INTRODUCTION TO TEXT ANALYTICS

DxU Methods Workshop

Adam Jassem

March 09, 2022

Department of Quantitative Economics

OUTLINE

- Introduction
    - What is text mining/text analytics?
- Natural Language Processing
    - From text to numbers
- Text analysis
    - From numbers to insight

WHAT IS TEXT MINING?

Text mining is an umbrella term for a variety of techniques

- combining methods from:
  - linguistics,
  - statistics,
  - machine learning,
  - computer science.
- Common goal of deriving useful information from text data

TEXT MINING, TEXT ANALYTICS, NLP

Is text analytics something else?

- The two work interchangeably
- Usage might depend on application
  - text **mining** provides qualitative answers,
  - text **analytics** provides quantitative answers.

TEXT MINING, TEXT ANALYTICS, NLP

Is text analytics something else?

- The two work interchangeably
- Usage might depend on application
  - text **mining** provides qualitative answers,
  - text **analytics** provides quantitative answers.

Then what is Natural Language Processing?

- Unstructured text is just a long string of characters?
- We need to express it in a way that allows analysis.
- This is the goal of NLP
  - Process the texts
  - to capture the meaning/content
  - in a form that allows further analysis

The **choice of methods** used in text mining depend on the **questions that need to be answered**

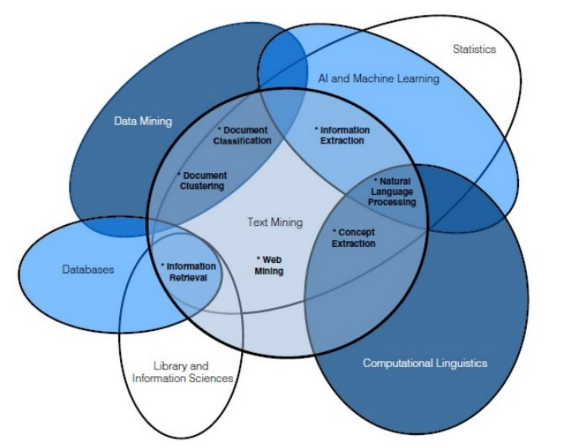


Figure: Miner et al. (2012), Practical text mining and statistical analysis for non-structured text data applications.

TYPES OF APPLICATIONS - INFORMATION RETRIEVAL/EXTRACTION

Information retrieval

- "I've got a question, **where** is the answer?"
- The granddaddy of text mining: library science (1940's-...)
- Nowadays: search engines

Information extraction

- "I've got a question, **what** is the answer?"
- First "real" text mining - military application (1980's-...)
- Nowadays: knowledge bases (Google, Siri etc.)
- Some economic research applications:
    - Analysis of patent applications, tax statements

TYPES OF APPLICATIONS - INFORMATION RETRIEVAL/EXTRACTION

Information retrieval

- "I've got a question, **where** is the answer?"
- The granddaddy of text mining: library science (1940's-...)
- Nowadays: search engines

Information extraction

- "I've got a question, **what** is the answer?"
- First "real" text mining - military application (1980's-...)
- Nowadays: knowledge bases (Google, Siri etc.)
- Some economic research applications:
  - Analysis of patent applications, tax statements

Generally, building a whole, domain specific system such that

- Ask questions $\rightarrow$ get answers (or at least the location).
- Usually, and **answer is based on one text** (or just a few).

TYPES OF APPLICATIONS - SUMMARIZING TEXTS

Many applications in economic research:

- **Single question** known upfront (or just a few)
- **Many texts** that are informative
- We want to consider all/most of them
- We therefore want to summarize the information
- Usually in a quantitative manner.

Many applications in economic research:

- **Single question** known upfront (or just a few)
- **Many texts** that are informative
- We want to consider all/most of them
- We therefore want to summarize the information
- Usually in a quantitative manner.

Document classification and clustering:

- Generally, rather simple questions and answers
  - "Are those text optimistic or pessimistic?"
  - "What issues do those texts mention?"
- Either **predefined** (classification) or **learned** (clustering) labels
- Hard or soft (degree/score instead of label)
- Per-document, not 100% accurate, but useful when aggregated.

EXAMPLES OF APPLICATIONS IN ECONOMIC RESEARCH

- Economic Policy Uncertainty index (Baker et al., 2016)
  - Analyzes news coverage, identifies articles about economic policy uncertainty
  - measure their frequency, create a time series

- Economic Policy Uncertainty index (Baker et al., 2016)
  - Analyzes news coverage, identifies articles about economic policy uncertainty
  - measure their frequency, create a time series
- How much does the news drive the economy (Larsen and Thorsrud, 2019)
  - Identifies the news topics that have impact on the stock market and macroeconomic aggregates.

- Economic Policy Uncertainty index (Baker et al., 2016)
  - Analyzes news coverage, identifies articles about economic policy uncertainty
  - measure their frequency, create a time series
- How much does the news drive the economy (Larsen and Thorsrud, 2019)
  - Identifies the news topics that have impact on the stock market and macroeconomic aggregates.
- The economic impact of statements by Fed (Hansen and McMahon, 2016) and other policymakers
  - Identifying news about tax changes from U.S. presidential speeches (Jassem et al., 2021)

NATURAL LANGUAGE PROCESSING

WHY DO WE NEED NATURAL LANGUAGE PROCESSING?

Main goal: Make the computer "understand" the texts

- We want to **capture the meaning** of a text
- in a way that a computer/algorithm can use.

WHY DO WE NEED NATURAL LANGUAGE PROCESSING?

Main goal: Make the computer “understand” the texts

- We want to **capture the meaning** of a text
- in a way that a computer/algorithm can use.

Where does the meaning come from?

WHY DO WE NEED NATURAL LANGUAGE PROCESSING?

Main goal: Make the computer "understand" the texts

- We want to **capture the meaning** of a text
- in a way that a computer/algorithm can use.

Where does the meaning come from?

- Semantics - the meaning of words
  - Challenge: synonymy, polysemy (multiple meanings)

WHY DO WE NEED NATURAL LANGUAGE PROCESSING?

Main goal: Make the computer "understand" the texts

- We want to **capture the meaning** of a text
- in a way that a computer/algorithm can use.

Where does the meaning come from?

- Syntax - the meaning of grammatical structures
  - Challenge: syntax can be ambiguous

Main goal: Make the computer "understand" the texts

- We want to **capture the meaning** of a text
- in a way that a computer/algorithm can use.

Where does the meaning come from?

- Pragmatics - the meaning derived from context
  - Challenge: how to gather and incorporate context?

WHY DO WE NEED NATURAL LANGUAGE PROCESSING?

Main goal: Make the computer “understand” the texts

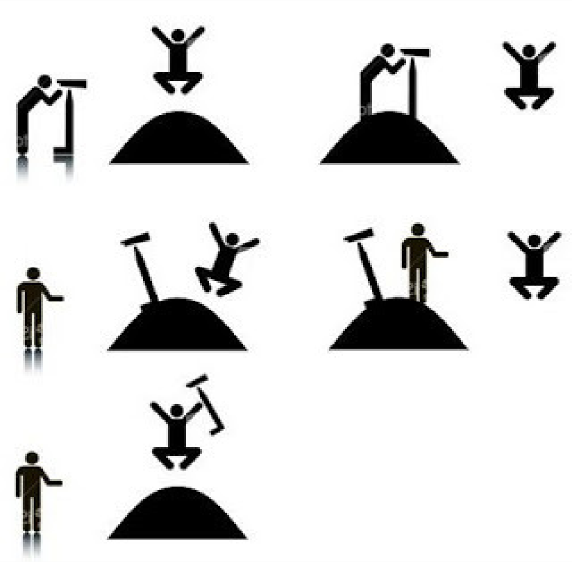
- We want to **capture the meaning** of a text
- in a way that a computer/algorithm can use.

Where does the meaning come from?

- Semantics - the meaning of words
    - Challenge: synonymy, polysemy (multiple meanings)
- Syntax - the meaning of grammatical structures
    - Challenge: syntax can be ambiguous
- Pragmatics - the meaning derived from context
    - Challenge: how to gather and incorporate context?

“I saw a man on a hill with a telescope.”

I SAW A MAN ON A HILL WITH A TELESCOPE



Feature selection

- What information am I interested in? (application dependent)
- What features of the text carry that information?
- What amount of detail do I need to consider?
    - What words are used in the text?
    - Do I need to consider the syntax?
    - Or punctuation (e.g. ?!@#) ?
    - Or the context (e.g. named entities)?
- Trade-off: simplicity vs complexity

We want to process the texts to express them in terms of the relevant features.

For analysis, we usually want to quantify the features.

- Often as simple as counting how many times each feature appears in a text

Basic approach: Bag-of-words approach

- If the meaning is carried by the words used
- just count how many times each word was used in each text.

1. We have D documents $\boldsymbol{w}_d$

   $\boldsymbol{w_1} =$ "Thank you. Wow. Well, you know..."

   $\boldsymbol{w_2} =$ "We meet here at a moment of unlimited potential..."

   $\vdots$

1. We have D documents $\boldsymbol{w}_d$
2. We break them up into individual words $\boldsymbol{w}_d = (w_{d,1}, \ldots, w_{d,N_d})$ (tokenization), identify the vocabulary $\mathcal{V} = \{v_1, \ldots, v_V\}$
   $\mathcal{V} = \{\text{`aardvark'}, \text{`abbreviate'}, \ldots, \text{`zymotic'}\}$
   $\boldsymbol{w_1} = (\text{`thank'}, \text{`you'}, \text{`wow'}, \text{`well'}, \text{`you'}, \text{`know'}, \ldots)$
   $\boldsymbol{w_2} = (\text{`we'}, \text{`meet'}, \text{`here'}, \text{`at'}, \text{`a'}, \text{`moment'}, \text{`of'}, \text{`unlimited'}, \ldots)$
   $\vdots$

1. We have D documents $\boldsymbol{w}_d$
2. We break them up into individual words $\boldsymbol{w}_d = (w_{d,1}, \ldots, w_{d,N_d})$ (tokenization), identify the vocabulary $\mathcal{V} = \{v_1, \ldots, v_V\}$
3. We count how many times each word appears in each document (vectorization),

$$f_{d,i} = \sum_{n=1}^{N_d} \mathbb{1}(w_{d,n} = v_i), \quad \boldsymbol{f}_d = (f_{d,1}, \ldots, f_{d,V}$$

End result - a document-term counts matrix:.

$$F = \begin{bmatrix} f_{1,1} & \cdots & f_{1,V} \\ \vdots & \ddots & \vdots \\ f_{D,1} & \cdots & f_{D,V} \end{bmatrix}$$

IMPROVING ON THE *BAG-OF-WORDS*

Basic ways to improve a bag-of-words approach:

- Cleaning the vocabulary - removing stopwords ('a', 'the', 'and'...), rare words (e.g. typos), common words.
- Stemming/lemmatization - 'dogs' $\rightarrow$ 'dog', 'running' $\rightarrow$ 'run'

Basic ways to improve a bag-of-words approach:

- Cleaning the vocabulary - removing stopwords (‘a’, ‘the’, ‘and’...), rare words (e.g. typos), common words.
- Stemming/lemmatization - ‘dogs’ $\rightarrow$ ‘dog’, ‘running’ $\rightarrow$ ‘run’
- Not all words are equally informative.
  - Term frequency - inverse document frequency (TF-IDF) score
  - Scale down the score of the terms that appear in many texts, e.g.:

$$\mathrm{tfidf}_{d,i} = \mathrm{tf}_{d,i} \times \mathrm{idf}_i$$

$$\mathrm{tf}_{d,i} = \frac{f_{d,i}}{N_d}$$

$$\mathrm{idf}_i = \log\left(\frac{D}{\sum_{d=1}^{D} \mathbb{1}(f_{d,i} > 0) + 1}\right)$$

GOING PAST *BAG-OF-WORDS*

By considering only individual words we lose a lot of information.

- n-grams - Meaningful phrases consisting of more than one word.
  - Identify combinations of consecutive words that appear "abnormally" often, e.g. bigrams:

$$P(w_{d,n} = v_i, w_{d,n+1} = v_j) > P(w_{d,n} = v_i)P(w_{d,n} = v_j)$$

- Part-of-speech tagging - the meaning of the word can depend on whether it's a noun, verb etc.
- Chunking (shallow parsing)
  - Based on the POS tags we can define certain syntax structures we're interested in
  - For example, noun-phrases: $<\text{PREP}>?<\text{ADJ}>*<\text{NOUN}>+$ (potential preposition, any number of adjectives, one or more nouns)

ANALYSIS OF THE TEXTS

Example:

Consumer reviews: predict rating based on features of the text

1. We define the relevant features of the texts
2. We create quantitative representation of the texts

Example:
Consumer reviews: predict rating based on features of the text

1. We define the relevant features of the texts
2. We create quantitative representation of the texts
3. We move onto statistical analysis. A regression model?

$$\texttt{rating}_d = \alpha + \sum_{i=1}^{V} \beta_i f_{d,i} + \varepsilon_d$$

Example:
Consumer reviews: predict rating based on features of the text

1. We define the relevant features of the texts
2. We create quantitative representation of the texts
3. We move onto statistical analysis. A regression model?

$$\texttt{rating}_d = \alpha + \sum_{i=1}^{V} \beta_i f_{d,i} + \varepsilon_d$$

This might be **thousands or even millions of variables.**

- OLS likely is not going to cut it

Some methods are better suited for the high-dimensional setting:

- Penalized regression (LASSO, Ridge etc.)
- Support-Vector Machines, Naive Bayes classifier, …

Those methods don't really tell us much about the underlying meaning of the texts.

- For this, we might want to perform dimensionality reduction
  - Represent the features/texts using smaller number of dimensions
- Semantic space, such that the dimensions that have some meaningful interpretation

General idea:

- Simple neural network
- Predicting word usage using the rest of the text
- Hidden layer has lower dimensionality.
  - It creates a **lower-dimensional representation** of a word

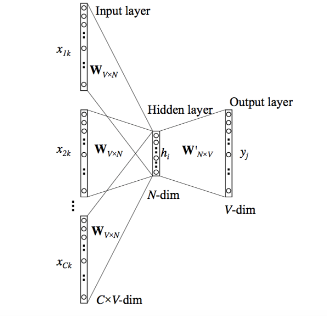


Figure: Rong (2014)

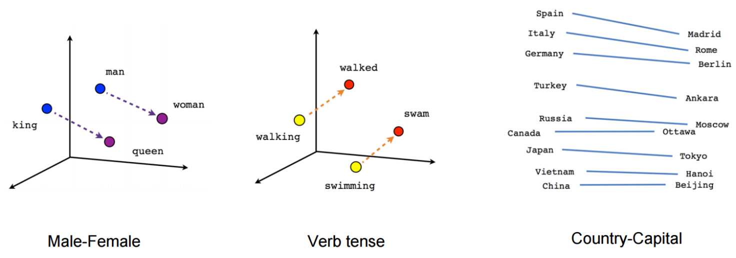


Figure: Visualisation of word2vec

RULES- AND LEXICON-BASED APPROACH

RULES-BASED APPROACH

General idea:
We know a priori that some words convey some particular meaning.
Let's just look for those.

Example: Economic Policy Uncertainty Index (Baker et al., 2016)
For each news article check if it contains at least one word that's:

1. related to economy - {`'economy'`, `'economic'`, ...}
2. related to policy - {`'policy'`, `'congress'`, ...}
3. related to uncertainty - {`'uncertain'`, `'uncertainty'`, ...}

It does? Then it is about economic policy uncertainty.

Count how many of those in a quarter - there's your EPU index.
(almost 7000 citations)

For each meaning of interest we can make a list of related words, so called lexicon.
Then we might ask - how much of a given document is within each lexicon?

- D documents $\boldsymbol{w}_d = (w_{d,1}, \ldots, w_{d,N_d}), \quad w_{d,n} \in \mathcal{V}$.
- K categories, each with a lexicon $L_k \subset \mathcal{V}, \quad k = 1, \ldots, K$

For each meaning of interest we can make a list of related words, so called lexicon.
Then we might ask - how much of a given document is within each lexicon?

- D documents $\boldsymbol{w}_d = (w_{d,1}, \dots, w_{d,N_d}), \quad w_{d,n} \in \mathcal{V}$.
- K categories, each with a lexicon $L_k \subset \mathcal{V}, \quad k = 1, \dots, K$

Then we can simply express the documents as $\boldsymbol{p}_d = (p_{d,1}, \dots, p_{d,K})$ where:

$$p_{d,k} = \frac{\sum_{n=1}^{N_d} \mathbb{1}(w_{d,n} \in L_k)}{N_d}$$

(or any alternative aggregation scheme)

Lexicon based approach is commonly used in sentiment analysis, answering questions such as:

- Is the consumer review positive or negative?
- Is the news article optimistic or pessimistic?
- What emotions do the policymakers convey?

Based on pre-defined lexicons for each sentiment

- Those can be “all-purpose” or domain-specific

Common extension: valence/intensity $v(\text{term})$

- How strong is the sentiment of the word, e.g.:

$$0 < v(\text{`okay'}) < v(\text{`good'}) < v(\text{`great'}) < v(\text{`perfect'}) \leq 1$$

TOPIC MODELLING

FROM SUPERVISED TO UNSUPERVISED METHODS

Let's say instead of sentiments, we want to identify news topics.

- How would we specify a lexicon for every possible topic?
- With so many words being used in different contexts, would we try to define rules for every topic?

Let's say instead of sentiments, we want to identify news topics.

- How would we specify a lexicon for every possible topic?
- With so many words being used in different contexts, would we try to define rules for every topic?
- The results would depend heavily on our specifications

FROM SUPERVISED TO UNSUPERVISED METHODS

Let's say instead of sentiments, we want to identify news topics.

- How would we specify a lexicon for every possible topic?
- With so many words being used in different contexts, would we try to define rules for every topic?
- The results would depend heavily on our specifications

However, we see that certain terms tend to co-occur in the documents.

- For each topic, certain words are used more often
- Can we learn from the data which words are indicative of which topics?

We observe the data: $D \times V$ document-term counts matrix.

We want to explain is using $K \ll V$ topics:

- a $D \times K$ document-term matrix - what are the documents about
- and $K \times V$ topic-term matrix - what terms are the topics using,

To achieve this, we are going to perform clustering. We are effectively clustering:

- terms based on their co-occurance in documents,
- at the same time, documents based on what terms they use.

We are expressing both the terms and the documents in a lower-dimensional (semantic) space.

It is quite intuitive to consider the probability of a term being used.

Let's take a step back: consider the simplest probabilistic model - unigram model.

It is quite intuitive to consider the probability of a term being used.

Let's take a step back: consider the simplest probabilistic model - unigram model.

Each token $w_{d,n}$ is assumed to be a draw from a categorical distribution over the vocabulary, given by a probability vector $\boldsymbol{\phi} = (\phi_1, \ldots, \phi_V)$:

$$w_{d,n} \sim \mathsf{Cat}(\boldsymbol{\phi})$$

$$\mathbb{P}(w_{d,n} = v_i) = \phi_i$$

It is quite intuitive to consider the probability of a term being used.

Let's take a step back: consider the simplest probabilistic model - unigram model.

Each token $w_{d,n}$ is assumed to be a draw from a categorical distribution over the vocabulary, given by a probability vector $\boldsymbol{\phi} = (\phi_1, \ldots, \phi_V)$:

$$w_{d,n} \sim \mathrm{Cat}(\boldsymbol{\phi})$$

$$\mathbb{P}(w_{d,n} = v_i) = \phi_i$$

Those probabilities doesn't depend on anything, it's the same for all the words in all the documents. Not very informative.

Instead, assume there are K such distributions, one for each topic

$$\boldsymbol{\phi}_k = (\phi_{k,1}, \ldots, \phi_{k,V}), \quad k = 1, \ldots, K$$

- This is the basis for the LDA model proposed by Blei et al. (2003)

The key assumption of LDA is that each token $w_{d,n}$ has a latent topic assignment $z_{d,n}$

$$w_{d,n} | z_{d,n} \sim \mathsf{Cat}(\boldsymbol{\phi}_{z_{d,n}})$$

$$\mathbb{P}(w_{d,n} = v_i | z_{d,n}) = \phi_{z_{d,n},i}$$

Which terms are used depends on what the token is about.

For each document we can consider:
What is the proportion of tokens that are about a particular topic?

We can express this with a probability vector:

$$\boldsymbol{\theta}_d = (\theta_{d,1}, \ldots, \theta_{d,K}), \quad d = 1, \ldots, D$$

such that:

$$z_{d,n} \sim \mathsf{Cat}(\boldsymbol{\theta}_d)$$

$$\mathbb{P}(z_{d,n} = k) = \theta_{d,k}$$

LATENT DIRICHLET ALLOCATION MODEL, CONT.

Using Bayesian methods we can find $\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_K$ and $\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_D$ that best explain the actual data we see.

- $\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_K$ tell us how each of the topics is discussed
    - We don't really know which distribution is about what
    - We need to assign interpretations to them
- $\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_D$ tell us what each of the documents is about
    - We've effectively summarized each document
    - We can now use this lower-dimensional representation in subsequent analysis

LATENT DIRICHLET ALLOCATION MODEL, CONT.

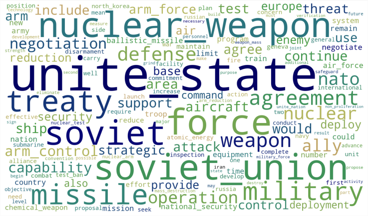
Using Bayesian methods we can find $\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_K$ and $\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_D$ that best explain the actual data we see.

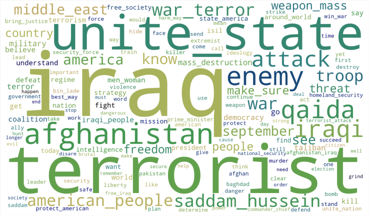
- $\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_K$ tell us how each of the topics is discussed
  - We don't really know which distribution is about what
  - We need to assign interpretations to them
- $\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_D$ tell us what each of the documents is about
  - We've effectively summarized each document
  - We can now use this lower-dimensional representation in subsequent analysis

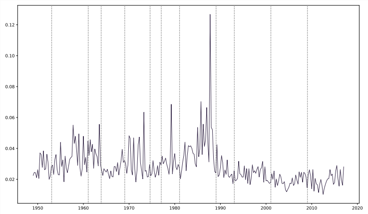
Let's consider the example of U.S. presidential speeches.

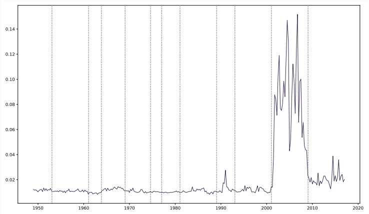
- We might even want to aggregate the per-document proportion into per-quarter measures.

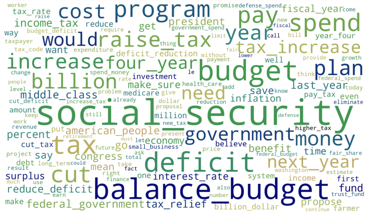
U.S. PRESIDENTIAL SPEECHES



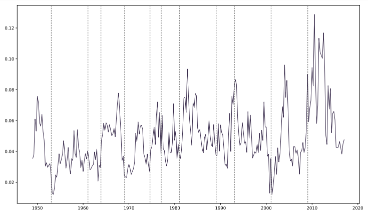


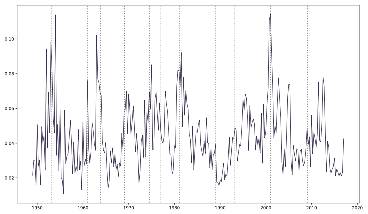





social_security
balance_budget
deficit
budget
program
cost
pay
spend
raise_tax
tax_increase
increase
four_year
billion
plan
government
money
tax
cut
need
next_year
would
year
middle_class
income_tax
federal_government
tax_relief
interest_rate
reduce_deficit
american_people
congress
benefit
inflation
last_year
make_sure
president
fiscal_year
surplus
percent
economy
fund
first
propose


small_business
recession
new_job
economy
last_year
interest_rate
unemployment
unemployment_rate
percent
business
create_job
tax_cut
economic_growth
million_new
tax
growth
job
investment
inflation
tax_relief
tax_rate
company
america
worker
cut_tax
recovery
incentive
middle_class
year_ago
million_job
today
cut
last_month
tax_credit
private_sector
tax_break


0.12
0.10
0.08
0.06
0.04
0.02
1950
1960
1970
1980
1990
2000
2010
2020


0.10
0.08
0.06
0.04
0.02
1950
1960
1970
1980
1990
2000
2010
2020

QUESTIONS?

PLAN FOR THE SECOND SESSION

Hand-on experience using `python` (jupyter)

- `colab` - an online service for running `python` code
  - https://colab.research.google.com/
  - No need to install python/packages
- Notebook from GitHub > amjassem > DxU > **Book Reviews**
  - Basics of NLP
  - Sentiment analysis
  - Basic statistical models
  - Topic modelling
- ... > **Trump Tweets**

Baker, S. R., N. Bloom, and S. J. Davis (2016). Measuring economic policy uncertainty. The quarterly journal of economics 131(4), 1593–1636.

Blei, D. M., A. Y. Ng, and M. I. Jordan (2003). Latent dirichlet allocation. the Journal of machine Learning research 3, 993–1022.

Hansen, S. and M. McMahon (2016). Shocking language: Understanding the macroeconomic effects of central bank communication. Journal of International Economics 99, S114–S133.

Jassem, A., L. Lieb, R. J. Almeida, N. Baştürk, and S. Smeekes (2021). Min (d) ing the president: A text analytic approach to measuring tax news. arXiv preprint arXiv:2104.03261.

Larsen, V. H. and L. A. Thorsrud (2019). The value of news for economic developments. Journal of Econometrics 210(1), 203–218.

Miner, G., J. Elder IV, A. Fast, T. Hill, R. Nisbet, and D. Delen (2012). Practical text mining and statistical analysis for non-structured text data applications. Academic Press.

Rong, X. (2014). word2vec parameter learning explained. arXiv preprint arXiv:1411.2738.