Text as Data

A guided tour

Dr. Travis G. Coan
T.Coan@exeter.ac.uk

Text as data

Available Text
Time
Sumarian tablets
3,500 BC
Papyrus scrolls
2,400 BC
The Guttenburg Bible
1455 AD
Printing press
1490 AD
Digital Revolution

Language is the medium for politics and policy

Language is the medium for politics and political conflict. Candidates debate and state policy positions during a campaign. Once elected, representatives write and debate legislation. After laws are passed, bureaucrats solicit comments before they issue regulations. Nations regularly negotiate and then sign peace treaties, with language that signals the motivations and relative power of the countries involved. News reports document the day-to-day affairs of international relations that provide a detailed picture of conflict and cooperation. Individual candidates and political parties articulate their views through party platforms and manifestos. Terrorist groups even reveal their preferences and goals through recruiting materials, magazines, and public statements.

(Grimmer and Stewart 2013)

The **problem**: Too much text to read!

"Here's a list of 100,000 warehouses full of data. I'd like you to condense them down to one meaningful warehouse."

“Why Text Mining May Be The Next Big Thing”

Time Magazine, March 20, 2012

Text ~~mining~~ as data

Fig. 1 An overview of text as data methods.

Computer-**assisted** content analysis

We emphasize that the complexity of language implies that automated content analysis methods will never replace careful and close reading of texts. Rather, the methods that we profile here are best thought of as **amplifying** *and* **augmenting** *careful reading and thoughtful analysis. Further, automated content methods are* **incorrect** *models of language. This means that the performance of any one method on a new data set cannot be guaranteed, and therefore validation is essential when applying automated content methods.*

(Grimmer and Stewart 2013, emphasis in original)

Four **principles** of text as data

Grimmer and Stewart (2013) outline four general principles that are absolutely essential to keep in mind when using text as data:

Principle 1: All Quantitative Models of Language Are Wrong—But Some Are Useful.

Principle 2: Quantitative methods for text amplify resources and augment humans.

Principle 3: There is no globally best method for automated text analysis.

Principle 4: Validate, Validate, Validate.

Disclaimer: programming required!

Off-the-shelf software: There are a number of "off-the-self" software solutions for text analysis (e.g., WordStat). However, this software is expensive and lacks flexibility.

Open-source solutions: There are a number of excellent open-source alternatives for text mining.

1. **R**: There are a number of different options to choose from, but I highly recommend taking a look at quanteda.
2. **Python**: My personal preference!

Why Python?

Why Python?

1. Any language named after the Monty Python is worth giving a try!
2. It's easy
3. It's fast enough (and pretty easy to speed up)
4. Large user community
5. It is becoming the industry standard for scientific computing.

Why Python?

Most In-Demand Languages
Indeed Job Openings - Dec. 2017
Java 26,269
JavaScript 24,248
C# 13,523
Python 11,757
C++ 8,584
C 8,111
PHP 4,971
Ruby 4,417
Go 4,038
Perl 3,243
PL/SQL 2,962
Scala 1,827
Objective-C 1,730
Apex 1,609
R 1,599
Swift 1,510
SAS 975
MATLAB 789
Crystal 789
Scratch 700

Why Python?

Python, R, Both, or Other platforms for
Analytics, Data Science, Machine Learning
50%
40%
30%
20%
10%
0%
34%
8.5%
42%
16%
41%
12%
36%
11%
Share in 2016
Share in 2017
Python
Both
R
Other
KDnuggets™
Poll 2017

Learning Python

There are hundreds (maybe thousands) of resources to learn Python on the web. Here are some of my favorites:

- **Online courses**: Codeacademy's Learn Python course (https://www.codecademy.com/learn/learn-python)
- **Books**: *A Byte of Python* (https://python.swaroopch.com/)
- **Q-Step Workshop**: Iulia runs an introductory workshop on Python (http://vle.exeter.ac.uk/mod/folder/view.php?id=692789)

We recommend installing the Anaconda distribution of Python (https://conda.io/docs/user-guide/install/download.html). Detailed instructions on installation are available at the following: XXX.

Popular text analysis libraries in Python

There are a large number of Python libraries (or "modules") that are useful for text analysis. Several of the most popular include:

- The Natural Language Toolkit (**NLTK**) (https://www.nltk.org/)
- **scikit-learn** (http://scikit-learn.org/stable/)
- **gensim** (https://radimrehurek.com/gensim/)
- **spaCy** (https://spacy.io/)

These libraries are a "must have" for doing text analysis in Python!

I. Prepocessing

"Standard" **preprocessing** decisions

There are a number of "standard" preprocessing steps that researchers carryout prior to a text as data project. The main steps include:

- **Tokenization**: Splitting documents into smaller units of analysis (typically words).
- **Stopword removal**: Removing frequently occurring words that add little information to a text.
- **Removing numbers**: Typically, numbers do not add much to a particular analysis (note: this is not always true however!)
- **Normalization**: stemming or lemmatizing

Let's take a look at preprocessing in Python.

II. Dictionary-based methods

What are dictionary-based methods?

Dictionaries (or "lexicons") offer perhaps the simplest method to classify the content of a particular document. In their simplest form, you are simply counting the number of words in a document that appear in the dictionary, and then dividing by the total number of words. That's it!

An important use of dictionaries: **sentiment analysis**

Table 8 Mean rank table for all datasets

3-classes			2-classes			
Pos	**Method**	**Mean Rank**	**Pos**	**Method**	**Mean Rank**	**Coverage (%)**
1	VADER	4.00 (4.17)	1	SentiStrength	2.33 (3.00)	29.30 (28.91)
2	LIWC15	4.62	2	Sentiment140	3.44	39.29
3	AFINN	4.69	3	Semantria	4.61	62.34
4	Opinion Lexicon	5.00	4	Opinion Lexicon	6.72	69.50
5	Semantria	5.31	5	LIWC15	7.33	68.28
6	Umigon	5.77	6	SO-CAL	7.61	72.64
7	SO-CAL	7.23	7	AFINN	8.11	73.05
8	Pattern.en	9.92	8	VADER	9.17 (9.79)	82.20 (83.18)
9	Sentiment140	10.92	9	Umigon	9.39	64.11
10	Emolex	11.38	10	PANAS-t	10.17	5.10
11	Opinion Finder	13.08	11	Emoticons	10.39	10.69
12	SentiWordNet	13.38	12	Pattern.en	12.61	65.02
13	Sentiment140_L	13.54	13	SenticNet	13.61	84.00
14	SenticNet	13.62	14	Emolex	14.50	66.12
15	SentiStrength	13.69 (13.71)	15	Opinion Finder	14.72	46.63
16	SASA	14.77	16	USent	14.89	44.00
17	Stanford DM	15.85	17	Sentiment140_L	14.94	93.36
18	USent	15.92	18	NRC Hashtag	17.17	93.52
19	NRC Hashtag	16.31	19	Stanford DM	17.39	87.32
20	LIWC	16.46	20	SentiWordNet	17.50	61.77
21	ANEW_SUB	18.54	21	SASA	18.94	60.12
22	Emoticons	21.00	22	LIWC	19.67	61.82
23	PANAS-t	21.77	23	ANEW_SUB	21.17	94.20
24	Emoticons DS	23.23	24	Emoticons DS	23.61	99.36

An important use of dictionaries: **sentiment analysis**

In a recent article comparing a large number of sentiment libraries, Ribeiro et al. (2016) suggests that (overall) the VADER (Valence Aware Dictionary and sEntiment Reasoner) rule-based system performed quite well (see Hutto and Gilbert 2014 for details on the algorithm).

We can call VADAR from Python after installing the **vadarSentiment** library (https://github.com/cjhutto/vaderSentiment):

```
pip install vadarSentiment
```

Let's take a look at VADAR in action.

III. Unsupervised learning

What is **unsupervised learning**?

Unsupervised learning *is a type of machine learning algorithm used to draw inferences from datasets consisting of input data without labeled responses. The most common unsupervised learning method is cluster analysis, which is used for exploratory data analysis to find hidden patterns or grouping in data.*

(MathWorks)

What's a **topic model**?

Generally speaking, **topic models** offer an **unsupervised** approach to extracting the themes (or "topics") present in a large corpus of data. Topic models are described in various ways across the literature:

1. Yet another set of clustering algorithms in a long line of clustering algorithms.
2. A data reduction technique
3. An unsupervised classification algorithm
4. (my take) A set of algorithms to automatically (sort of) learn the dictionary keywords described last week.

Topic models are all of these things!

The LDA model

> *Latent Dirichlet allocation (LDA) is a generative probabilistic model of a corpus. The basic idea is that documents are represented as random mixtures over latent topics, where each topic is characterized by a distribution over words.*
>
> *(Blei et al. 2003, p. 996)*

The Latent Dirichlet Allocation (LDA) model is the post popular topic model. It has been shown to be useful in a wide-range of research fields and has severed as the basis for many extensions over the past decade.

The LDA model

> *Latent Dirichlet allocation (LDA) is a* **generative** *probabilistic model of a corpus. The basic idea is that documents are represented as* **random mixtures** *over* **latent topics**, *where each topic is characterized by a distribution over words.*
>
> *(Blei et al. 2003, p. 996)*

The Latent Dirichlet Allocation (LDA) model is the post popular topic model. It has been shown to be useful in a wide-range of research fields and has severed as the basis for many extensions over the past decade.

The LDA's generative story

Adapted from Blei (2012)

The **big picture**: Words + model to estimate 2 matrices

The model parameters are as follows (from left to right):

α: prior distribution for the *topic distribution* (1 x K).

θ_d: the *topic distribution* for document d (1 x K)

$z_{i,n}$: topic assignment for word n in document d.

$w_{i,n}$: word n in document d.

ϕ_k: the so-called *word distribution*.

β: prior distribution for the so-called *word distribution*.

Inferring ϕ and θ in **Python**

We will use the **gensim** library to estimate topic models in Python. As per usual, open the Anaconda prompt and type:

```
pip install gensim
```

Let's see the **gensim** library in action!

Exploring your topic solution: **PyLDAvis**

What constitues a "quality topic"

If "fit" alone is a poor indicator of topic quality, the next question is how should we conceptualize (and measure) **topic quality**. The literature tends to focus on two main qualities of "good" topics:

1. They are **semantically coherent**—i.e., the semantic relationships between words aid in the understanding and interpretation of a topic.
2. They are **exclusive**—i.e., probable words for a particular topic are *unique* to that topic. Put simply, probable words allow us to *discriminate* across themes running through our corpus.

Various measures of these concepts have been proposed in the literature (see Wallach et al. 2009).

Using topics "as data"

After estimating the LDA model and getting the document-topic proportions, you can generate a wide variety of measures.

Using topics "as data"

1: Climate sensitivity
2: Fossil fuel production
3: Sea level rise
4: No scientific consensus
5: Long-term climate trends
6: Public opinion
7: U.S. politics
8: Renewable energy
9: Costs of gov't intervention
10: Environmentalism
11: Climate models
12: Solar forcing and clouds
13: Temperature data
14: Scientific misconduct
15: Gov't agencies
16: Alarmism
17: International relations
18: Agricultural industry
19: Human health
20: Corporations & the environ.
21: Urban development
22: Reuse & recycle
23: Nuclear power
24: Green jobs
25: Econ. impact of climate policy
26: Monckton
27: IPCC integrity
28: Storms
29: Emissions reduction
30: Plant impacts
31: Int'l development
32: Tax & spend
33: Conservation
34: Forest impacts
35: Cap & trade
36: Public transportation
37: Climate adaptation
38: EPA
39: Law
40: State climate reports
41: State clmate policy
42: Acidification
43: Disaster costs
44: International agreements
45: Pollution
46: Endangered species
47: Fuel standards

Decisions, decisions, decisions

We need to make a number of decisions when estimating an LDA model. Specifically, some key questions include:

1. How will you preprocess the text prior to inference?
2. How will you choose (or tune) hyperparameters?
3. How many topics will you choose? (difficult decision!)
4. How will you validate the model? (difficult decision!)

What we didn't have time to cover!

Fig. 1 An overview of text as data methods.

References I

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022.

Hutto, C. and Gilbert, E. (2014). Vader: A parsimonious rule-based model for sentiment analysis of social media text.

Ribeiro, F. N., Araújo, M., Gonçalves, P., André Gonçalves, M., and Benevenuto, F. (2016). Sentibench - a benchmark comparison of state-of-the-practice sentiment analysis methods. *EPJ Data Science*, 5(1):23.

Wallach, H. M., Murray, I., Salakhutdinov, R., and Mimno, D. (2009). Evaluation methods for topic models. In *ICML*.