Intro to automated text analysis

6 June 2019

Matt W. Loftis

Goals for today

1. Important things computerized text analysis can and cannot do
2. Develop vocabulary for talking about text analysis
3. How to select appropriate text analysis tools
4. Tools for your own basic analyses

Program

- Key definitions
- Types of text analyses:
  - Simpler:
    - Simple scales (dictionaries/word counts)
    - Supervised classification
  - More advanced:
    - Topic modeling
    - Multidimensional scaling
- Conclusion

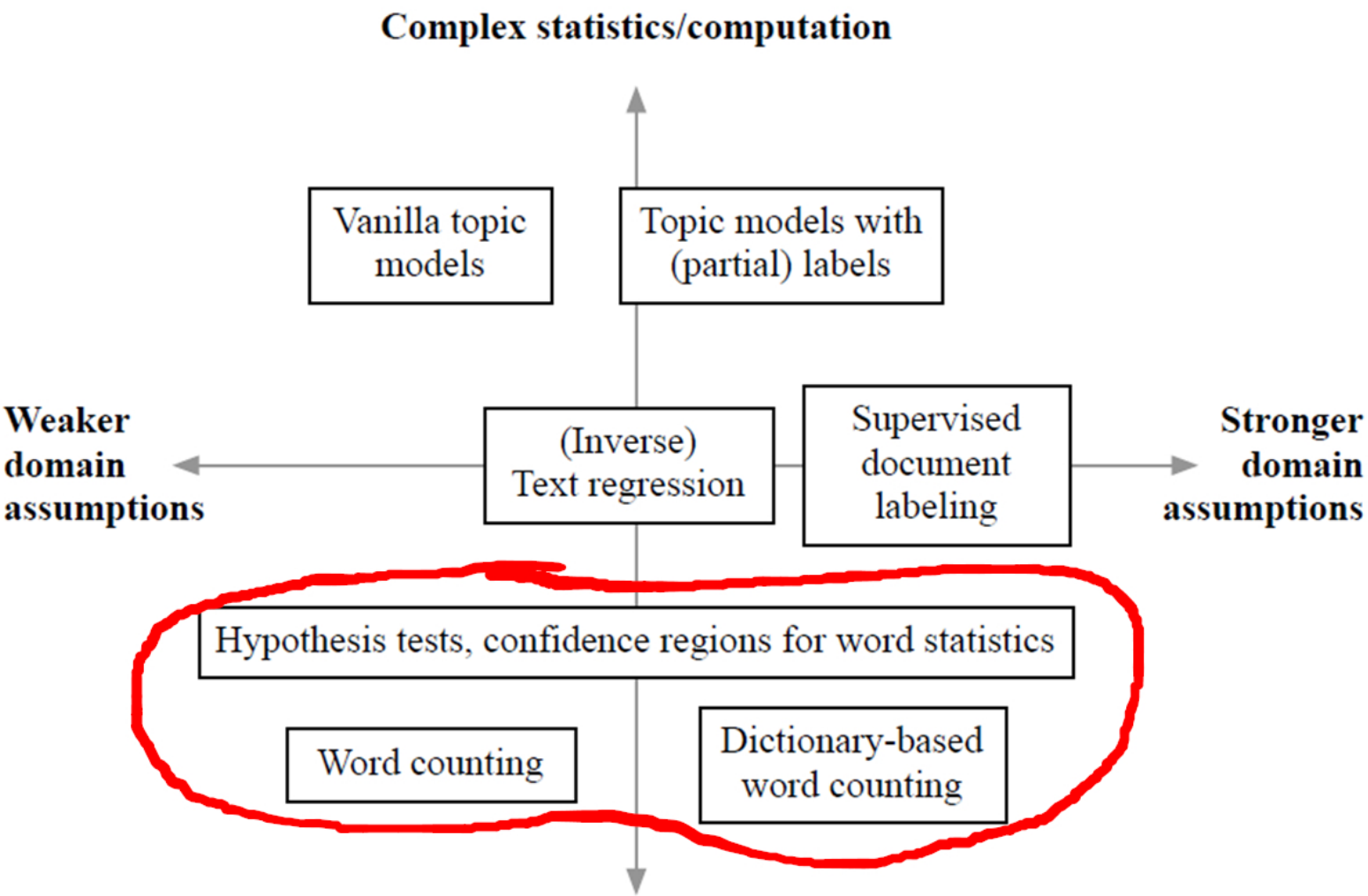
Key definitions

- **Algorithm**: Set of instructions, like a recipe. Also a type of model.
- **Document**: A unit of observation. A speech. A book. A tweet. A law.
- **Corpus**: All of your documents.
- **Feature**: Things appearing in documents. Words, phrases, emojis, etc.
- **Dictionary**: List of words/phrases pertaining to a concept. (e.g. happiness, certainty, etc.)

Basic steps

1. Get some text in electronic format
2. Clean the text
3. Run the analysis
4. Interpret results
5. Make the right graph, table, etc.

Methods (see, O'Conner et al. 2011)



Simple scales

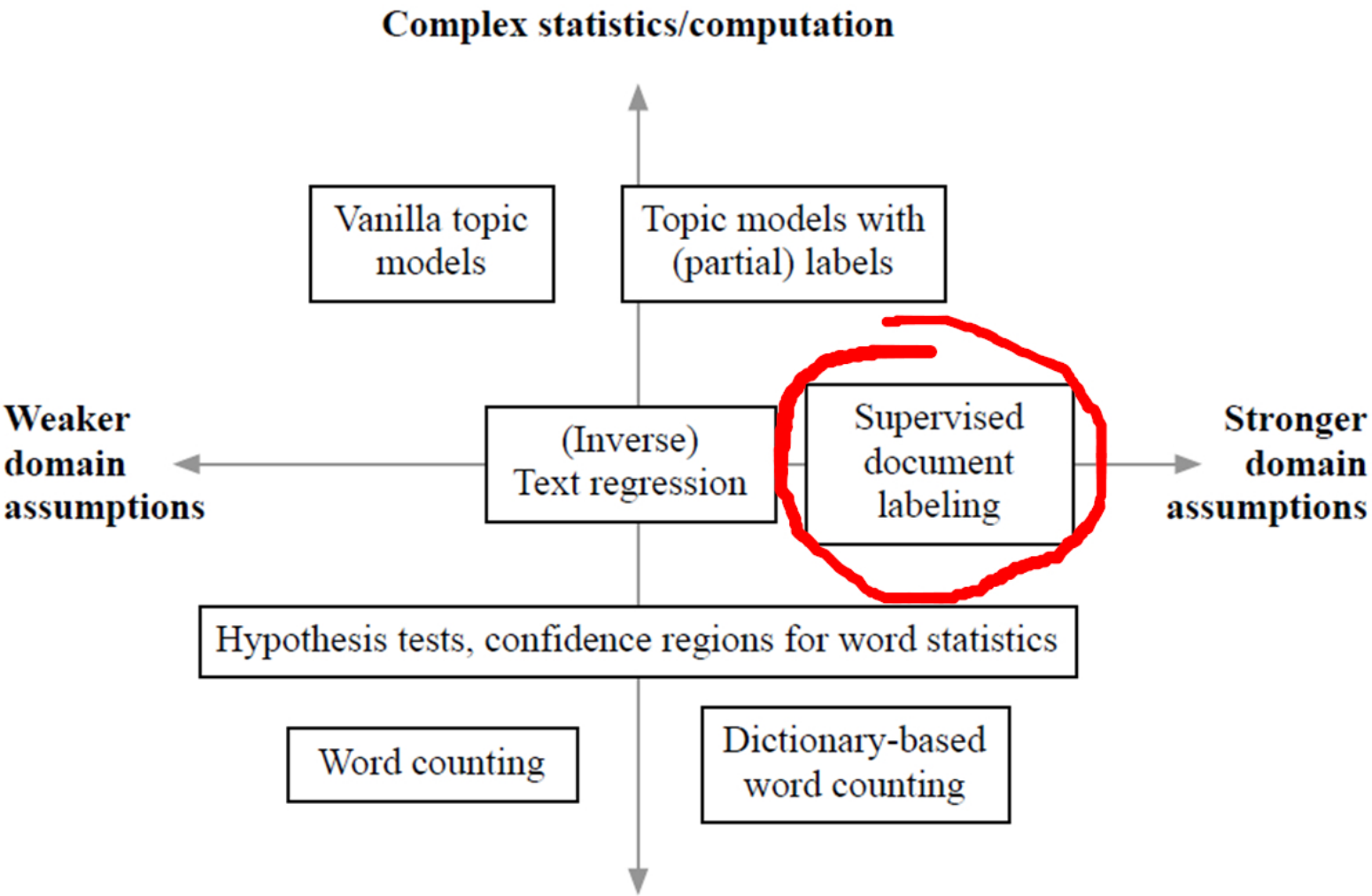
Weaker assumptions:

- Word or phrase counts or relative frequencies
- Example: see here

Stronger assumptions:

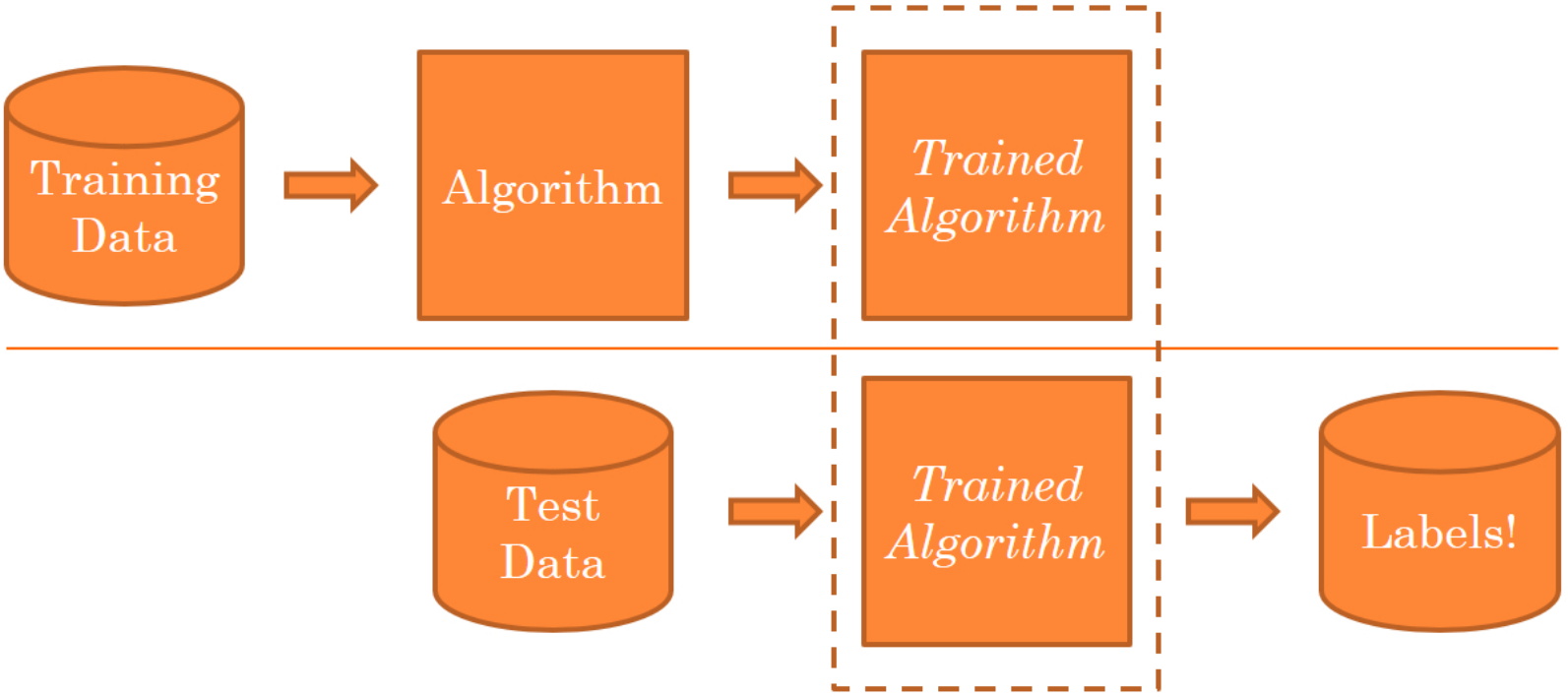
- Counts or relative frequencies of concepts or ideas
- Example: see here
- Example: and here

Methods


Complex statistics/computation
Vanilla topic models
Topic models with (partial) labels
Weaker domain assumptions
(Inverse) Text regression
Supervised document labeling
Stronger domain assumptions
Hypothesis tests, confidence regions for word statistics
Word counting
Dictionary-based word counting

Classification basics

Goal: Use a set of documents with labels to 'teach' the computer to classify new documents.



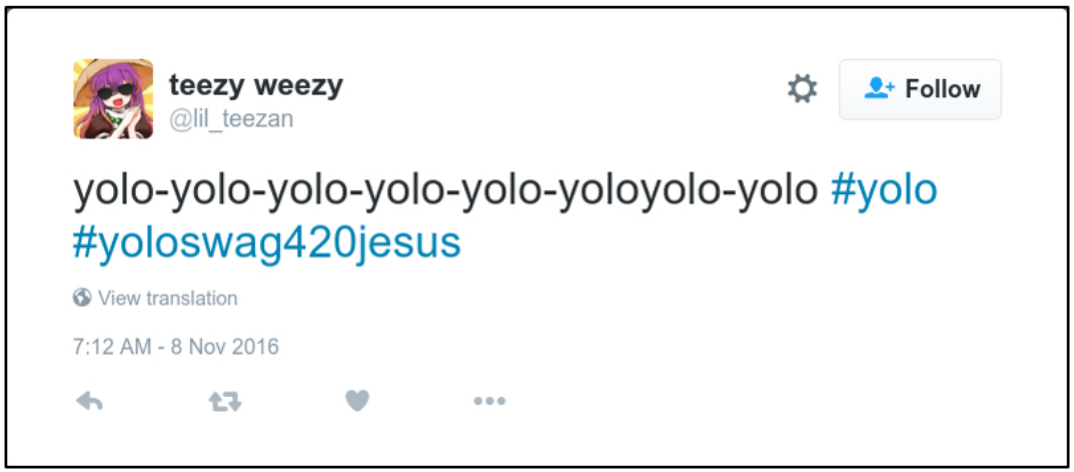
Classification basics

- Training data = documents with labels
- Test data = documents without labels

- Many different methods exist:
  - Support vector machines
  - Naïve Bayes
  - Neural networks
  - Maximum entropy
  - Decision trees
  - Etc.

Just use the one that works best!

Twitter example



#yolo

#subtweet

Classification example

- Training data = #yolo and #subtweet (450 each)
- Test data = 10% subsample
- Method = Naïve Bayes
- Confusion matrix for 10% subsample test

		Predicted		
		#subtweet	#yolo	*Total*
Actual	#subtweet	38	5	*43*
	#yolo	17	30	*47*

Classification example

		Predicted		
		#subtweet	#yolo	*Total*
Actual	#subtweet	38	5	*43*
	#yolo	17	30	*47*

- Precision = % correct out of predictions
  - #subtweet: 0.69
  - #yolo: 0.86
- Recall = % correct out of true
  - #subtweet: 0.88
  - #yolo: 0.64

Interpreting results

- **High precision** means the classifier can tell the difference between classes relatively well
- **High recall** means the classifier can catch most or all instances of a class
- Compare against chance!
- For example, with two classes, the probability of guessing correctly is 50%. So, if overall accuracy is around 50% your model isn't working...

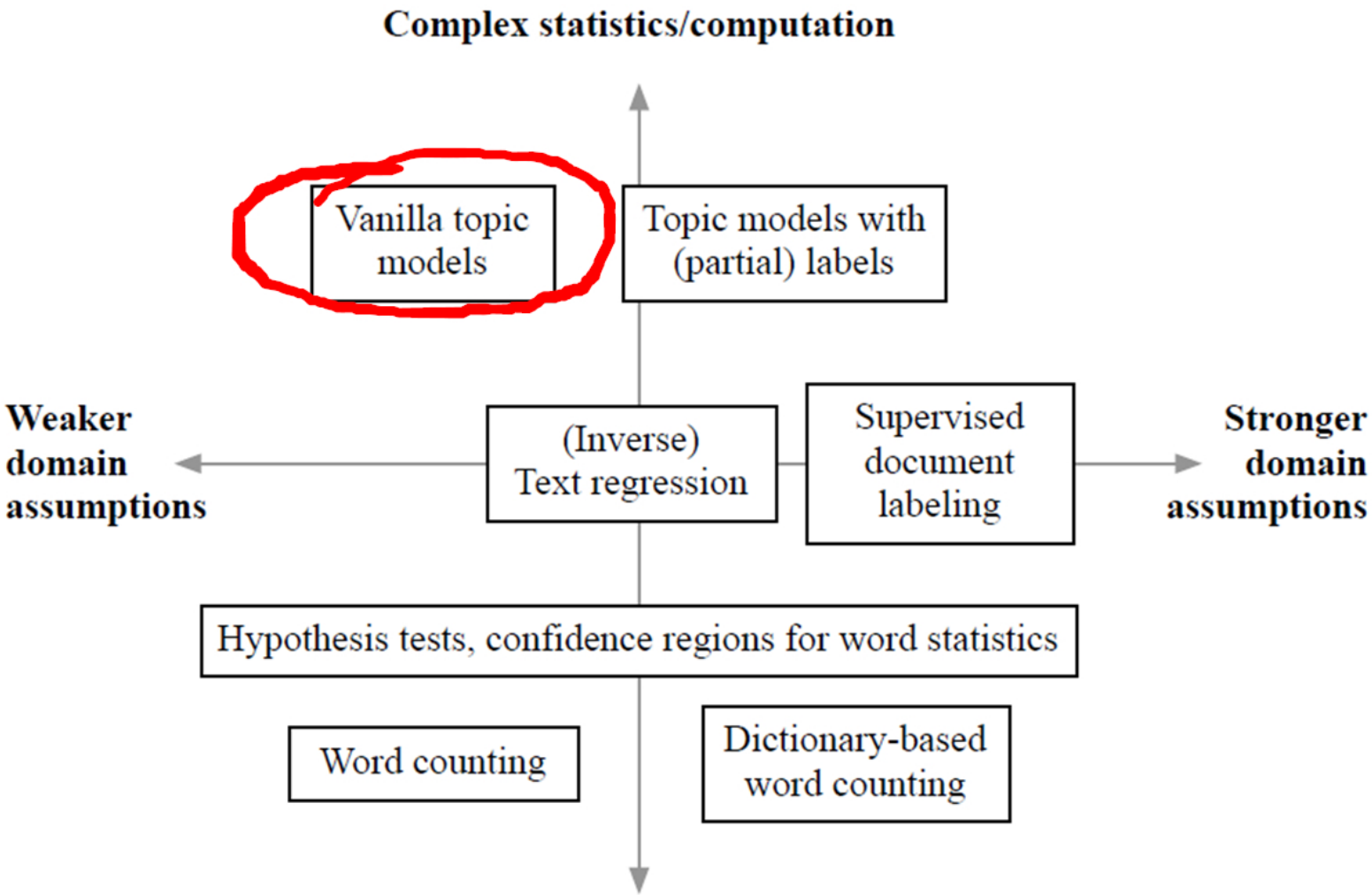
Do-it-yourself classification

- "Easy" software: http://mallet.cs.umass.edu/index.php
- MALLET: (MAchine Learning for LanguagE Toolkit)
- Command-line interface
- Windows, Mac, or Linux
- Fast and relatively very easy
- *Why is this useful???*

Do-it-yourself classification

- See MALLET examples here:
  - https://www.youtube.com/watch?v=zVzUotS9GpQ
  - https://www.youtube.com/watch?v=eBJF5heX5yc

Methods


Complex statistics/computation
Vanilla topic models
Topic models with (partial) labels
Weaker domain assumptions
(Inverse) Text regression
Supervised document labeling
Stronger domain assumptions
Hypothesis tests, confidence regions for word statistics
Word counting
Dictionary-based word counting

Topic modeling

Extracts themes (topics) from documents:

- Give computer documents and number of 'topics'
- Computer returns two things:
  - Features associated with topics
  - Association of documents to topics
- More advanced methods account for other info

Example uses

- See here
- Quick glance at content of lots of documents
- Quantitative measure of document content!

Multidimensional scaling

Place documents on a continuum (scale):

- Computer extracts most important dimension(s) from text
- Returns a number associated with each document (or writer/speaker, etc.)

Example uses

- Place legislators on left-right line from speeches in parliament
- Place interest groups on a for-against dimension based on their public statements on a policy

Quick note on software

- Text processing, counts, and relative frequencies:
  - R, Python, Perl, Ruby, etc.
- Classification, topic models, scaling, etc.
  - R, Python, Ruby, etc.
  - MALLET
- Practical advice:
  - MALLET is easy if you might do this often-ish
  - R or Python are harder, but worth it if this is part of your work
  - Otherwise, get someone trained to do this! But:
    - Know what you want
    - Use the right vocabulary

General points

- Broader fields are:
  - *Natural language processing (NLP)*
  - *Computer science / data science / computational linguistics*
- Automated text analysis is growing fast
- Basically all computer (data) scientists learn these tools
- Used effectively, they make life much easier

Hands-on with MALLET

1. Supervised classification
2. Topic modeling

Supervised classification hands-on

1. Unzip "tweet_data" and "tweet_test" into MALLET folder
2. Build data:
   ```
   bin\mallet import-dir --input tweet_data/*
   --output twitter.mallet
   ```
3. Try out classifier:
   ```
   bin\mallet train-classifier --input
   twitter.mallet --training-portion 0.9
   ```
4. Train classifier:
   ```
   bin\mallet train-classifier --input
   twitter.mallet --output-classifier my.classifier
   ```
5. Classify test data:
   ```
   bin\mallet classify-dir --input tweet_test
   --output results.txt --classifier my.classifier
   ```

Examine output

1. Open a new spreadsheet in Google Sheets
2. Change "spreadsheet settings" country to US
3. Open your file "results.txt" from MALLET folder
4. Copy paste from "results.txt" into spreadsheet
5. Insert title row, label columns
6. Delete useless columns
7. Make new column, "prediction" fill with formula:

```
=IF( [subtweet_prob] > [yolo_prob],
"subtweet", "yolo")
```

Topic modeling

- Group words that co-occur most into *topics*
- Identify how "salient" topics are in documents
  - depends on how many topics you tell it to find
  - depends on how it weights the topics

Topic modelling hands-on

1. Unzip "blm_tweets" into MALLET folder
2. Prep data:
   ```
   bin\mallet import-dir --input blm_tweets --output
   blm.mallet --keep-sequence --remove-stopwords
   ```
3. Sample topic model output:
   ```
   bin\mallet train-topics --input blm.mallet
   ```
4. Topic model with saved output:
   ```
   bin\mallet train-topics  --input blm.mallet
   --num-topics 6 --output-topic-keys blm_keys.txt
   --output-doc-topics blm_composition.txt
   --optimize-interval 10
   ```

Examine output

1. Run topic models MALLET code
2. "Name" topic keys
3. Paste document composition into Google Sheets
4. Make new column, "main topic" fill with formula:
   `=INDEX([topic col 1]$1:[topic col 6]$1, 1, MATCH(MAX([topic 1 doc 1]:[topic 6 doc 1]), [topic 1 doc 1]:[topic 6 doc 1], 0))`
5. Drag duplicate to bottom of column
6. Make a table to the side using "COUNTIF" function