ordinary Least Squares Regression (OLS) assumes the functional form $\mathbb{E}(y|x) = f(x;\theta) = x_i'\theta$ and minimizes the mean squared error (MSE)

$$\min_{\hat{\theta}} \frac{1}{n_D} \sum_{i=1}^{n_D} (x_i'\hat{\theta} - y_i)^2$$

- This minimand has a closed form solution

$$\hat{\theta} = (\boldsymbol{x}'\boldsymbol{x})^{-1}\boldsymbol{x}'\boldsymbol{y}$$

  - most machine learning models (more general $f(x;\theta)$) do **not** have a closed form solution $\rightarrow$ use numerical optimization instead (gradient descent).

e.g., OLS is Machine Learning

- Ordinary Least Squares Regression (OLS) assumes the functional form $\mathbb{E}(y|x) = f(x;\theta) = x_i'\theta$ and minimizes the mean squared error (MSE)

$$\min_{\hat{\theta}} \frac{1}{n_D} \sum_{i=1}^{n_D} (x_i'\hat{\theta} - y_i)^2$$

- This minimand has a closed form solution

$$\hat{\theta} = (\boldsymbol{x}'\boldsymbol{x})^{-1}\boldsymbol{x}'\boldsymbol{y}$$

  - most machine learning models (more general $f(x;\theta)$) do **not** have a closed form solution $\rightarrow$ use numerical optimization instead (gradient descent).
- OLS is not good for ML because it tends to overfit the training data. ML models (e.g. Lasso) add regularization to improve extrapolation to held-out data.

XGBoost ("Extreme Gradient Boosting")

- A good starting point for any machine learning task.

- easy to use
- actively developed
- available in Python and R
- efficient / parallelizable
- provides model explanations
- takes sparse matrices as input

```
from xgboost import XGBClassifier
model = XGBClassifier()

model.fit(X_train, y_train,
          early_stopping_rounds=10,
          eval_metric="logloss",
          eval_set=[(X_eval, y_eval)]
          )

y_pred = model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
```

Word Embeddings

e.g. Pennington et al (2014) (GloVe = Global Vectors):

- Input: C_{ij} = local co-occurrence counts between words $i, j \in \{1, ..., n_w\}$.

Word Embeddings

e.g. Pennington et al (2014) (GloVe = Global Vectors):

- Input: C_{ij} = local co-occurrence counts between words $i, j \in \{1, ..., n_w\}$.

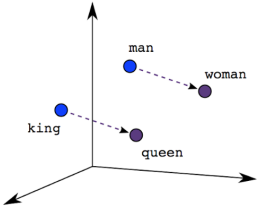
Learn word vectors $\boldsymbol{w} = (w_1, ..., w_i, ..., w_{n_w})$, where $w_i \in (-1, 1)^{100}$, to solve

$$\min_{\boldsymbol{w}} \sum_{i,j} \left(w_i^T w_j - \log(C_{ij}) \right)^2$$

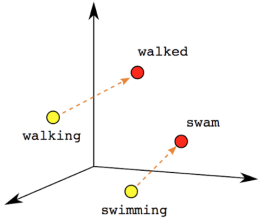
- Minimizes **squared difference** between:
  - **dot product of word vectors,** $w_i^T w_j$
  - **empirical co-occurrence,** $\log(C_{ij})$
- Intuitively: words that co-occur should have high correlation (dot product)

Vector Directions ↔ Meaning

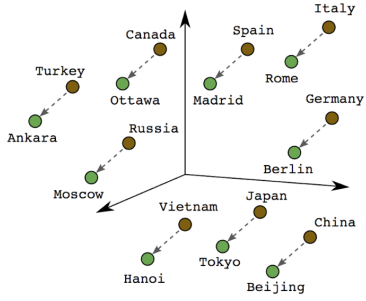
- Intriguingly, word-embedding algebra can depict conceptual, analogical relationships between words:



Male-Female

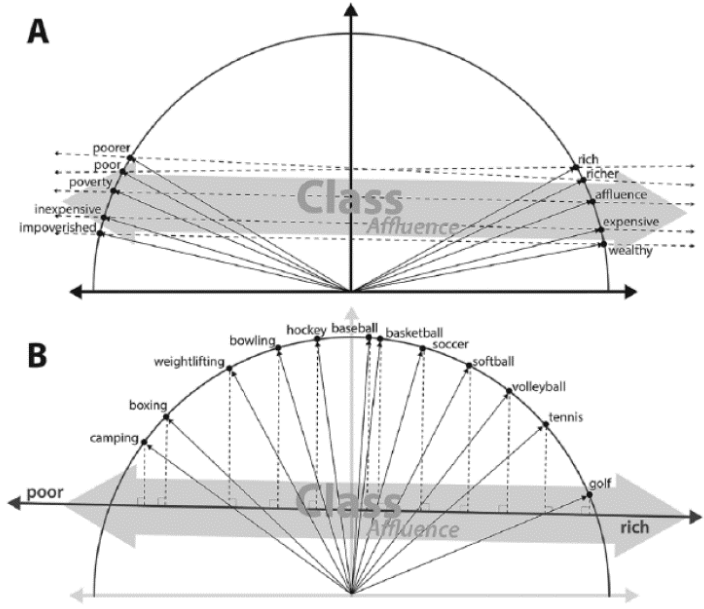


Verb Tense



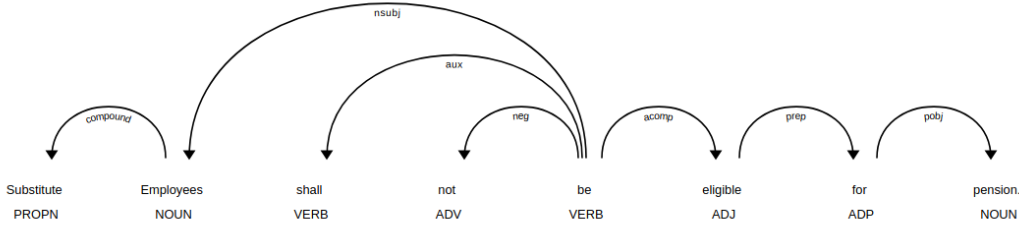
Country-Capital

Social Science Applications (e.g. Kozlowski, Evans, and Taddy ASR 2019)



Linguistics: Syntactic Parsing (e.g. Ash et al 2020)

- Syntactic parsers identify **dependencies** – functional **relations** between words in the sentence:



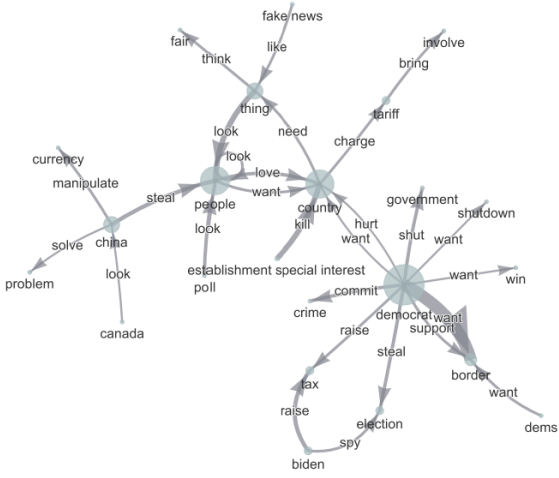
- e.g. subject-verb (who does what), verb-object (what to whom?)

- In contracts, modal verbs impose legal requirements:
  - strict (*shall*, *will*, *must*) modals express obligations/prohibitions.
  - permissive (*may*, *can*) modals express permissions.

Linguistics: Semantic Parsing (e.g. Ash et al 2022)

- Semantic role labeling (SRL) recovers characters and connections:
  - `ARG0`: *Who?* $\rightarrow$ Agents
  - `V`: *Does what?* $\rightarrow$ Verbs
  - `ARG1`: *To whom?* $\rightarrow$ Patients

Narrative Graph, Trump Tweet Archive



sites.google.com/view/trump-narratives