Text Analytics for Business Application (TABA)

Session # 1: Introduction and Overview

Text Analytics for Batch 12, CBA @ ISB

June 2019

Sudhir Voleti

1

Course Plan - I

- Session 1: Introduction to Text Analysis
  - Regex Primer, Tokenization procedures
  - Simple Bag-of-words (BOW) representation
  - Token-Document Matrix (TDM) structures
  - Intro to *tidytext, NLTK* in Py
  - *Stringdist* introduction
- Session 2: Basic Sentiment Analysis
  - Recap and Exercise
  - Intro to Sentiment-an
  - Sentiment scoring schemes in R (tidytext)
  - Sentiment-An in Py
  - Cluster-an with text data (stringdist and k-means)

2

2

Course Plan - II

- Session 3: Vector-Space modeling
  - Factor-An primer
  - Latent Topic Modeling (LTM) primer and Simulation
  - Latent Dirichlet Allocations (LDA) notes
  - LDAtuning to find optimal #topics
- Session 4: Basic Natural Language Processing (NLP)
  - Word and sentence annotations (*Spacy*)
  - Parts of Speech (POS) tagging, chunking (*Spacy*)
  - Named Entity Recognition (NER) with *Spacy*
  - Chunking and Phrase Extractions (*UDpipe*)
- Session 5: Text Translation, Classification, miscell
  - Elementary text classifiers, Examples
  - Course summary, recap, wrap-up.

3

3

Session Plan

- Introduction – What, Why and How of Text-An
- Tokenization and DTMs
  - A **Regex** primer
  - Introducing **Tidytext**
  - The **DTM** data object
  - Weighing DTMs – TF and TFIDF
- **Basic Text-An** in BoW using Tidytext
  - Wordclouds, bar charts, co-occurrence graphs
- Basic Text-An and DTM construction with **NLTK in Py**
- Introducing **Stringdist** functionality

4

4

Introduction

5

Why Text-An

- The vast **majority of data** flooding in (~ 80%) is unstructured → much of this unstructured data is textual in form.
- **Multiple customer touch-points** – most of which yield unstructured text data
  - Call (or email) transcripts of calls to a call/service centers
  - Social media outreach (FB comments, tweets, blog entries
  - Legal or conference proceedings, press articles etc.
  - *Statutory filings* by friendly and rival firms
  - Notes by field agents, salespeople, insurance inspectors, auditors etc.
  - Open-ended questions in direct interviews, surveys. Etc.
- **No getting away** from text analysis in Business … → **Potential for unlocking** sizeable value and advantage exists.

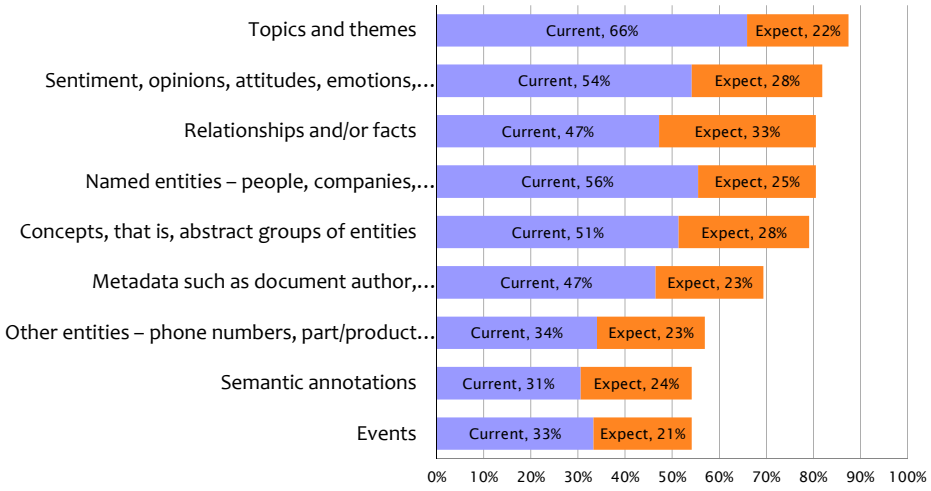
DC recap: What sources would give DC for these?

6

6

What are Firms Mining in text?

Do you currently need (or expect to need) to extract or analyze…

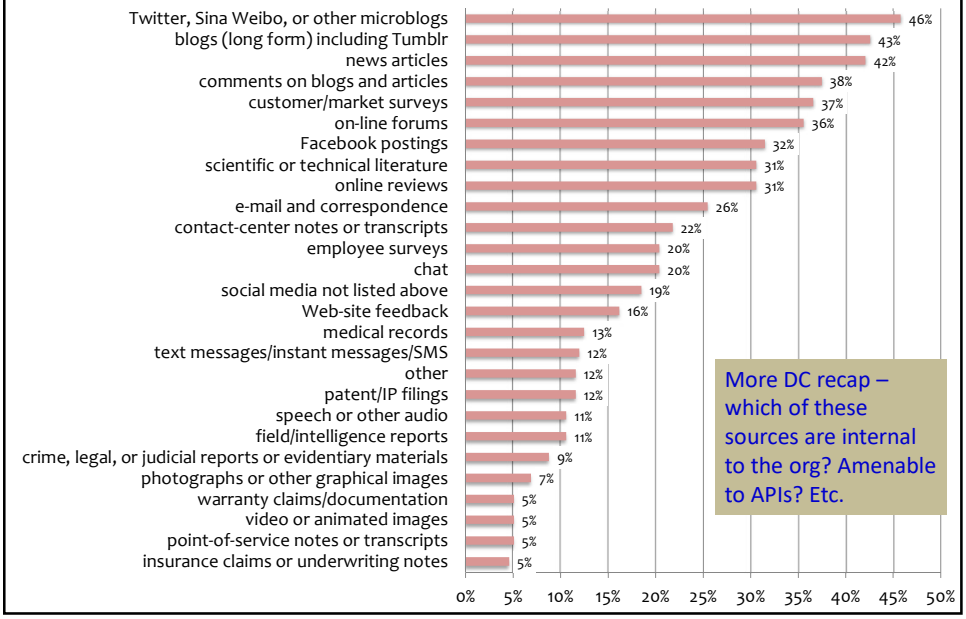


**Source: JDAT conference. JDAT is the Journal of Data Analysis Techniques*

7

Where are they mining it from?

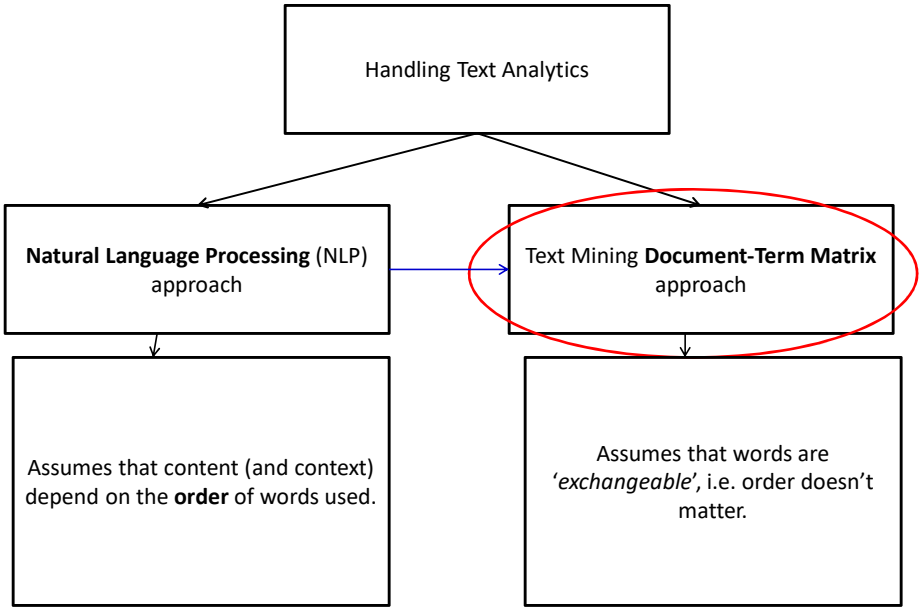
What textual information are you analyzing or do you plan to analyze?



More DC recap – which of these sources are internal to the org? Amenable to APIs? Etc.

8

The How of Text Analytics: Extant Approaches



9

Session Plan

- Introduction – What, Why and How of Text-An

- Tokenization and DTMs
  - A **Regex** primer
  - Introducing **Tidytext**
  - The **DTM** data object
  - Weighing DTMs – TF and TFIDF

- **Basic Text-An** in BoW using Tidytext
  - Wordclouds, bar charts, co-occurrence graphs

- Basic Text-An and DTM construction with **NLTK in Py**

- Introducing **Stringdist** functionality

10

10

A Regex Primer

11

A Regex Primer

- What is Regex?
- Why care about Regex in Text-An?
- Open the markdown "*Regex_basic_primer*" and examine its sections.
  - Also, see the Rstudio Regex cheatsheet pdf.
- We have [a] 12 metacharacters,
- [b] 2 character classes, [0-9], [A-Za-z]
- [c] 7 shorthand character classes, [\d, \w, \s, \t, \n, \r, \b]
- [d] 2 Anchors for positional anchoring, [^ and $]
- [e] Alternations and Repetitions, [|, {n}, +, *, ?]
- [f] Finally Grouping and Capturing ["()"]

12

12

A Regex Primer: Exercise

- Go to https://regex101.com/
- Now open the notepad 'Regex primer exercise on app.txt'. It has one paragraph of text.
- **Task:** Write regex to find:
- 1. All instances of numbers in the text (with & w/o '[]')
- 2. Acronyms (capitalized)
- 3. All instances where the letter 's' appears twice at word's ending
- 4. Sentence endings
- 5. Phone numbers.

Bottomline: Find a character pattern for quantities of interest. Then match with regex for detection, extraction or replacement.

13

13

Tokenization

14

Tokenization and its aftermath

- *Tokenization* is *t*he process of breaking up the cleaned corpus into individual terms …
- … comprising 1-, 2- or more word phrases.
- E.g., 'Ice-cream' is a 2-word token (or, *bigram*) whereas *Ice* and *cream* are 1-word tokens (a.k.a. *unigrams)*
- After a corpus is tokenized, a simple frequency table can be built…
- … that shows how many times each token occurred in each document.
- Called the Document-Token-Matrix (DTM) – its the *basic* object of analysis in text analytics.

15

The How of Text Analysis: Bag-of-words

- A passage in Shakespeare's `As You Like It':

"All the world's a stage, and all the men and women merely players:
they have their exits and their entrances;
and one man in his time plays many parts..."

- What the statistician sees:

world	stage	men	women	play	exit	entrance	time
1	1	2	1	2	1	1	1

- This is the **Bag-of-Words** representation of text.
- It's a logical, *mechanical* way to handle text input.
- And it remains, the *only* way to handle text input.

16

16

How Text Analysis: Ice-cream Dataset – Some Terminology

AH
Question 25: If Wows offered a line of light ice cream, what flavors would you want to see? Please be as specific as possible.
Vanilla, chocolate, cookies and cream, seasonal variations
vanilla & chocolate
Chocolate/peanut butter swirl; Moose Tracks
chocolate chip cookie dough
Chocolate, cookies n cream, butter pecan, vanilla fudge.
all of them my varied family member will eat anything. A really good rich vanilla is the most important becase that goes with everything.
Chocolate chip cookie dough!!!!, cinnamon, vanilla bean, cake flavored

- Each row (including the empty rows) is a *document*
- The stack of documents is a document *corpus*
- Notice the quirks of language
  - Terms with typos (e.g., 'choclate')
  - Terms in both lowercase and uppercase (e.g., 'Vanilla' and 'vanilla')
  - Punctuations ('&', '!', etc)
  - Filler words, connectors, pronouns ('all', 'for', 'of', 'my', 'to', etc.)
- *Stemming* terms: E.g., the Stem-word '*run*' replaces 'runs', 'running' etc.

17

Intro to the Tidytext package

18

Intro to Tidytext in R

- There are today many (and growing) numbers of R libraries for text-An:
  - tm, tidytext, text2vec, quanteda, …
- Each has its own pros and cons. We can choose to learn and combine their use to leverage the pros and mitigate the cons.
- Given time constraints, I'll prioritize covering perhaps the most versatile of the lot – tidytext – which operates on tidy data principles.
- Open the HTML file: *Tidytext intro for text an*
- Here's the plan, roughly speaking …

19

Markdown contents

- In this RMD, we'll cover:
- [1] Inbuilt tokenization routines for a variety of text units
- [2] Basic text manipulation using tidy data principles:
  - Grouping, Aggegation and Join ops
- [3] Bigram ops
  - Filtering, splitting and uniting, etc.
- [4] DTM casting
- [5] DTM weighing
  - TF and the TF-IDF schemes

20

Weighing the DTM – the TF mode

- Two ways of DTM weighing: TF and TFIDF.
- TF denotes 'term frequency' – a simple count of # occurrences.
- TFIDF stands for 'Term Frequency Inverse Document Frequency'
- Consider a 100 document corpus of Nokia Lumia reviews on Amazon.

DTM with a Term Freq (TF) Weighting

Documents	Terms -> Screen size	Battery life	Windows OS	...
1	3	1	0	
2	2	1	2	
...	...	...	...	
50	4	0	1	
...	...	...	...	
70	1	2	0	
...	...	...	...	
100	0	0	1	
Terms Sum	200	50	100	
Doc. Freq (DF)	2	0.5	1	

Column means, in this case

21

21

Weighing the DTM - TFIDF

- TFIDF discounts or *normalizes* the TF by document frequency → better reflects each term's presumed *importance*…
- E.g., which docs place how much emphasis on which terms, below?

DTM with TFIDF weighting

Documents	Terms --> Screen Size	Battery Life	Windows OS
1	3*(1/2) = **1.5**	1*(1/0.5) = **2**	0
2	2*(1/2) = **1**	1*(1/0.5) = **2**	2*(1/1) = **2**
...	...	...	...
50	4*(1/2) = **2**	0	1*(1/1) = **1**
...	...	...	...
70	1*(1/2) = **0.5**	2*(1/0.5) = **4**	0
...	...	...	...
100	0	0	1
Term Sum	200	50	100
Doc. Freq	2	0.5	1

Is battery life more important to #70 than screen size is to #50?

22

22

TFIDF weighing schemes

- Several schemes exist with which to weigh the DTM.
  - Our previous example is too simplistic to be practically useful.
- One common scheme is the one below:

$$TFIDF\left(\text{term}\right) = TF\left(\text{term}\right) * IDF\left(\text{term}\right)$$

Where:

$$TF\left(\text{term}\right) = \ln\left(1 + frequency\right),$$

$$IDF\left(\text{term}\right) = \ln\left(\frac{n_{documents}}{n_{docs.with.terms}}\right).$$

- We will henceforth use the inbuilt TFIDF schemes provided by various packages.
  - But know that one can invent a TFIDF scheme to suit particular situations.

23

23

Building DTMs and N-grams: Recap & Ponder

- What is a DTM? An n-gram?
- Why care about **DTM**s?
- What DTM **weighing schemes** are there? How do they differ? When to use which one?
- Reg **n-grams**, most aren't useful. E.g., "is a".
- So how to separate the wheat from the chaff, i.e., the useful n-grams from the rest?
- How to decide which “n” in n-gram is best?

24

Recapping the Tidytext exercise

- A good time to take a step back and review learnings from the tidytext exercise.
- What libraries did we call?
- What main inbuilt functions did we use?
- What user-defined functions did we use?
- What inputs and outputs to tidytext did we see?
- Time now to write a set of basic text-an functions and use them:
  - Open file '*basic text an funcs in tidytext*'

25

25

Session Plan

- Introduction – What, Why and How of Text-An
- Tokenization and DTMs
  - A **Regex** primer
  - Introducing **Tidytext**
  - The **DTM** data object
  - Weighing DTMs – TF and TFIDF
- **Basic Text-An** in BoW using Tidytext
  - Wordclouds, bar charts, co-occurrence graphs
- Basic Text-An and DTM construction with **NLTK in Py**
- Introducing **Stringdist** functionality

26

26

Basic Text-An:

Using Tidytext package

27

A few basic text-an ops *functionized*

- Func 1: Cleaning input data using a text.clean() func
- Func 2: Constructing DTMs using the dtm_build() func
- Func 3: Our first text display aid build_wordcloud()
- Func 4: Next display aid plot.barchart()
- Func 5: 3rd display aid is co-occurrence graph COG via distill.cog()
- Func 6: Combo of COG and wordcloud via build_cog_ggraph()

- Plan is to use the IBM analyst call data we have to test drive these funcs we are building.

- P.S. Pay attention to the way the funcs are defined, argument construction and their defaults, etc.

28

Introducing Term Co-occurrences

- Wordclouds are very basic, can only say so much.
- I might want to know not just which tokens occur most frequently in the corpus ...
- ... but also which tokens tend to most occur-together within a document.
- E.g., do 'Blockchain' and 'Innovation' go together? Or do 'Blockchain' and 'Solution' go together more?
- A co-occurrence graph highlights token-pairs that tend to most co-occur within documents.

29

29

Basic text-an exercise* on Ice-cream data

- Apply the basic text-an funcs to the file 'ice-cream data.txt'.
- Now answer these Qs:
- 1. What was the size of the DTM finally built?
- 2. Which stand-alone (or single) flavours seem to be the most popular?
- 3. Which flavour combinations (2 or more) seem to be popular?
- 4. What display aids did you use to answer the above questions?

30

30

Basic Text Analysis: Recap

- Q: What could we accomplish with just elementary text analysis?
- Able to rapidly, scale-ably, cheaply crunch through raw text input,
- Transform unstructured text data into token-counts, thereby
- … reducing open-ended text to a finite dimensional object (DTM).
- Able to display broad contours of which tokens arise most,
- Able to weigh differential emphasis using a simple IDF weighing scheme.
- Able to sense some simple within-document co-occurrences.

31

Session Plan

- Introduction – What, Why and How of Text-An
- Tokenization and DTMs
  - A **Regex** primer
  - Introducing **Tidytext**
  - The **DTM** data object
  - Weighing DTMs – TF and TFIDF
- **Basic Text-An** in BoW using Tidytext
  - Wordclouds, bar charts, co-occurrence graphs
- Basic Text-An and DTM construction with **NLTK in Py**
- Introducing **Stringdist** functionality

32

32

Basic Text-An in Py

33

DTM building with NLTK in Py

- We've so far focused on R for BoW.
- Let us now see a mirror to R's BoW capabilities with NLTK in Py.
- Plan is to demo basic text-an opns like:
  - tokenization,
  - stop-word removal,
  - corpus cleanup and pre-processing
  - building a (TFIDF) DTM
- using Py's NLTK module.
- Open the file "*DTM building in py.ipynb*"

34

34

DTM building in Py - review and recap

- Which modules did we invoke?
- Which major funcs do you recall?
- For tokenization?
- for pre-processing and html junk removal?
- for DTM building?
- How much time did it take?
- Can we build downstream functionality like wordclouds, COGs etc also in py?

35

35

Session Plan

- Introduction – What, Why and How of Text-An
- Tokenization and DTMs
  - A **Regex** primer
  - Introducing **Tidytext**
  - The **DTM** data object
  - Weighing DTMs – TF and TFIDF
- **Basic Text-An** in BoW using Tidytext
  - Wordclouds, bar charts, co-occurrence graphs
- Basic Text-An and DTM construction with **NLTK in Py**
- Introducing **Stringdist** functionality

36

36

Introducing *Stringdist*

37

Exploring Stringdist Functionality

- We've seen text tokenization at the word level. Now we head to the character level.
- Stringdist R package contains 10 ways to measure distances between two strings.
- E.g., "busness" and "business" - how would you measure the inter-string 'distance' between them?
- What possible use-cases and biz applications can this lead to, besides?
- Open file '*Stringdist functions in R.Rmd*' and follow me.

38

38

Stringdist Functionality - Review and Recap

- What is *stringdist*? What does it do?
- What packages did we use?
- What main distance functions and metrics do you recall?
- What downstream applications can inter-string distances lead to?
- **Hint**: Think of cluster-an applications
- Any other comments?

39

39

Q & A

40