LING 410X: Language as Data

Semester: Spring '18

Instructor: Sowmya Vajjala

Iowa State University, USA

20 February 2018

Class Outline

- Last class' exercise: Discussion
- Article discussion (Thursday's reading)
- Overview of Macro Analysis of Texts
  - Text classification
  - Topic modeling
  - Clustering
- Reminder: Assignment 3 submission due this week

Last Class' exercise - 1

- Q1: KWIC modification
- Q2: Getting most frequent ngrams and plotting.

Quick notes on Assignment 3

- For Q1: If you revisit the slides and code from Week 5, you will be able to do this question.
- ngram package: You have to first install it (tools $->$ install packages) and then load it into Rstudio (library(ngram)) before starting to use it.
- For Q2, if you take a look at the slides for last week's classes again, you should be able to finish it without any additional information.
- Note: I will not ask you to anything brand new that was not mentioned in the class. You have to be patient and be willing to go back and see the notes you wrote and the slides/code I shared.

Open tutorial on 22nd Feb

- Time: 5-7pm, Ross 137 (Our thursday's classroom)
- Theme: I will prepare some problems/exercises; You can ask your questions
- Format: Not like a classroom session. You can come in and go out as you want, and work on 410X related stuff, ask questions.
- You can also help each other.
- Post specific questions on the forum titled "Tutorial Session on Feb 22nd", by thursday morning.

Article Discussion
https://goo.gl/qhT3u4

Article discussion: Thursday's reading

- What is the article about?

Article discussion: Thursday's reading

- What is the article about?
- Did the article mention anything about the size of data collected (number of tweets)?

Article discussion: Thursday's reading

- What is the article about?
- Did the article mention anything about the size of data collected (number of tweets)?
- What is the scale of sentiment in the article's data?

Article discussion: Thursday's reading

- What is the article about?
- Did the article mention anything about the size of data collected (number of tweets)?
- What is the scale of sentiment in the article's data?
- Where do they get the sentiment score from?

Article discussion: Thursday's reading

- What is the article about?
- Did the article mention anything about the size of data collected (number of tweets)?
- What is the scale of sentiment in the article's data?
- Where do they get the sentiment score from?
- What is your general perception about the project - do you think it makes sense?

Article discussion: Thursday's reading

- What is the article about?
- Did the article mention anything about the size of data collected (number of tweets)?
- What is the scale of sentiment in the article's data?
- Where do they get the sentiment score from?
- What is your general perception about the project - do you think it makes sense?
- Do you think it works successfully?

Quick Recap of what we learnt so far

Reading in text content into R

- Different file formats: mostly .txt, but saw examples for doc, pdf, html
- I uploaded examples for accessing twitter (I can have an optional session for those who want to know this, after the break)
- How to get text from specialized libraries for different websites such as Guardian, NYT.

Reading in text content into R

- Different file formats: mostly .txt, but saw examples for doc, pdf, html
- I uploaded examples for accessing twitter (I can have an optional session for those who want to know this, after the break)
- How to get text from specialized libraries for different websites such as Guardian, NYT.
- Your task: Look for other such special libraries for Wikipedia or Gutenberg.org, and practice working with them by looking at the documentation provided by the creators.
- Figure out how to learn about other such libraries and whether you need anything like that.

Analyzing textual data: Micro Analysis

- Counting words
- Sorting them in terms of frequencies
- Studying the distribution of words in a text
- Doing some basic plots of dispersion, distribution and frequency

Chapters 2–4 in Textbook.

Analysing textual data: Meso Analysis

- Lexical variety: average word frequency, type token ratio
- Measuring rare-word occurrences
- Looking for keywords in context
- Knowing how to get ngrams, check for overlapping ngrams between two lists

Chapters 5–9 in textbook.

What is Macro Analysis?

Macro Analysis

- Going beyond a single text or a small collection of texts and working with larger collections.
- Instead of focusing on specifics of a text, focus on figuring out general patterns across texts
- Finding out ways to "group" texts into some pre-defined number of groups.
- Coming up with methods to create "aggregated knowledge" about texts.
- Finding out how to evaluate whether our aggregated knowledge is accurate.

What sorts of analysis exist?

Primarily, three forms:

- Text categorization/classification (when we know what the groups are)
- Text clustering (when we do not know what the groups are)
- Topic modeling (when we want to know what is the over arching theme in a collection of texts)

Text Classification

Text Classification

- In text classification, we know what the possible categories for our texts can be.
- We also have a collection of texts, already assigned these categories.

Text Classification

- In text classification, we know what the possible categories for our texts can be.
- We also have a collection of texts, already assigned these categories.
- We want to create a "model" of categorization based on this pre-categorized text collection such that the model can "predict" or "assign" categories to new texts.

Examples of text classification

- Classifying the text of a tweet into one of the 5 languages: English, French, German, Chinese, Arabic. (language identification)
- Predicting whether a review about a product on amazon.com is positive or negative (or neutral) about the product (sentiment)
- Telling whether an email is a spam message or a normal message
- Whether a webpage's text is suitable for children or not.

... and so on.

Where do we get those "pre-assigned" categories?

- sometimes, historical data (e.g., if we want to predict weather based on past trends)

Where do we get those "pre-assigned" categories?

- sometimes, historical data (e.g., if we want to predict weather based on past trends)
- sometimes, ask human annotators to categorize small collection of text (e.g., if I keep clicking spam for all spam I see in my inbox, after a while, there is enough data for automatic spam classification)

Where do we get those "pre-assigned" categories?

- sometimes, historical data (e.g., if we want to predict weather based on past trends)
- sometimes, ask human annotators to categorize small collection of text (e.g., if I keep clicking spam for all spam I see in my inbox, after a while, there is enough data for automatic spam classification)
- For most of the known classification tasks, there are some standard datasets one can use to develop classification models
- Eventual evaluation: when you actually use these classifiers somewhere, and you learn something.. or in ecommerce, if the user is satisfied, and revenue is increased.

What kind of "features" or "patterns" will the model learn?

- Word occurrences are the most commonly used patterns.
- We can also look at word sequences (Ngrams)
- Part of speech tag patterns
- All of them put together
- Or some other stuff, such as some specialized linguistic patterns (e.g., number followed by some preposition, three adjectives preceding a noun etc.)

... we will focus on the first kind of features in this class.

How does the machine "learn" these patterns?

- Lot of machine learning algorithms are already in place to "learn" from several forms of data.
- Our job is to pick a couple of them and compare them with our data, and choose the best one.

How does the machine "learn" these patterns?

- Lot of machine learning algorithms are already in place to "learn" from several forms of data.
- Our job is to pick a couple of them and compare them with our data, and choose the best one.
- Good thing about this is: it is like driving a car. you do not have to know all the internal working details to drive it.
- Bad thing: you end up working with a black box.

Clustering

Clustering

- Let us say all you got is 10,000 tweets from different Tennis players. Someone now asks you to sort them into 5 groups based on their content.
- How do you do that?

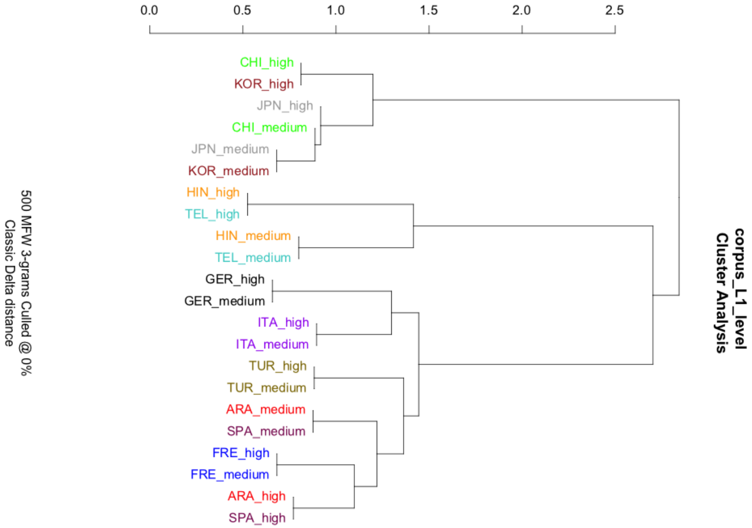
Clustering

- Let us say all you got is 10,000 tweets from different Tennis players. Someone now asks you to sort them into 5 groups based on their content.
- How do you do that?
- The main idea is: if we have a reason to believe that there are groups and not just a large collection, we should have some notion of "belongingness to a group"

Clustering

- Let us say all you got is 10,000 tweets from different Tennis players. Someone now asks you to sort them into 5 groups based on their content.
- How do you do that?
- The main idea is: if we have a reason to believe that there are groups and not just a large collection, we should have some notion of "belongingness to a group"
- If we have some notion of such belongingness, then it is easy to form groups. Members of same group are similar/closer to each other i.e., belong together.
- members of different groups should be away from each other.

Clustering - Example



classification and clustering

- Similarity: we have a collection of texts, we know what are the features we want to look for (words)

classification and clustering

- Similarity: we have a collection of texts, we know what are the features we want to look for (words)
- Difference: In the case of classification, we have a few examples classified into some categories, and we want the machine to learn to classify like this for new texts.

classification and clustering

- Similarity: we have a collection of texts, we know what are the features we want to look for (words)
- Difference: In the case of classification, we have a few examples classified into some categories, and we want the machine to learn to classify like this for new texts.
- In clustering, we do not have such examples, we have no idea how many groupings or possible. We want the machine to figure that out AND do the grouping.

Topic Modeling

Topic Modeling

- Idea 1: each document is a mixture of topics
- Idea 2: each topic can be represented by groups of words associated with it.

Topic Modeling

- Idea 1: each document is a mixture of topics
- Idea 2: each topic can be represented by groups of words associated with it.
- Idea 3: a word may be very important in one topic, but may not be so important in another topic

Topic Modeling

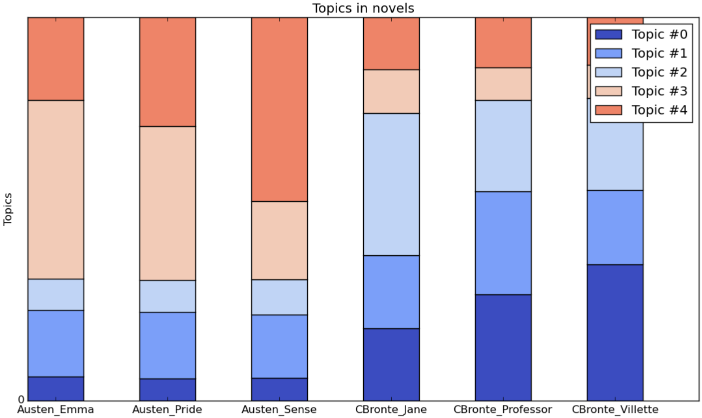
- Idea 1: each document is a mixture of topics
- Idea 2: each topic can be represented by groups of words associated with it.
- Idea 3: a word may be very important in one topic, but may not be so important in another topic
- So how about looking at a collection of documents, extracting main topics from them and forming clusters of words based on topical similarity?

Topic Modeling output - example

TOPIC_1	0.01986	TOPIC_2	0.19236	TOPIC_3	0.10298	TOPIC_4	0.05993
div	0.01981	org	0.03005	entity	0.04879	storm	0.02926
dns	0.01450	elasticsearch	0.02632	world	0.04836	get	0.01910
zmq	0.01031	index	0.02618	get	0.04317	topology	0.01776
ac	0.00892	field	0.02019	bukkit	0.03213	field	0.01538
top	0.00822	name	0.01350	block	0.02661	org	0.01369
name	0.00798	query	0.01329	craft	0.02657	task	0.01290
type	0.00794	value	0.01318	event	0.02349	apache	0.01270
file	0.00731	builder	0.01264	item	0.02103	fields	0.01080
echo	0.00703	request	0.01156	server	0.01868	conf	0.01072
margin	0.00681	search	0.01139	player	0.01607	backtype	0.01066
TOPIC_5	0.02106	TOPIC_6	0.15545	TOPIC_7	0.24803	TOPIC_8	0.12545
version	0.03604	get	0.03997	channel	0.05271	android	0.03539
artifact	0.02302	index	0.02490	netty	0.02303	view	0.02652
group	0.02123	request	0.02102	buffer	0.01939	name	0.02159
clojure	0.02094	search	0.02014	handler	0.01749	get	0.01821
java	0.01955	query	0.01879	http	0.01609	item	0.01725
org	0.01591	field	0.01858	get	0.01413	action	0.01699
pendency	0.01306	org	0.01670	io	0.01282	menu	0.01521
contrib	0.01211	response	0.01553	socket	0.01203	com	0.01238
test	0.01191	builder	0.01425	buf	0.01157	layout	0.01144
file	0.01037	test	0.01327	license	0.01048	text	0.01141
TOPIC_9	0.04541	TOPIC_10	0.02947				
get	0.01488	java	0.01628				
name	0.01327	can	0.01540				
clojure	0.01277	will	0.01152				
session	0.01222	com	0.00850				
key	0.01214	just	0.00725				
val	0.01194	plugin	0.00719				
fn	0.00994	one	0.00680				
method	0.00979	also	0.00621				
type	0.00959	use	0.00611				
first	0.00943	using	0.00610				

source: http://shritir.weebly.com/uploads/2/6/3/4/26348989/2922455.png?1410730090

Topic Modeling output - another example



source: `https://de.dariah.eu/tatom/_images/plot_doctopic_stacked_bar.png`

clustering and topic modeling

- Similarity: in both, we do not know what are the groups we are looking for and how many are they?

clustering and topic modeling

- Similarity: in both, we do not know what are the groups we are looking for and how many are they?
- Difference: In clustering, we generally talk about grouping texts together based on word patterns in them.

clustering and topic modeling

- Similarity: in both, we do not know what are the groups we are looking for and how many are they?
- Difference: In clustering, we generally talk about grouping texts together based on word patterns in them.
- In topic modeling, we generally talk about clustering words together based on the notion that all words in a cluster represent the vocabulary of a topic.

Doing all these in R

- There are several options.
- tm is a popular library used to do such analysis, and there are several libraries that depend on this.
- Textbook uses other libraries (different ones for different tasks)
- I will use textbook examples where they are simpler, but make you work with tm for assignments (4–6).

Doing all these in R

- There are several options.
- tm is a popular library used to do such analysis, and there are several libraries that depend on this.
- Textbook uses other libraries (different ones for different tasks)
- I will use textbook examples where they are simpler, but make you work with tm for assignments (4–6).
- Assignments 4 and 5 are on text classification and topic modeling respectively, and you have to follow a R tutorial and write down reports.
- Assignment 6 - you have to create some visualizations following instructions given.

How should we do all these in R?

Typical steps:

- First, split your corpus into two groups: training data (to construct your model), testing data (to test your model accuracy)
- Read in your collection of training texts (and their categories, if it is classification)

How should we do all these in R?

Typical steps:

- First, split your corpus into two groups: training data (to construct your model), testing data (to test your model accuracy)
- Read in your collection of training texts (and their categories, if it is classification)
- Do your pre-processing (typical: lower casing, removing punctuations, removing numbers, removing "stop words", stemming)

How should we do all these in R?

Typical steps:

- First, split your corpus into two groups: training data (to construct your model), testing data (to test your model accuracy)
- Read in your collection of training texts (and their categories, if it is classification)
- Do your pre-processing (typical: lower casing, removing punctuations, removing numbers, removing "stop words", stemming)
- Create a term-document matrix

How should we do all these in R?

Typical steps:

- First, split your corpus into two groups: training data (to construct your model), testing data (to test your model accuracy)
- Read in your collection of training texts (and their categories, if it is classification)
- Do your pre-processing (typical: lower casing, removing punctuations, removing numbers, removing "stop words", stemming)
- Create a term-document matrix
- Use this matrix along with any existing classification/clustering/topic modeling algorithm (100s of them exist. Some common algorithms for classification: naive bayes, support vector machines, decision trees, logistic regression etc.)

How should we do all these in R?

Typical steps:

- First, split your corpus into two groups: training data (to construct your model), testing data (to test your model accuracy)
- Read in your collection of training texts (and their categories, if it is classification)
- Do your pre-processing (typical: lower casing, removing punctuations, removing numbers, removing "stop words", stemming)
- Create a term-document matrix
- Use this matrix along with any existing classification/clustering/topic modeling algorithm (100s of them exist. Some common algorithms for classification: naive bayes, support vector machines, decision trees, logistic regression etc.)
- Use this model to make predictions on the test data if it is classification
- Figure out what to infer from topic clusters or document clusters otherwise

Thursday

- Read these before coming to the class:
  1. `https://vvvvw.aaai.org/Papers/Workshops/1998/WS-98-05/WS98-05-001.pdf`
  2. An encyclopaedia article I wrote recently, "Machine Learning and Applied Linguistics" just to get some perspective on real-world applications of this kind of macro analyses in a specific domain. It is uploaded on Canvas: Modules - Week 7
- Post a summary (3-4 bullet points) of each reading before you come on Thursday in the forum with Today's date.
- We will continue this discussion, with some discussion exercises.