SOSC 4300/5500: Text Analysis Basics

Han Zhang

Sep 29, 2020

Outline

Assignment 1

- Why we need this?
  - After the assignment, you will be more familiar about using statistical models to make predictions.
  - With rectangular data (e.g., survey data)
  - With these experiences, we can move on to machine learning/prediction on other complex data types (e.g., text)
- There is no unique solution; you can solve the problem in tons of different ways
  - Through trial and error
  - This is common in both research and work.

Logistics

- From this lecture, each lecture will contain a guided coding component
  - I will show how to implement what we taught in slides
    - Mostly in R, but I will use Python when appropriate
  - Please download and try these codes during/after class.
  - You can only learn by doing
- In assignment and tutorial, we will go over these again (at a greater depth)

Text as data

- Policy documents by governments
- Newspaper articles
- Social media text
- Patent's content
- Scientific articles
- Historical archive
- Else?

A complete workflow (Grimmer and Stewart, 2013)

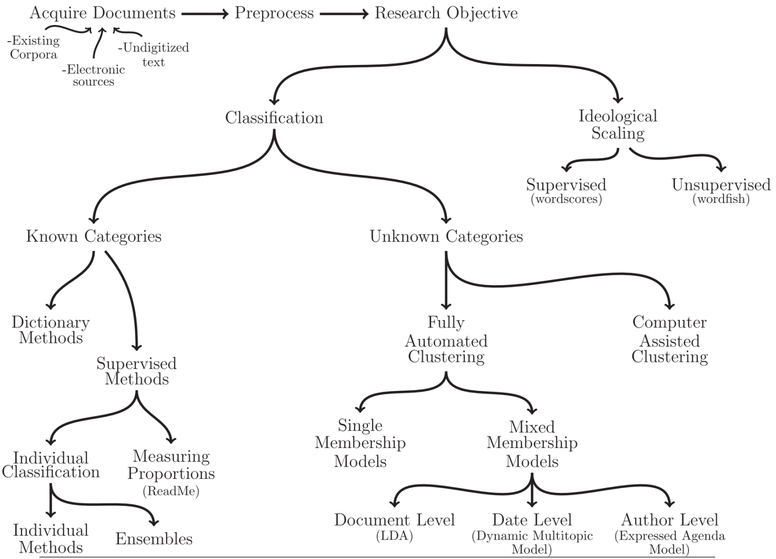

Fig. 1 An overview of text as data methods.

Corpora

- Corpora: a collection of text document
  - A list of text files
  - Or a big CSV/TXT file, with a text column
- Word, term, token (often used interchangeably)
- Vocabulary is all unique words in your corpus

Acquiring texts: Existing Corpora

- Existing Corpora
  - Someone already cleaned the data for you and
  - Easiest to begin
  - But do not always contain what you need
- Some famous collection of datasets:
  - Kaggle Dataset: `https://www.kaggle.com/datasets?tags=14104-text+data`
  - UCI's machine learning repository: `https://archive.ics.uci.edu/ml/datasets.php`
  - US patents: `https://www.google.com/googlebooks/uspto-patents-assignments.html`
  - Wikipedia texts: `https://en.wikipedia.org/wiki/Wikipedia:Database_download`

A cautionary note

- Be cautious if you are using EXCEL to view your files
- Two problems
- If data is huge, it takes forever to open that file
- And excel will insert some characters/do some auto correction for you
- Ziemann, M., Eren, Y. & El-Osta, A. Gene name errors are widespread in the scientific literature. *Genome Biology* 17, 177 (2016). `https://doi.org/10.1186/s13059-016-1044-7`
  - Gene symbols are automatically transformed into something else when opening CSV files with Excel
    - `SEPT2` to date
    - `2310009E13` to numbers (2.31E+13)
  - And it's hard to undo these auto transformation
  - 19.6% of articles in top journals in genetics were impacted

How should you look at the data

- Generate a copy that is not touched by your EXCEL
- Produce a small sample and look at it using excel or something
  - use `head` or `tail` to peek first and last rows
  - `head -n 100 file.csv > head_file.csv` will output first 100 rows of `file.csv` to `head_file.csv`
- Or use a professional text editor to open the file
  - Sublime
  - Visual Studio Code

Acquiring texts: electronic sources

- Electronic sources:
  - Electronic searchable newspaper databases
    - e.g., Factiva
  - Social media
  - Websites

Acquiring data: electronic sources

- Collecting data from electronic sources
  - Manual approach: download and save each page,
  - Automatic approach: web scraping/crawling
    - In tutorial, we taught you how to crawl a basic website
    - Next tutorial: clean HTML files to produce a corpora
  - Now, we assume that we have a cleaned corpora to work with

Acquiring data: undigitized text

- E.g., PDF of scanned books
- Need a lot more work
- Often some OCR (Optical character recognition) is required
  - OCR: Given an image containing texts, predict texts in it.

Stemming and Lemmatization

- Stemming: words with suffixes removed (using set of rules)
  - E.g., "family, families, families, familial" → `famili`
  - Stemming may be problematic, because the not all base form can be obtained by removing suffixes
- Lemmatization: a more complex version that "seeks to reduce words to their base forms".

word	win	winning	wins	won	winner
stem	win	win	win	won	winner
lemma	win	win	win	win	win

- Stop words: common words that may not be relevant to your task.
  - `https://www.aclweb.org/anthology/W18-2502.pdf`

Word segmentation

- For digitized Latin-language families, words have boundary
- But for Chinese, there is no word boundary
- So word segmentation has to be used
  - jieba: easy and quick; precision is relatively low
  - pkuseg and THULAC : better precision; no R version

From Words to Numbers

- Still, there is no easy way for us to use text as variables
- The next step is to turn a corpora into a matrix X with numeric values
    - Or, turn each document into a numeric vector
- Then we can feed this matrix representation of a corpora into a prediction model
    - regression, tree, forest, SVM, etc.,)

Document-Term Matrix

- When Grimmer and Stewart wrote the article in 2013
- Turning corpus into a matrix is usually achieved by obtaining document-term matrix, which rely on word frequencies
- $W : < N \times M >$ matrix; N is the number of documents and M is the size of vocabulary
- W_{im}: the number of times the m-th word occurs in the i-th document.
- The matrix W then can be used as the variables in prediction models
  - E.g., `lm(y ~ W)`

docs	words made	because	had	into	get	some	through	next	where	many	irish
t06_kenny_fg	12	11	5	4	8	4	3	4	5	7	10
t05_cowen_ff	9	4	8	5	5	5	14	13	4	9	8
t14_ocaolain_sf	3	3	3	4	7	3	7	2	3	5	6
t01_lenihan_ff	12	1	5	4	2	11	9	16	14	6	9
t11_gormley_green	0	0	0	3	0	2	0	3	1	1	2
t04_morgan_sf	11	8	7	15	8	19	6	5	3	6	6
t12_ryan_green	2	2	3	7	0	3	0	1	6	0	0
t10_quinn_lab	1	4	4	2	8	4	1	0	1	2	0
t07_odonnell_fg	5	4	2	1	5	0	1	1	0	3	0
t09_higgins_lab	2	2	5	4	0	1	0	0	2	0	0
t03_burton_lab	4	8	12	10	5	5	4	5	8	15	8
t13_cuffe_green	1	2	0	0	11	0	16	3	0	3	1
t08_gilmore_lab	4	8	7	4	3	6	4	5	1	2	11
t02_bruton_fg	1	10	6	4	4	3	0	6	16	5	3

Document-Term Matrix: bag-of-words assumption

- Document-term matrix makes the bag-of-words assumption
- Word order do not matter, only presence maters
  - For some problems it's reasonable (e.g., whether an article mentions China or not)
  - For many other problems, it's clearly wrong (e.g., sentiment)
- A remedy: n-gram approach
  - Adding concurrent words into vocabulary
- E.g., "I am the instructor"
- With 2-gram
- "I am", "am the", "the instructor" are added into vocabulary

Document-Term Matrix: weighting

- Another common problem: some words appear too often
- Instead of just removing them
- We can add weights to document-term matrix, by penalizing words that appear in too many documents
- This is called inverse-document frequency (idf) score.
  - Low idf score suggest the word is common

$$idf_w = log \frac{\text{number of documnet}}{\text{number of documents in which the term w1 par appears}}$$

- tf-idf matrix: combining document-term matrix (term frequency) and inverse-document frequency matrix together
  - Each cell in W multiplies the corresponding word's idf score

Document-Term matrix: curse of dimensionality

- $W : < N \times M >$ matrix; N is the number of documents and M is the size of vocabulary
- M is ofter larger than N, because:
  - Vocabulary size (on a scale of 10K to 100K) is often larger than the number of documents
  - The size of vocabulary can increase exponentially, if n-gram is used
- Therefore, by its design, document-term matrix suffers from the curse of dimensionality
  - Again, simple linear regression does not work well on high-dimensional data, with more columns than rows
- In two weeks we will introduce something called "word embedding"
  - It represents documents into low-dimensional matrix

A complete workflow (Grimmer and Stewart, 2013)

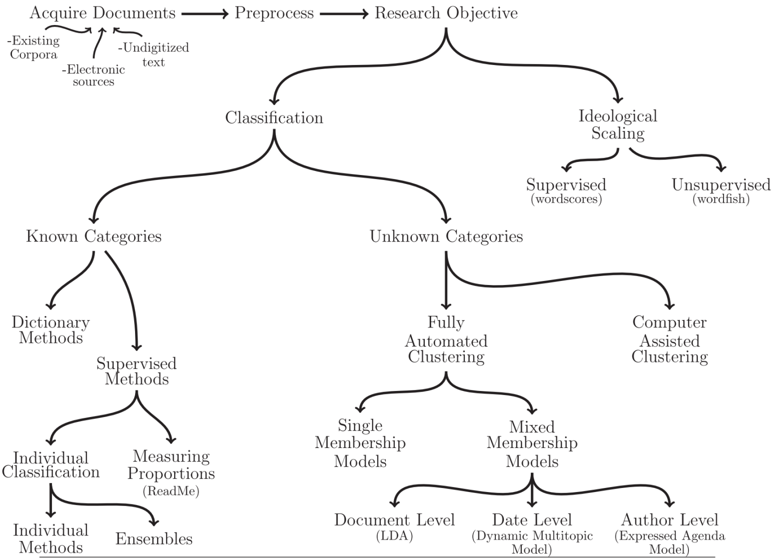

Fig. 1 An overview of text as data methods.

Research objectives

- Supervised: known categories/outcomes
  - Example: sentiment analysis; each document is mapped to either of the three category:
    - positive
    - negative
    - neutral
  - Dictionary methods: deterministic
  - Supervised machine learning: probabilistic
    - linear/logistic regression
    - decision tree/random forests
    - SVM
    - Neural networks and deep learning (the state of art)
    - And many others
- Unsupervised: unknown categories/outcomes
  - The goal is to find patterns in text data

Dictionary method

- The simplest supervised method
  - Often the first step before you jump to some more complex methods
- Dictionary methods relies on curating a list of words
  - Each word is attached with one category
  - Documents with more words in a category is treated as belonging to that category

Dictionary method: one dictionary

- We have collected a bunch of newspaper articles worldwide
- E.g., our research question: whether more foreign news media are reporting more about China after the "Belt and Road Initiative"
- Dictionary: [China, Chine, ...]
- Outcome of each document can be:
  - or, whether a document mentions at least one word in the dictionary (0/1)
  - the number of times a document mentions at least one word in the dictionary (continuous numbers)
  - or, the proportion that a document contains China-related words (to control for document length)
- We have a mapping of document $\rightarrow$ to outcome

Dictionary method: two dictionaries

- Sentiment analysis
- Research question: whether the news report is positive or negative toward China?
- Two dictionaries
  - One for words with positive sentiments;
  - The other for words with negative sentiments;
- A binary measure of sentiment for each document:
  - Positive, if there are more positive words than negative words
  - Negative, vice versa
- A continuous measure of sentiment for each document is:

$$\frac{\text{(number of positive words in that document) - (number of negative words in that document)}}{\text{number of total words in that document}}$$

Or write it down mathematically (Grimmer and Stewart)

- We have a vocabulary of size M
- Document-term matrix: W_{im}, the number of times the m-th word occurs in the i-th document.
- And each word m has a weight s_m, which can take three values:
  - 0 (if it is irrelevant to sentiments)
  - 1 (if it shows positive sentiment)
  - -1 (if it shows negative sentiment)
- Each document i has a length of $N_i = \sum_{m=1}^{M} W_{im}$
- Then sentiment score for a document i can be calculated as:

$$t_i = \frac{1}{N_i} \sum_{m=1}^{M} s_m W_{im}$$

Off-the-shelf dictionaries

- Lots of off-the-shelf dictionaries are available
  - For different tasks
- Some commonly used dictionaries for sentiments
  - Minqing Hu and Bing Liu, *Mining and summarizing customer reviews*, Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (New York, NY, USA), KDD '04, Association for Computing Machinery, 2004, pp. 168–177
    - 6800 words, collected from customer review of products on Amazon: careras, DVD player, MP3 and cellular phone, developed by computer scientists
    - `http://www.cs.uic.edu/~liub/FBS/opinion-lexicon-English.rar`
  - LIWC is more complex collection (not free)
    - Developed by psychologists
    - `https://liwc.wpengine.com/`

Off-the-shelf dictionaries

- Another example: detecting political events from newspapers with dictionaries
- GDELT (https://www.gdeltproject.org/data.html#intro)
  - categories include
    - Making public statement
    - Appealing for help
    - Calling for cooperation
    - Threatening
    - Protesting
    - Military fight
    - And many many more
- Each category has its own dictionary
- If an newspaper article contains more words in a corresponding categories, it is assigned to that category

Construct your own dictionary

- Sometimes off-the-shelf dictionary are not satisfactory
  - Words that are meaningful for restaurant reviews may not be working for your problem
- Construct by yourself!
  - Read your documents closely
  - And pick it up by yourself

Some modern approaches of constructing dictionary

- William L. Hamilton, Kevin Clark, Jure Leskovec, and Dan Jurafsky, *Inducing Domain-Specific Sentiment Lexicons from Unlabeled Corpora*, Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, 2016, pp. 595–605
- Intuition: human have recall biases when constructing dictionaries
- Snow-ball sampling:
  - Start with a set of seed words
  - Find similar words in the corpus, and decide whether to add them to dictionary
  - Iterate the above process until you reach a satisfactory dictionary

Shortcomings

- polysemy: multiple meanings in word

Sentences	Sentiment word	Part-of-speech	Sentiment polarity
Jane is patient to children.	patient	adjective	
Now there is a patient in the class.	patient	noun	

Shortcomings

- What words to keep?
  - Often arbitrary decisions; even experts do not agree with each other
- Size of dictionary:
  - How large the dictionary should be? Is 200 positive words enough? Or we need to have 2,000 positive words?

Shortcomings

- Precision-recall tradeoff
- Often it's tempting to select words that are general
- This choice leads to high recall, but results in low precision
- On the other hand, select specific words result in high precision but low recall
- For instance, select keywords associated with Boston Marathon bombings in 2013
    - #prayforboston selects relevant results, but most tweets about Boston Bombing may not contain this hashtag
    - "Boston" do not miss too much, but the rate it hits an relevant post is very low
- Gary King, Patrick Lam, and Margaret E. Roberts, *Computer-Assisted Keyword and Document Set Discovery from Unstructured Text*, American Journal of Political Science **61** (2017), no. 4, 971–988