



PennState
College of the
Liberal Arts

C-SoDA
Center for Social Data Analytics

Day 1 - Introduction

Advanced Text as Data: Natural Language Processing
Essex Summer School in Social Science Data Analysis

Burt L. Monroe (Instructor) & Sam Bestvater (TA)
Pennsylvania State University

July 26, 2021

Why are we here?

Amazing advances in technology for understanding and generating natural language suggest that NLP has more to offer the social scientific study of “text-as-data.”

We’re going beyond the bag-of-words...

Facebook AI Creates Its Own Language In Creepy Preview Of Our Potential Future

Computers can now describe images using language you'd understand

The AI Text Generator That's Too Dangerous to Make Public

Researchers at OpenAI decided that a system that scores well at understanding language could too easily be manipulated for malicious intent.

How AI Can Create And Detect Fake News

A.I. breakthroughs in natural-language processing are big for business

BY JEREMY KAHN

Barbie Wants to Get to Know Your Child

With the help of A.I., America’s most famous doll tries to fulfill a timeless dream — convincing little girls that she’s a real friend. What will happen if they believe her?

Image source: Julia Hockenmaier, 2020

Maybe, I thought, I can get AI to teach the course!

Maybe, I thought, I can get AI to teach the course!
So I asked the model GPT-2 to tell me what you might be concerned about.

Maybe, I thought, I can get AI to teach the course!
So I asked the model GPT-2 to tell me what you might be concerned about.

<https://transformer.huggingface.co/doc/gpt2-large>

The students were apprehensive about the first day of class with Professor Monroe.
They had heard he was|

Maybe, I thought, I can get AI to teach the course!

So I asked the model GPT-2 to tell me what you might be concerned about.

<https://transformer.huggingface.co/doc/gpt2-large>

The students were apprehensive about the first day of class with Professor Monroe.

They had heard he was a bit of a crank, and they were still not certain if he was even a professor. The professor made no attempt to help the students, instead he sat down, crossed his arms, and started lecturing. "First, let me begin by acknowledging that you're the first person to attend this course. I'd like you to be the first to know that I have no interest in your particular subject. As such, you're the first one who will be going to any classes , or any lectures. We're all here for the learning, and I'd like you to help us learn it."

Maybe, I thought, I can get AI to teach the course!

So I asked the model GPT-2 to tell me what you might be concerned about.

<https://transformer.huggingface.co/doc/gpt2-large>

The students were apprehensive about the first day of class with Professor Monroe.

They had heard he was a bit of a crank, and they were still not certain if he was even a professor. The professor made no attempt to help the students, instead he sat down, crossed his arms, and started lecturing. "First, let me begin by acknowledging that you're the first person to attend this course. I'd like you to be the first to know that I have no interest in your particular subject. As such, you're the first one who will be going to any classes , or any lectures. We're all here for the learning, and I'd like you to help us learn it."

Hmmm ... I guess we do need a human at the wheel.

Maybe, I thought, I can get AI to teach the course!
So I asked the model GPT-2 to tell me what you might be concerned about.

<https://transformer.huggingface.co/doc/gpt2-large>

The students were apprehensive about the first day of class with Professor Monroe. They had heard he was a bit of a crank, and they were still not certain if he was even a professor. The professor made no attempt to help the students, instead he sat down, crossed his arms, and started lecturing. "First, let me begin by acknowledging that you're the first person to attend this course. I'd like you to be the first to know that I have no interest in your particular subject. As such, you're the first one who will be going to any classes , or any lectures. We're all here for the learning, and I'd like you to help us learn it."

Hmmm ... I guess we do need a human at the wheel.

Although my wife says this is flawless proof that it works.

Couldn't resist another. This one from GPT-3.

Human Prompt (given to GPT-3)

**At the party, I poured myself a glass of lemonade,
but it turned out to be too sour, so I added a little sugar.
I didn't see a spoon handy, so I stirred it with a cigarette.
But that turned out to be a bad idea because**

[GPT-3's generated continuation]

**it kept falling on the floor. That's when he decided to start
the Cremation Association of North America, which has become
a major cremation provider with 145 locations.**

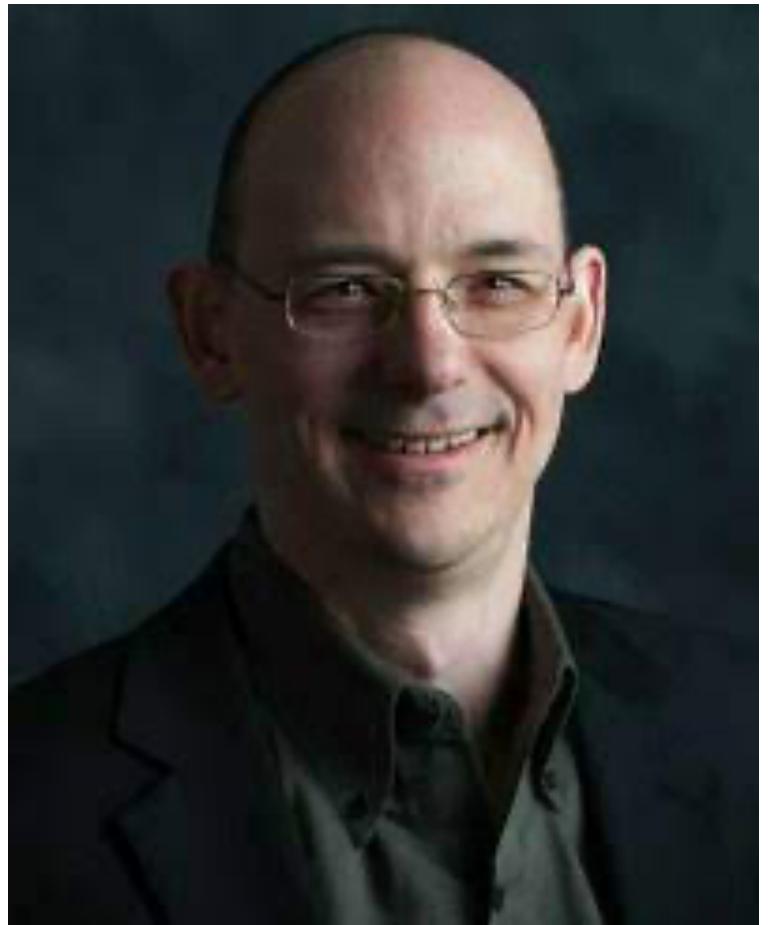
from Marcus & Davis '20

<https://www.technologyreview.com/2020/08/22/1007539/gpt3-openai-language-generator-artificial-intelligence-ai-opinion/>

Today

- Introductions
 - Course Logistics
 - Overview - “Text as Data” and/vs. NLP
 - Modern/neural NLP
-
- Demos: R in RStudio Cloud, Python in Google Colab

Introductions



Instructor: Burt Monroe
burtmonroe@psu.edu



TA: Sam Bestvater
seb64@psu.edu

Course Schedule Page

- <https://burtmonroe.github.io/TextAsDataCourse/Essex/>
- Readings (and any changes)
- Slides for that day's lecture.
- R and/or Python materials to be covered that day (more on that in a minute).
 - R materials will be in the class space on RStudioCloud.
 - Python materials will be on Google Colab.

Software

- We will be demonstrating various techniques in R and Python.
- Where sensible and feasible, we have tried to provide notebooks in both. This isn't entirely practical for all of the material. You will, in particular, notice that more neural net / deep learning examples are in Python than R.
- R material, in the form of notebooks, is provided through Essex's RStudio Cloud account. This will allow you to modify and run copies of the notebooks without installation, versioning, memory, etc. worries on your own machines. (If you do try to run on your own machines, be careful/deliberate about installations of things like RTools/devtools and Python / Java / etc. that might be required for some code.).
- Python material, also in notebooks is provided through Google Colab. Again, this allows you to modify and run copies of the notebooks without installation worries on your own machines. Further, Google Colab provides free access to GPU and TPU computing, which can be the difference in training neural nets between "that wasn't too bad" and "I'm going to have grandchildren before this finishes."

Stuff I am assuming you already know, more or less

- Basics of string manipulation / regular expressions (in R &/or Python)
- Basic text data management, preprocessing and creation of bag-of-words / doc-term matrix data.
- Cosine similarity.
- Building, training, evaluating and interpreting a “classical” (non-neural) text classifier [e.g., Naive Bayes, logistic regression / lasso / ridge regression, support vector machine, random forest, XGBoost] based on a bag-of-words representation.
- Building, training, evaluating and interpreting a topic model [e.g., LDA, CTM, STM] (based on a bag-of-words representation).
- I also won’t be going over any detail about data collection and cleaning – things like web scraping, OCR, and pdfs. (I do have some tutorials available.)

Bridging Text-as-Data and NLP – Beyond the “Bag of Words”

- Not just “words” – changing the things in the bag. (New or more targeted features.)
- Not discrete words, but infinite shades of meaning. (e.g., Embeddings)
- The bags are filled with different sets of words (e.g., Multilingual text)
- All the bags – leveraging what we’ve learned from *other* related bags. (Transfer learning)
- Not just a bag – getting words from a pipe or finding them on a network. (Better models of language production – more accurate measurement, more validity. Language generation.)

Bridging Text-as-Data and NLP – Beyond the “Bag of Words”

- Doing similar tasks differently and hopefully better (more validity, greater accuracy, less bias, more fairness).
 - New ways to turn text into data, which we analyze the same way.
 - New ways to analyze text-based data. (e.g., deep learning / neural NLP)
- Doing similar tasks more cheaply, at greater scale, at broader scope, in more depth. Increasing / democratizing the accessibility of tasks to more people.
- Doing new types of tasks.

What we *will* cover

- NLP “pipelines”. General concepts and practical application of NLP labeling tasks like part-of-speech tagging, named entity recognition, and dependency parsing. (Day 2)
- Word embeddings - what, why and how. How to use them. How to estimate them. (Day 3)
- Deep learning / neural NLP - General neural network concepts, feedforward networks, building and training models (mainly in Keras/Tensorflow). (Days 4 & 5)
- Recurrent & convolutional neural nets, attention and transformers (Day 6)
- Contextual embeddings, pretrained language models, and transfer learning (BERT and similar) (Day 7)
- Multilingual text as data, machine translation (Day 8)
- Natural language understanding (NLU) & Natural language generation (NLG) (Day 9)
- Fairness & bias in NLP (Day 10)

Text as Data

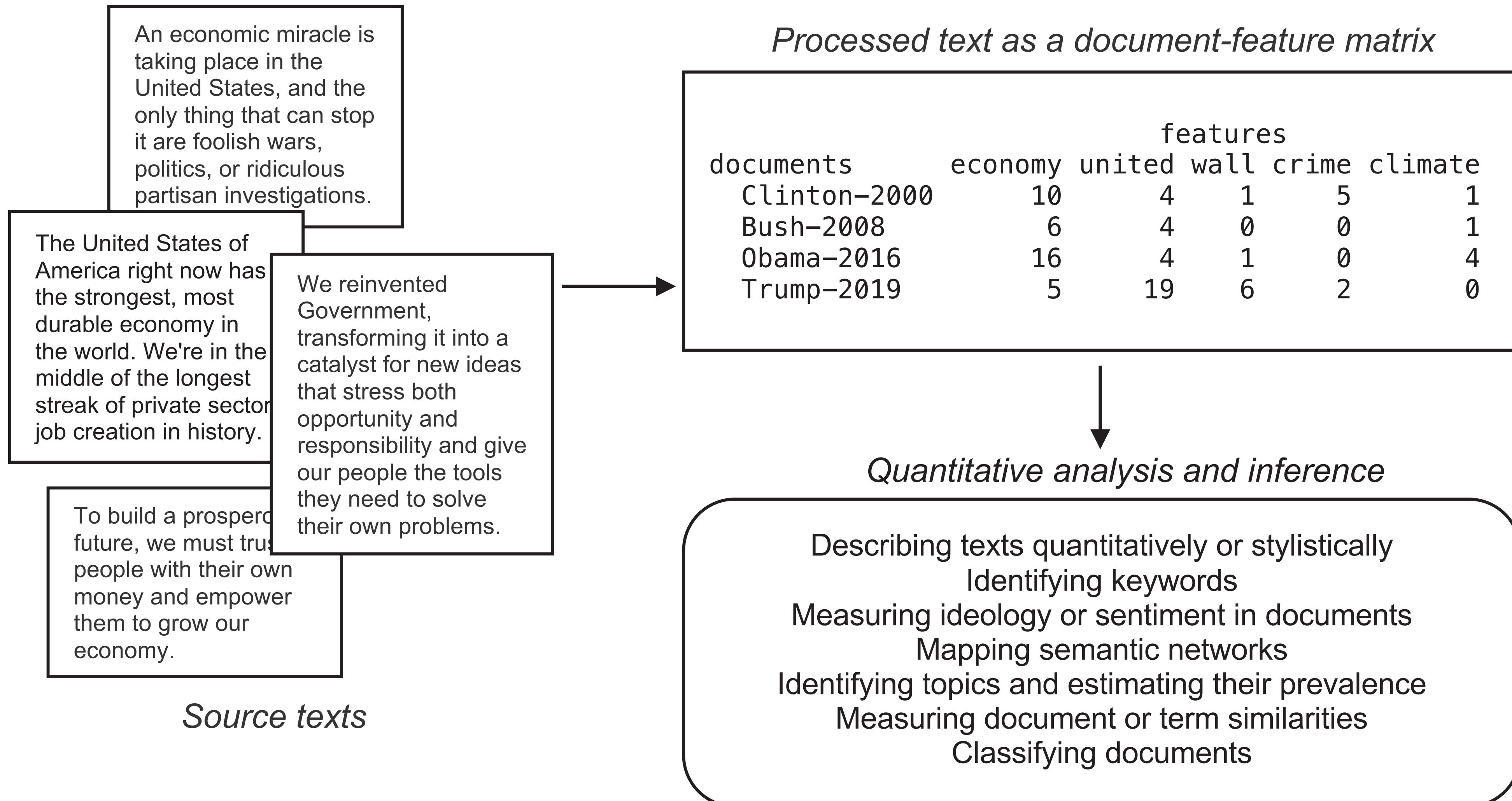


Figure 26.1 From text to data to data analysis

Source: Benoit. 2020

Common goals & methods in text as data

2

Justin Grimmer and Brandon M. Stewart

- Quantitative analysis and inference**
- Describing texts quantitatively or stylistically
 - Identifying keywords
 - Measuring ideology or sentiment in documents
 - Mapping semantic networks
 - Identifying topics and estimating their prevalence
 - Measuring document or term similarities
 - Classifying documents

Source: Benoit 2020.

Most text-as-data tasks are
latent measurement tasks

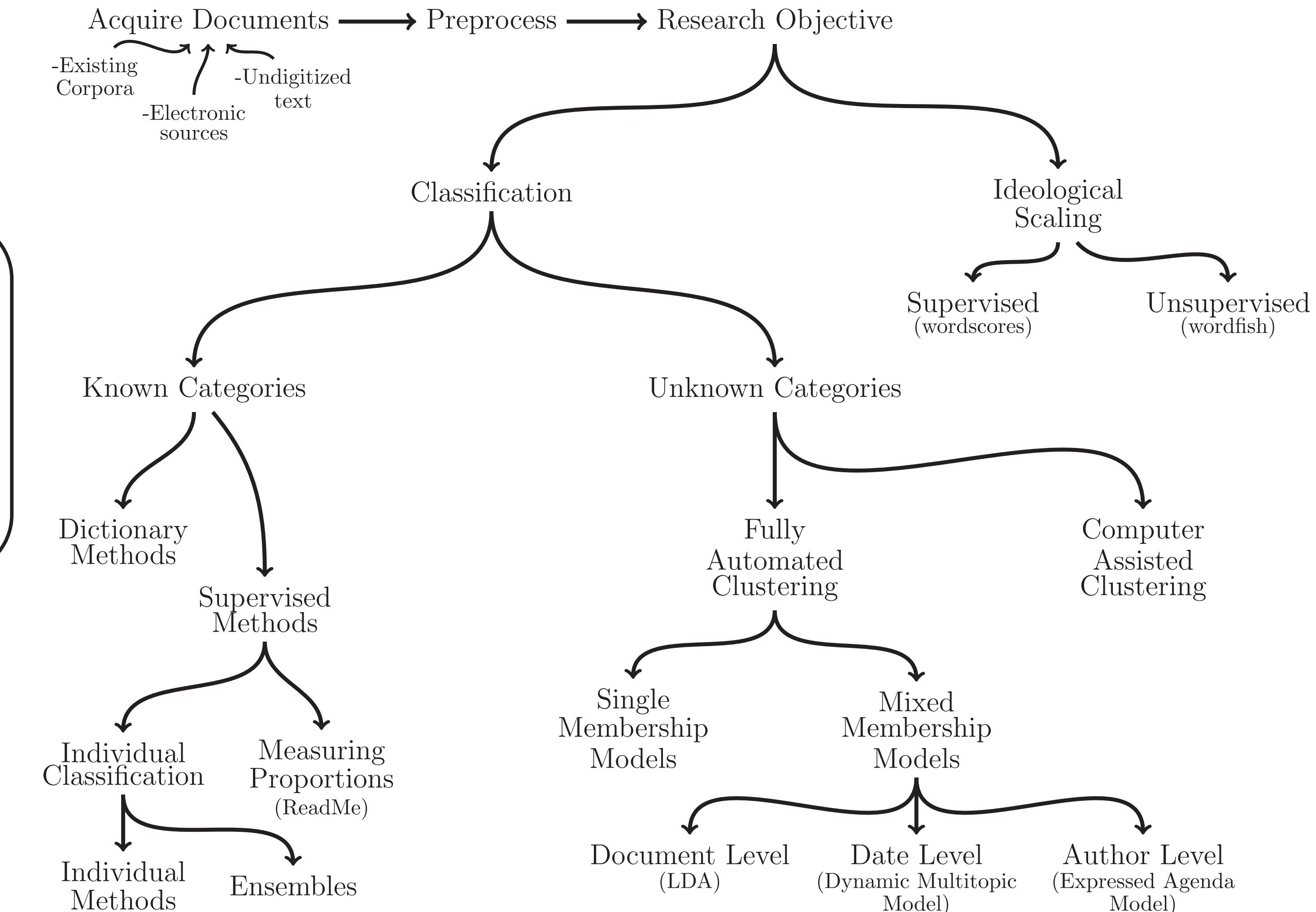


Fig. 1 An overview of text as data methods.

Source: Grimmer and Stewart, 2013.

Common goals & methods in text as data

2

Justin Grimmer and Brandon M. Stewart

Quantitative analysis and inference

- Describing texts quantitatively or stylistically can be
 - Identifying keywords can be
 - Measuring ideology or sentiment in documents
 - Mapping semantic networks
- Identifying topics and estimating their prevalence
- Measuring document or term similarities can be
 - Classifying documents

Source: Benoit 2020.

Most text-as-data tasks are
latent measurement tasks

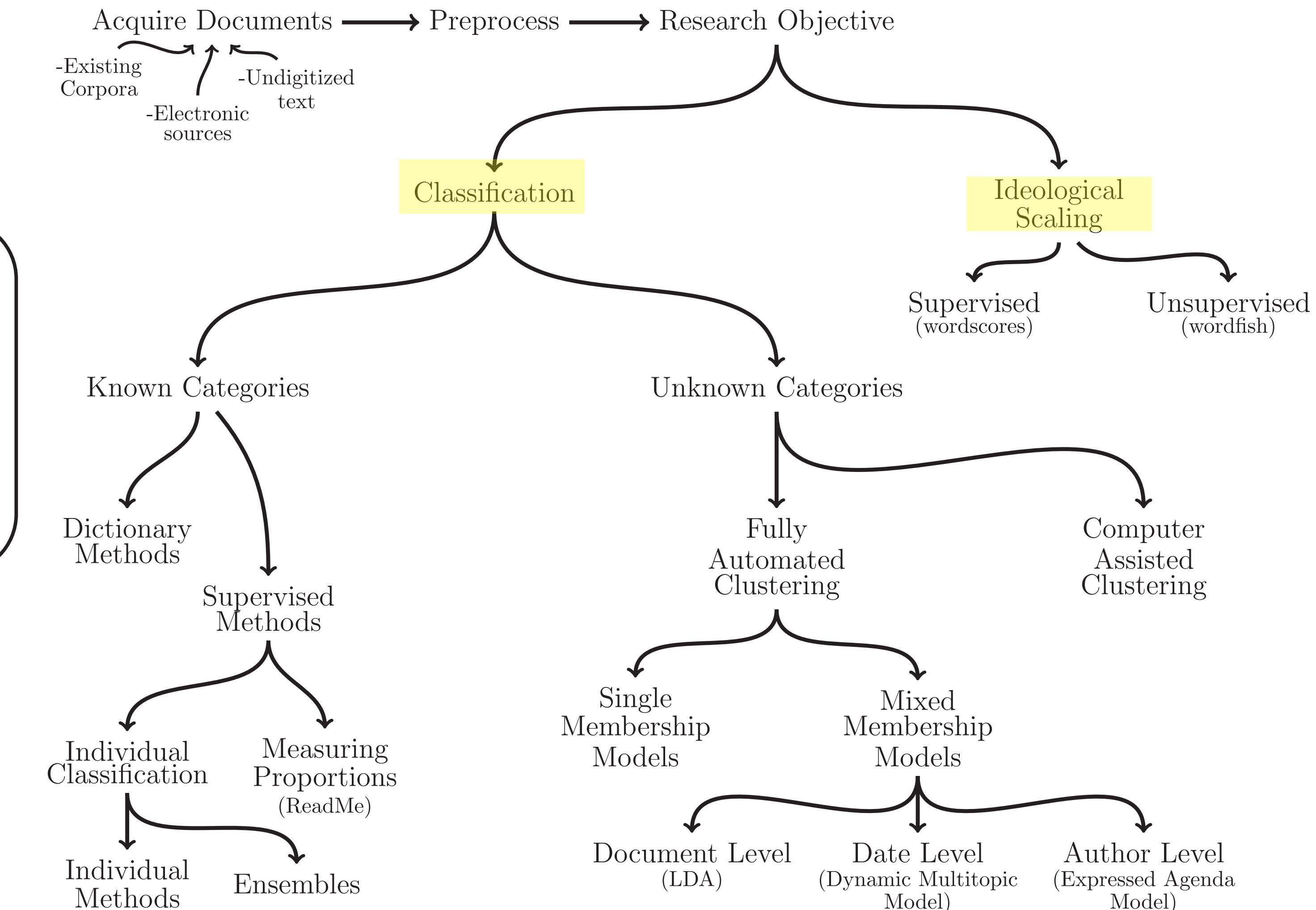


Fig. 1 An overview of text as data methods.

Source: Grimmer and Stewart, 2013.

This typical text-as-data pipeline converts text to numeric data through “pre-processing” that

- (1) divides the text into the units we might care about,
- (2) throws out some we’re pretty sure we don’t care about, and
- (3) combines units that we’re pretty sure should count as the same thing.

Tomorrow, we will look at NLP pipelines that do more elaborate and language-aware processing.

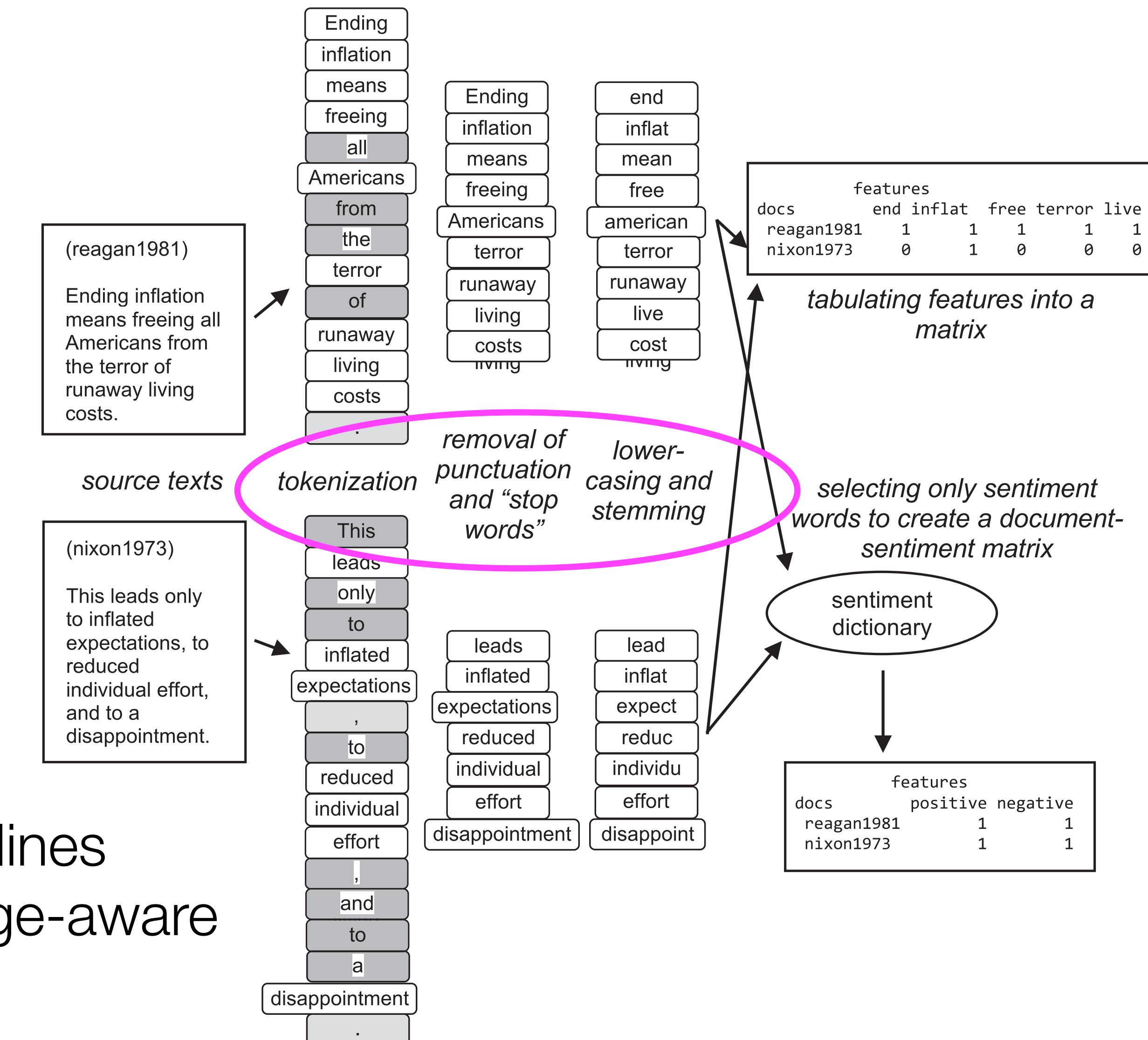


Figure 26.2 From text to tokens to matrix

Figure source: Benoit. 2020

Dictionary Methods

- Most basic - you have a list of words that count, or have a predetermined “score” for each category and you have the computer count/add them up.
- Very popular approach to sentiment analysis.
 - VADER
 - Lexicoder
 - LIWC
 -

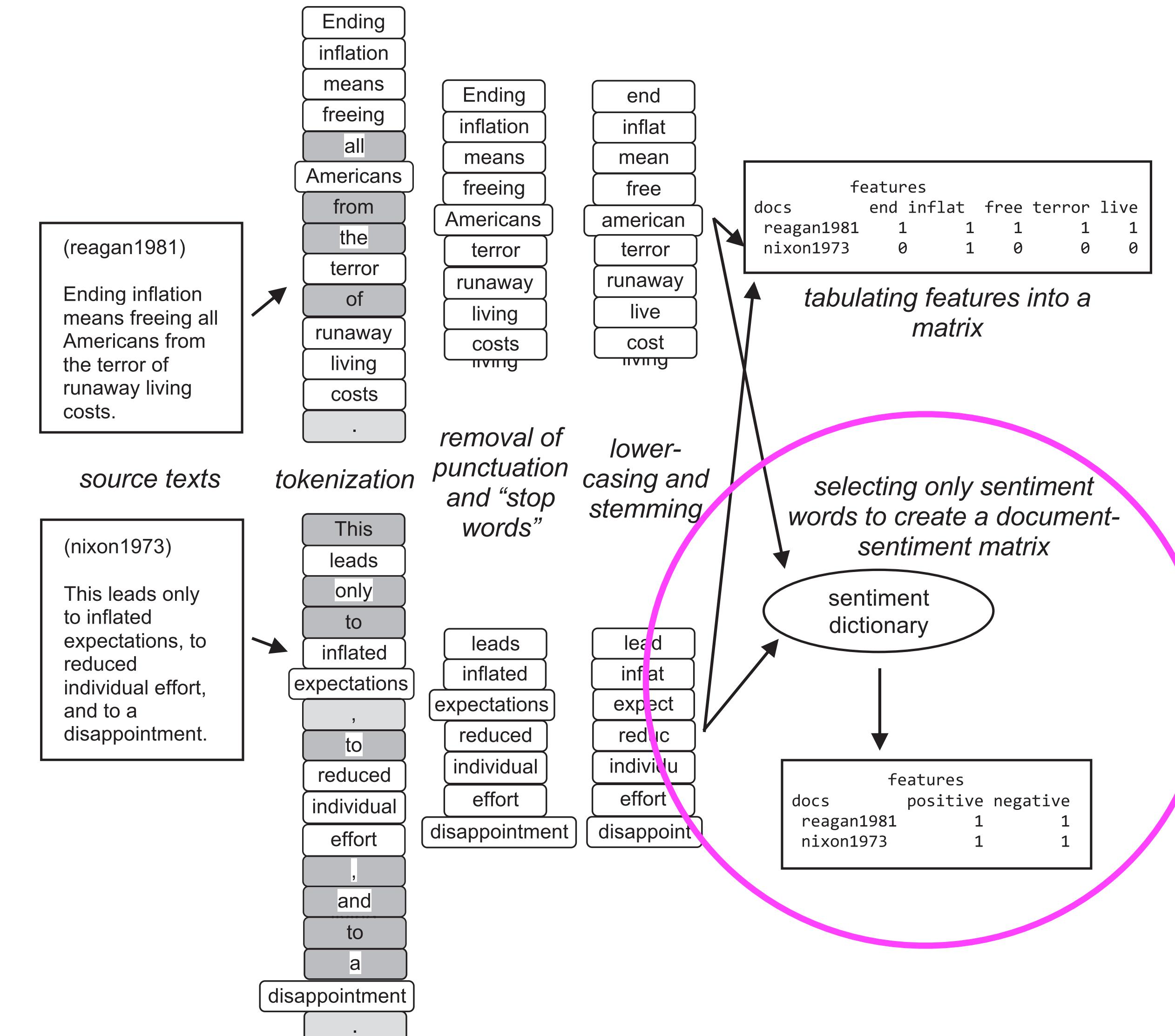


Figure 26.2 From text to tokens to matrix

Figure source: Benoit. 2020

Dictionary Methods

- Dictionary-based sentiment analysis is popular because it's easy, not because it's good.
- Consider this pair of sentences from JFK's optimism-filled inaugural speech, as coded by Lexicoder.
 -

```
dfm("Let us never negotiate out of fear. But let us never fear to negotiate.",  
    tolower = TRUE,      # casifold  
    stem = FALSE,        # do not stem  
    remove_punct = TRUE, # remove punctuation  
    dictionary = data_dictionary_LSD2015)
```

Document-feature matrix of: 1 document, 4 features (75.0% sparse).

1 x 4 sparse Matrix of class "dfm"

	features	negative	positive	neg_positive	neg_negative
docs					
text1		2	0	0	0

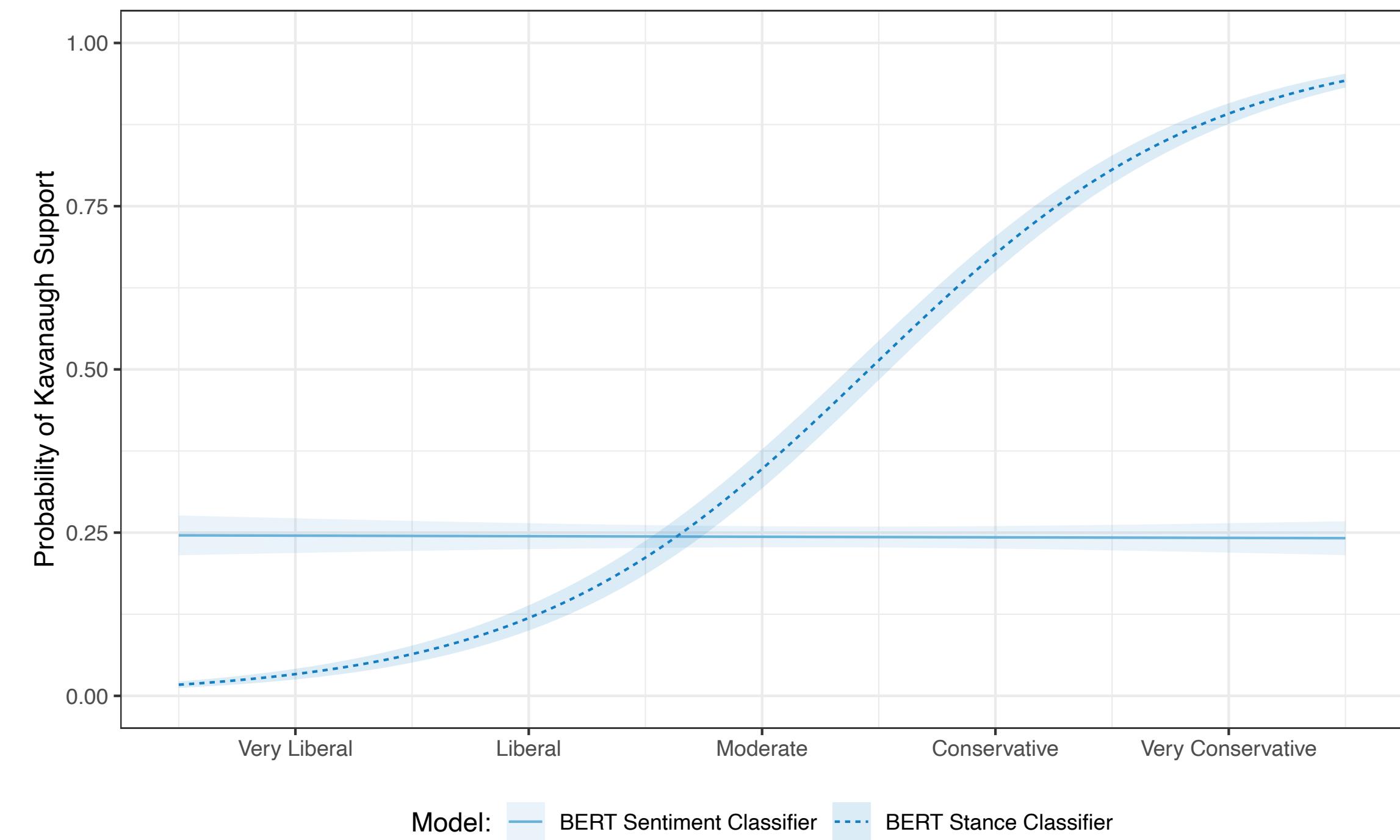
Supervised Learning - Classification

Table 6. Classifier Performance: Kavanaugh Tweets

Classifier	F1 Score (Predicting Sentiment)	F1 Score (Predicting Stance)
Lexicoder	0.788 (0.005)	0.572 (0.014)
VADER	0.754 (0.005)	0.514 (0.011)
SVM (sentiment-trained)	0.943 (0.003)	0.514 (0.012)
BERT (sentiment-trained)	0.954 (0.002)	0.582 (0.005)
SVM (stance-trained)		0.935 (0.006)
BERT (stance-trained)		0.938 (0.002)

Reported figures are the average F1 score over 5-fold cross validation
Standard Errors in parentheses

Figure 5. Predicting Kavanaugh Support from Ideology



Source: Bestvater and Monroe, 2020.

Interpreting differences in classes

Most Important Problems, Group-based Generation Label
McCourtney Mood of the Nation Poll



Interpreting differences in classes

Most Important Problems, Group-based Generation Label

McCourtney Mood of the Nation Poll



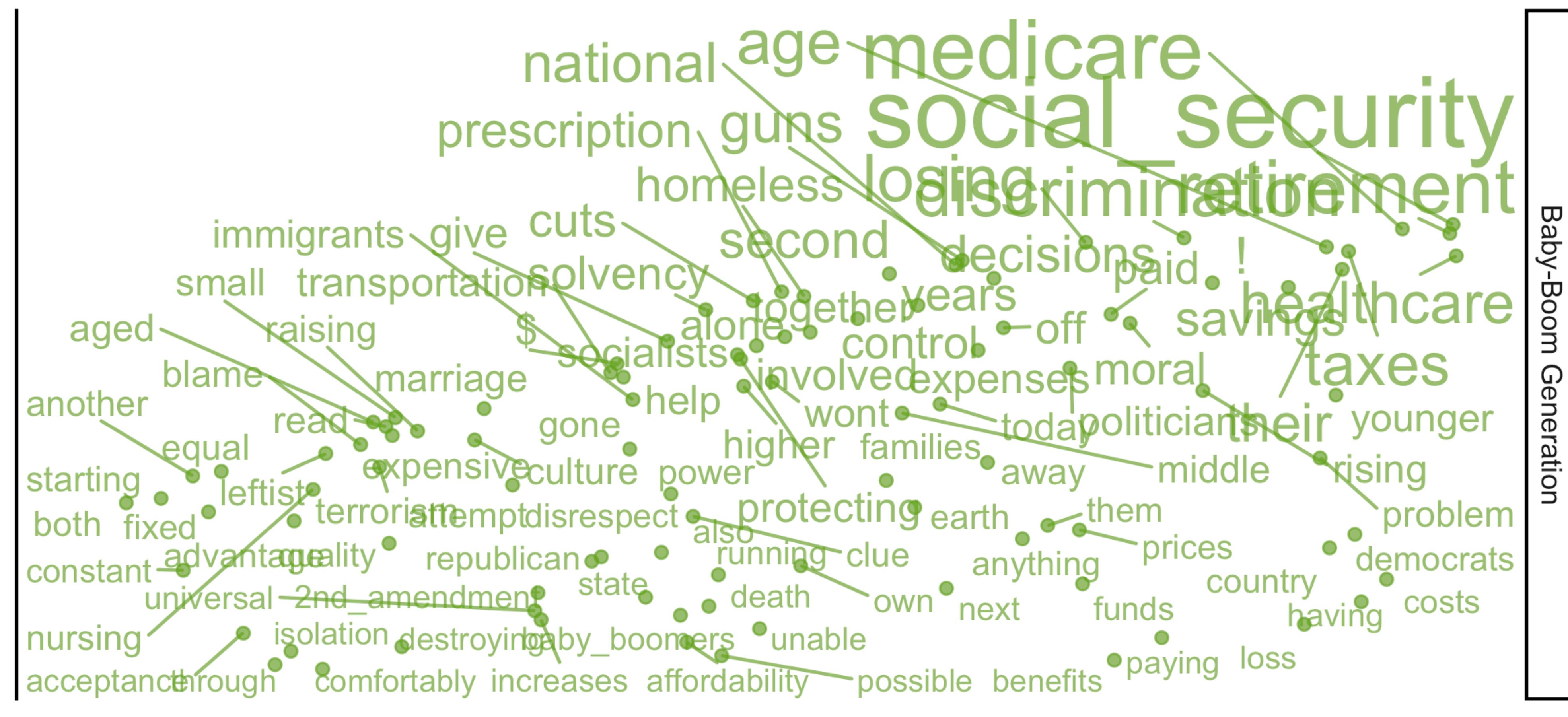
Interpreting differences in classes

Most Important Problems, Group-based Generation Label
McCourtney Mood of the Nation Poll



Interpreting differences in classes

Most Important Problems, Group-based Generation Label
McCourtney Mood of the Nation Poll



Interpreting differences in classes

Most Important Problems, Group-based Generation Label
McCourtney Mood of the Nation Poll



Unsupervised Learning

- We don't have labeled examples (documents) — no training data.
- This is often used as an exploratory / discovery exercise or for theory testing.
- Some of the main variants:
 - Clustering
 - Topic modeling
 - Scaling (e.g., on a left-right scale from liberal to conservative)
 - Word embeddings

Unsupervised Learning - Topic Modeling

- Unlike a classification problem, you don't have pre-defined categories.
- But one main output of a topic model is a measure of how words are associated with the topic.
- When a topic model really works well, the appropriate label is obvious.
- A model of speeches in the US Senate contained the topic to the right ... what label would you give it?
 - .

school
teacher
educ
student
children
test
local
learn
district
class
account
classroom
achiev
teach
better

Table 4: *Key Words on*

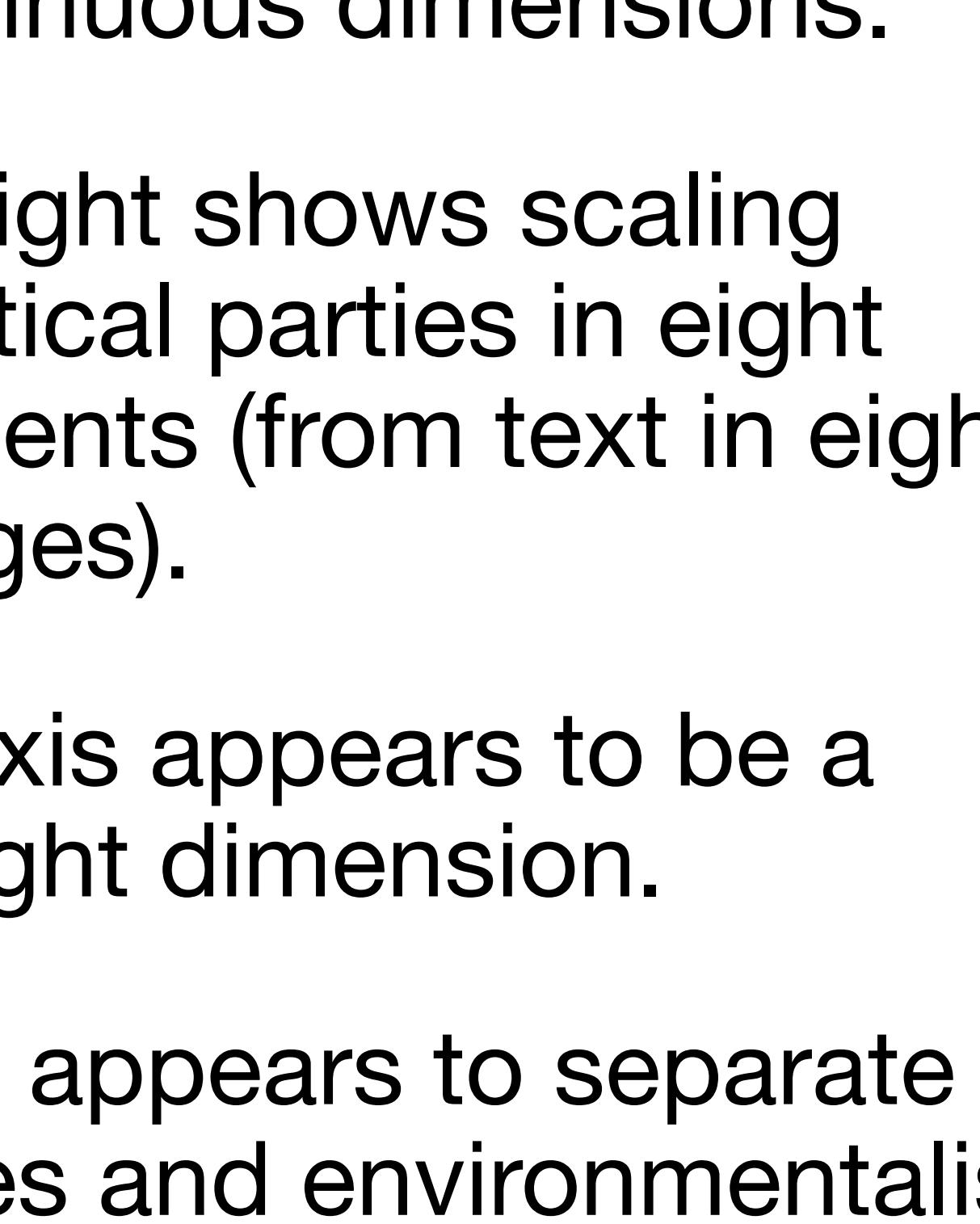
Unsupervised Learning - Topic Modeling

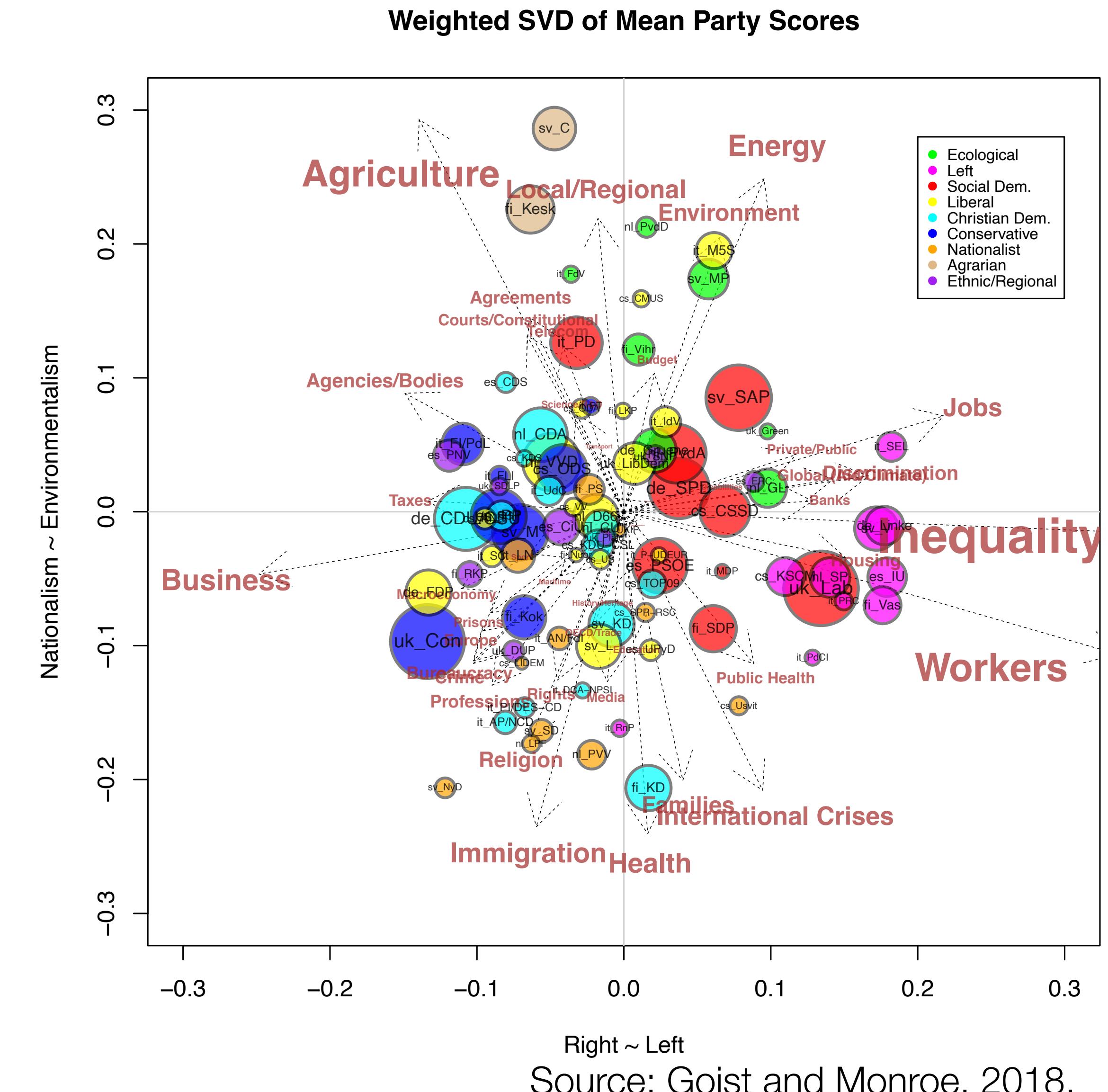
- Unlike a classification problem, you don't have pre-defined categories.
- But one main output of a topic model is a measure of how words are associated with the topic.
- When a topic model really works well, the appropriate label is obvious.
- A model of speeches in the US Senate contained the topic to the right ... what label would you give it?
-

school
teacher
educ
student
children
test
local
learn
district
class
account
classroom
achiev
teach
better

Table 4: *Key Words on 'Education'.*

Unsupervised Learning - Scaling

- Here, the target isn't categories but a location on continuous dimensions.
 - The plot to the right shows scaling locations of political parties in eight different parliaments (from text in eight different languages).
 - The horizontal axis appears to be a traditional left-right dimension.
 - The vertical axis appears to separate nationalist parties and environmentalist parties



Natural Language Processing (NLP)

Tasks in NLP

- Some break “NLP” into “NLU”, “NLG,” and the rest (back to “NLP”). These overlap.
- Core “NLP” is often traditionally thought of as a pipeline for “annotation” or “sequence labeling.” Text flows through the connected segments, emerging from each with some added embellishment. Tokenization/segmentation, morphological analysis/lemmatization, POS-tagging, word sense disambiguation, syntactic parsing, semantic parsing (NLU?), named-entity recognition, coreference resolution (NLU?)
- NLU - Natural language understanding: extract information, translate raw text into a representation of meaning, reason about information given in the text. Classification, sentiment analysis, relationship extraction, reading comprehension, distributed representations/embeddings.
- NLG - Natural language generation: convert meaning representations into natural language text or text-to-text, produce appropriate utterances/responses in a dialog or Q&A (NLU?), summarize (NLU?), machine translate, style transfer, caption images (image-to-text).

Tasks in NLP, NLU, and NLG

- Some break “NLP” into “NLU”, “NLG,” and the rest (back to “NLP”). These overlap.
- Core “NLP” is often traditionally thought of as a pipeline for “annotation” or “sequence labeling.” Text flows through the connected segments, emerging from each with some added embellishment. Tokenization/segmentation, morphological analysis/lemmatization, POS-tagging, word sense disambiguation, syntactic parsing, semantic parsing (NLU?), named-entity recognition, coreference resolution (NLU?) **There are a lot of objectives overlapping with “text-as-data” in there.**
- NLU - Natural language understanding: extract information, translate raw text into a representation of meaning, reason about information given in the text. Classification, sentiment analysis, relationship extraction, reading comprehension, distributed representations/embeddings.
- NLG - Natural language generation: convert meaning representations into natural language text or text-to-text, produce appropriate utterances/responses in a dialog or Q&A (NLU?), summarize (NLU?), machine translate, style transfer, caption images (image-to-text).

Online demos of a handful of NLP tasks (feel free to poke around).

How much “understanding” is happening?

Tokenization

<https://spacy.io/usage/linguistic-features#tokenization>

The screenshot shows a web browser window with the URL <https://spacy.io/usage/linguistic-features#tokenization>. The page is titled "Linguistic Features · spaCy Usage". On the left, there's a sidebar with links like "Installation", "Models & Languages", "Facts & Figures", "spaCy 101", "New in v3.0", and "New in v3.1". Under "GUIDES", the "Linguistic Features" section is expanded, showing sub-links: POS Tagging, Morphology, Lemmatization, Dependency Parse, Named Entities, Entity Linking, Tokenization (which is highlighted in dark blue), Merging & Splitting, Sentence Segmentation, Vectors & Similarity, Mappings & Exceptions, and Language Data. Below this, there are links for Rule-based Matching, Processing Pipelines, Embeddings & Transformers (marked as "NEW"), Training Models (marked as "NEW"), Layers & Model Architectures (marked as "NEW"), spaCy Projects (marked as "NEW"), Saving & Loading, and Visualizers.

The main content area starts with a paragraph about punctuation rules. It then shows an "Editable Code" block with Python code for tokenizing a sentence:

```
import spacy

nlp = spacy.load("en_core_web_sm")
doc = nlp("Apple is looking at buying U.K. startup for $1 billion")
for token in doc:
    print(token.text)
```

A "RUN" button is present below the code. To the right, the output of the code is shown as a list of tokens: Apple, is, looking, at, buying, U.K., startup, for, \$, 1, billion. A horizontal slider at the bottom allows for navigating through these tokens. The background of the page features a blue circuit board pattern.

Annotation: named entity recognition

<https://explosion.ai/demos/displacy-ent>

The screenshot shows a web browser window for the [displaCy Named Entity Visualizer](https://explosion.ai/demos/displacy-ent). The URL is visible in the address bar. The page has a dark blue header with the title "displaCy Named Entity Visualizer". On the left, there is a text input field containing a paragraph about Sebastian Thrun. Below it is a dropdown menu labeled "Model" set to "English - en_core_web_sm (v2.3.0)". To the right of the text input is a section titled "Entity labels (select all)" with a grid of checkboxes for various entity types: PERSON (checked), NORP, ORG, GPE, LOC, PRODUCT, EVENT, WORK OF ART, LANGUAGE, DATE (checked), TIME, PERCENT, MONEY, QUANTITY, ORDINAL, and CARDINAL. The main content area displays the annotated text with colored boxes around the identified entities: "Sebastian Thrun" (PERSON), "Google" (ORG), "2007" (DATE), "American" (NORP), "Thrun" (ORG), and "earlier this week" (DATE). Below the text, a note discusses the growth of self-driving startups. At the bottom, there are links for "Using and customising NER models" and "displaCy Named Entity Visualizer".

When Sebastian Thrun started working on self-driving cars at Google in 2007, few people outside of the company took him seriously. “I can tell you very senior CEOs of major American car companies would shake my hand and turn away because I wasn’t worth talking to,” said Thrun, now the co-founder and CEO of online higher education startup Udacity, in an interview with Recode earlier this week.

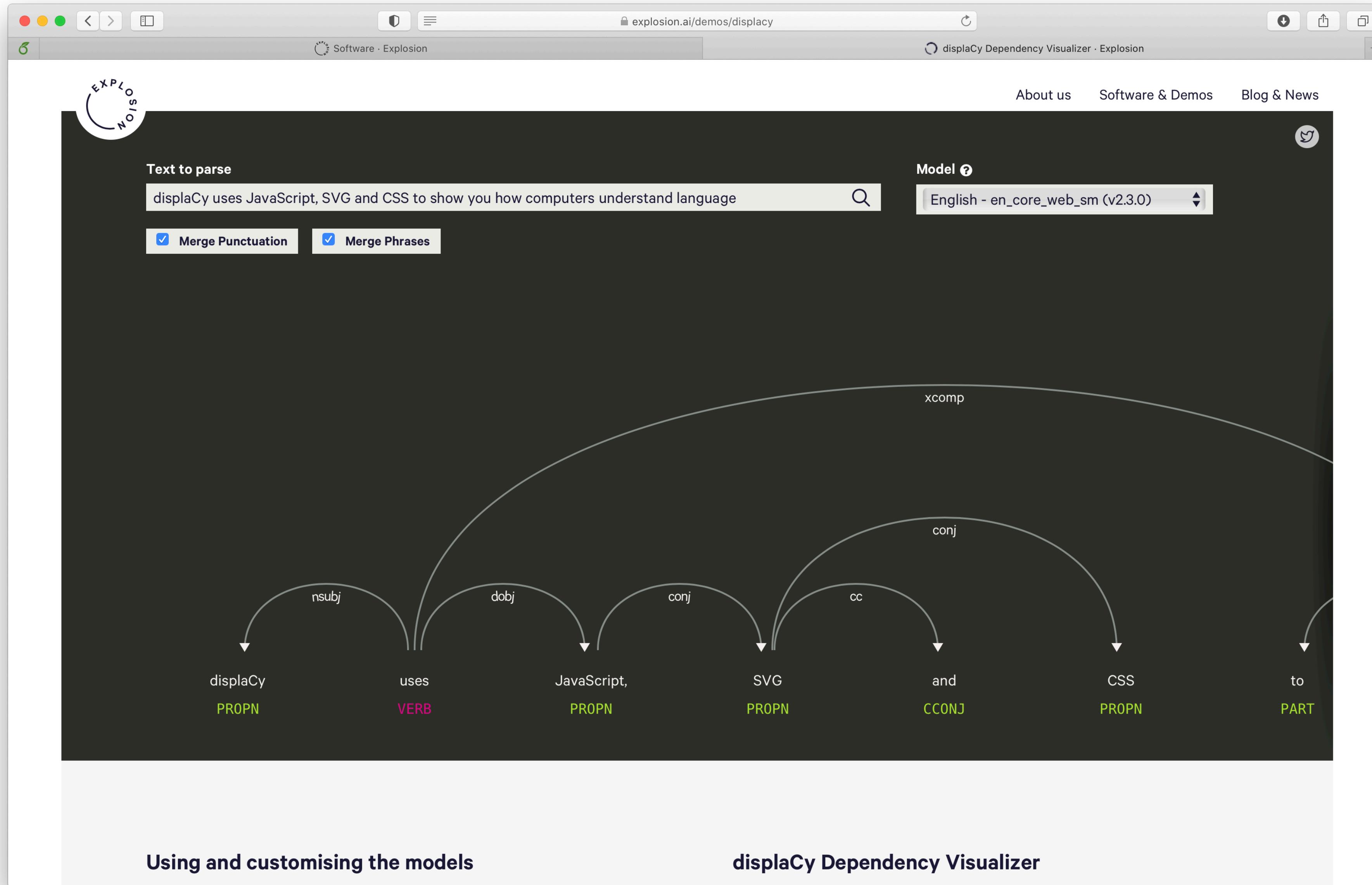
A little less than a decade later, dozens of self-driving startups have cropped up while automakers around the world clamor, wallet in hand, to secure their place in the fast-moving world of fully automated transportation.

Using and customising NER models

displaCy Named Entity Visualizer

Dependency parsing

<https://explosion.ai/demos/displacy>



Dependency parsing

<https://demo.allennlp.org>

Screenshot of the AllenNLP Demo website showing a dependency parse for the sentence "James ate some cheese whilst thinking about the play." The interface includes a sidebar with various NLP tasks and a main area for input, model selection, and output visualization.

Example Inputs:
James ate some cheese whilst thinking about the play.

Sentence:
James ate some cheese whilst thinking about the play.

Run Model

Model Output:

James ate some cheese whilst thinking about the play .

Dependency Parse Diagram:

```
graph TD; James[James  
PROPN] -- NSUBJ --> ate[ate  
VERB]; cheese[some cheese  
DEP] --- whilst[whilst  
DEP]; thinking[thinking  
DEP] --- about[about  
DEP]; play[the play  
PUNCT]
```

Sidebar (Dependency Parsing selected):

- Answer a question
- Reading Comprehension
- Visual Question Answering
- Annotate a sentence
- Named Entity Recognition
- Open Information Extraction
- Sentiment Analysis
- Dependency Parsing**
- Constituency Parsing
- Semantic Role Labeling
- Annotate a passage
- Coreference Resolution
- Generate a passage
- Language Modeling
- Masked Language Modeling
- Compare two sentences
- Textual Entailment
- Evaluate Reading Comprehension

Share

Information extraction

<https://demo.allennlp.org>

The screenshot shows the AllenNLP demo interface for Open Information Extraction. The left sidebar lists various NLP tasks: Answer a question, Reading Comprehension, Visual Question Answering, Annotate a sentence, Named Entity Recognition, Open Information Extraction (selected), Sentiment Analysis, Dependency Parsing, Constituency Parsing, Semantic Role Labeling, Annotate a passage, Coreference Resolution, Generate a passage, Language Modeling, Masked Language Modeling, Compare two sentences, Textual Entailment, and Evaluate Reading Comprehension. The main content area displays the Open Information Extraction model's demo page. It includes a brief description, navigation tabs for Demo, Model Card, and Model Usage, and sections for Example Inputs and Sentence. A "Run Model" button is present. The Model Output section shows 1 Total Extractions for the verb "published". The extraction details are: Albert Einstein , a German theoretical physicist (ARGO), published (V), the theory of relativity (ARG1), in 1915 (ARGM-TMP). The CLI Output section includes a link to "What is this?". A "Share" button is located in the Model Output section.

AI2 Allen Institute for AI

AllenNLP

Answer a question

Reading Comprehension

Visual Question Answering

Annotate a sentence

Named Entity Recognition

Open Information Extraction

Sentiment Analysis

Dependency Parsing

Constituency Parsing

Semantic Role Labeling

Annotate a passage

Coreference Resolution

Generate a passage

Language Modeling

Masked Language Modeling

Compare two sentences

Textual Entailment

Evaluate Reading Comprehension

demo.allennlp.org/open-information-extraction

AllenNLP - Demo

Open Information Extraction

A reimplemention of a deep BiLSTM sequence prediction model (Stanovsky et al., 2018).

Demo Model Card Model Usage

Example Inputs

Albert Einstein, a German theoretical physicist, published the theory of relativity in 1915.

Sentence

Albert Einstein, a German theoretical physicist, published the theory of relativity in 1915.

Run Model

Model Output

1 Total Extractions

Extractions for **published**:

Albert Einstein , a German theoretical physicist ARG0 , published V the theory of relativity ARG1 in 1915 ARGM-TMP .

CLI Output [What is this?](#)

> Input

Share

Coreference resolution: annotation?

<https://demo.allennlp.org>

The screenshot shows the AllenNLP Coreference Resolution demo page. On the left, a sidebar lists various NLP models: Answer a question (Reading Comprehension, Visual Question Answering), Annotate a sentence (Named Entity Recognition, Open Information Extraction, Sentiment Analysis, Dependency Parsing, Constituency Parsing, Semantic Role Labeling), Annotate a passage (Coreference Resolution, Generate a passage, Language Modeling, Masked Language Modeling), Compare two sentences, and Textual Entailment. The main content area is titled "Model" and "Coreference Resolution". It describes the model's basic outline: getting embedded representations for spans, scoring them, pruning unlikely ones, and then deciding antecedent spans to find coreferent clusters. Below this are tabs for "Demo" (which is selected), "Model Card", and "Model Usage". Under "Example Inputs", there is a text input field containing a sentence about Italy. Under "Document", there is a larger text block with the same sentence. A blue "Run Model" button is located between the inputs and outputs. The "Model Output" section shows the processed text with spans highlighted in boxes: "a region of central Italy bordering the Adriatic Sea" (blue box, index 0), "The area" (blue box, index 0), "Mt. Corno" (blue box, index 0), "the highest peak of the mountain range" (blue box, index 0), "many sheep" (pink box, index 1), and "them" (pink box, index 1). A "Share" button is in the top right of the output section. At the bottom, there is a "CLI Output" section with a link "What is this?".

Coreference resolution: NLU?

<https://demo.allennlp.org>

The screenshot shows the AllenNLP Coreference Resolution demo page. On the left, a sidebar lists various NLP models: Answer a question (Reading Comprehension, Visual Question Answering), Annotate a sentence (Named Entity Recognition, Open Information Extraction, Sentiment Analysis, Dependency Parsing, Constituency Parsing, Semantic Role Labeling), Annotate a passage (Coreference Resolution, Generate a passage, Language Modeling, Masked Language Modeling), and Compare two sentences (Textual Entailment, Evaluate Reading Comprehension). The 'Coreference Resolution' section is currently selected. The main content area has a dark blue header with the text 'AI2 Allen Institute for AI' and 'AllenNLP'. Below the header, the 'Model' section is titled 'Coreference Resolution' with a detailed description of the model's process. The 'Demo' tab is active, showing 'Example Inputs' with a dropdown menu 'Select a Document' containing 'Winograd schemas'. The 'Document' section displays the text: 'In the middle of the outdoor concert, the rain started falling, and it continued until 10. In the middle of the outdoor concert, the rain started falling, but it continued until 10. I poured water from the bottle into the cup until it was full. I poured water from the bottle into the cup until it was empty.' A large blue button labeled 'Run Model' is present. The 'Model Output' section shows the processed text with spans highlighted by color-coded boxes (pink, blue, purple, red, cyan) and numbered indices (1 through 10) placed next to them. A 'Share' button is located in the top right of this section. At the bottom, there is a 'CLI Output' section with a link 'What is this?'. The entire interface is set against a light gray background.

Textual Entailment

<https://demo.allennlp.org>

The screenshot shows the AllenNLP demo interface for Textual Entailment. On the left, a sidebar lists various NLP tasks: Answer a question, Annotate a sentence, Annotate a passage, Generate a passage, Compare two sentences, and Evaluate Reading Comprehension. The 'Textual Entailment' task is currently selected. The main area contains input fields for Premise ('An interplanetary spacecraft is in orbit around a gas giant's icy moon.') and Hypothesis ('The spacecraft has the ability to travel between planets.'), a 'Run Model' button, and a 'Model Output' section. The output states, 'It is **very likely** that the premise **entails** the hypothesis.' Below this is a probability chart showing the model's confidence in each judgement: Entailment (98.5%), Contradiction (0.1%), and Neutral (1.4%). A decorative triangle graphic is visible at the bottom.

Example Inputs

An interplanetary spacecraft is in orbit around a gas giant's icy moon.

Premise

An interplanetary spacecraft is in orbit around a gas giant's icy moon.

Hypothesis

The spacecraft has the ability to travel between planets.

Run Model

Model Output

It is **very likely** that the premise **entails** the hypothesis.

Share

Judgement	Probability
Entailment	98.5%
Contradiction	0.1%
Neutral	1.4%

E

C

N

Judgement

Probability

Entailment

Contradiction

Neutral

AI2 Allen Institute for AI

AllenNLP

Answer a question

Reading Comprehension

Visual Question Answering

Annotate a sentence

Named Entity Recognition

Open Information Extraction

Sentiment Analysis

Dependency Parsing

Constituency Parsing

Semantic Role Labeling

Annotate a passage

Coreference Resolution

Generate a passage

Language Modeling

Masked Language Modeling

Compare two sentences

Textual Entailment

Evaluate Reading Comprehension

The path we'll take through NLP

- Traditionally, abstract NLP tasks were preceded by pipeline annotation. This differs from, or is only loosely approximated by, the conventional “preprocessing” we’ve seen in text-as-data tasks. Tomorrow (DAY 2) we’ll explore what those pipelines look like, what problems they solve, and how/why we might incorporate them into our text-as-data analyses.
- Day 3 will focus on an area that overlaps TADA and NLP: word embeddings. Word embeddings share a lot of similarities with unsupervised topic model and scaling methods.
- Contemporary NLP, as with AI generally, is dominated now by neural network / deep learning approaches. These often sidestep the traditional NLP pipeline, using minimally preprocessed tokenized text as inputs and training directly on the task at hand, for “end-to-end” analysis. We’ll spend most of our time, from Day 4 to the end, on neural NLP.

Tomorrow

- NLP Pipelines / core NLP tasks
 - Tokenization / segmentation
 - Normalization / lemmatization / stemming / morphology
 - Sequence labeling – part-of-speech tagging, named entity recognition
 - Dependency parsing
- Demos: NLP pipelines in R and Python

Software / demos

- Intro to RStudio Cloud
- Intro to Google Colab
- We'll look (briefly) at some R and Python notebooks covering TADA material that you would have seen, more or less, in a prior class.
 - Log in to <https://rstudio.cloud> via SSO with your Essex account. The class should be open to you and the project for Day 1 should be accessible as well.
 - Google Colab notebook linked from course information page: <https://burtmonroe.github.io/TextAsDataCourse/Essex/>