



MET CS688 C1

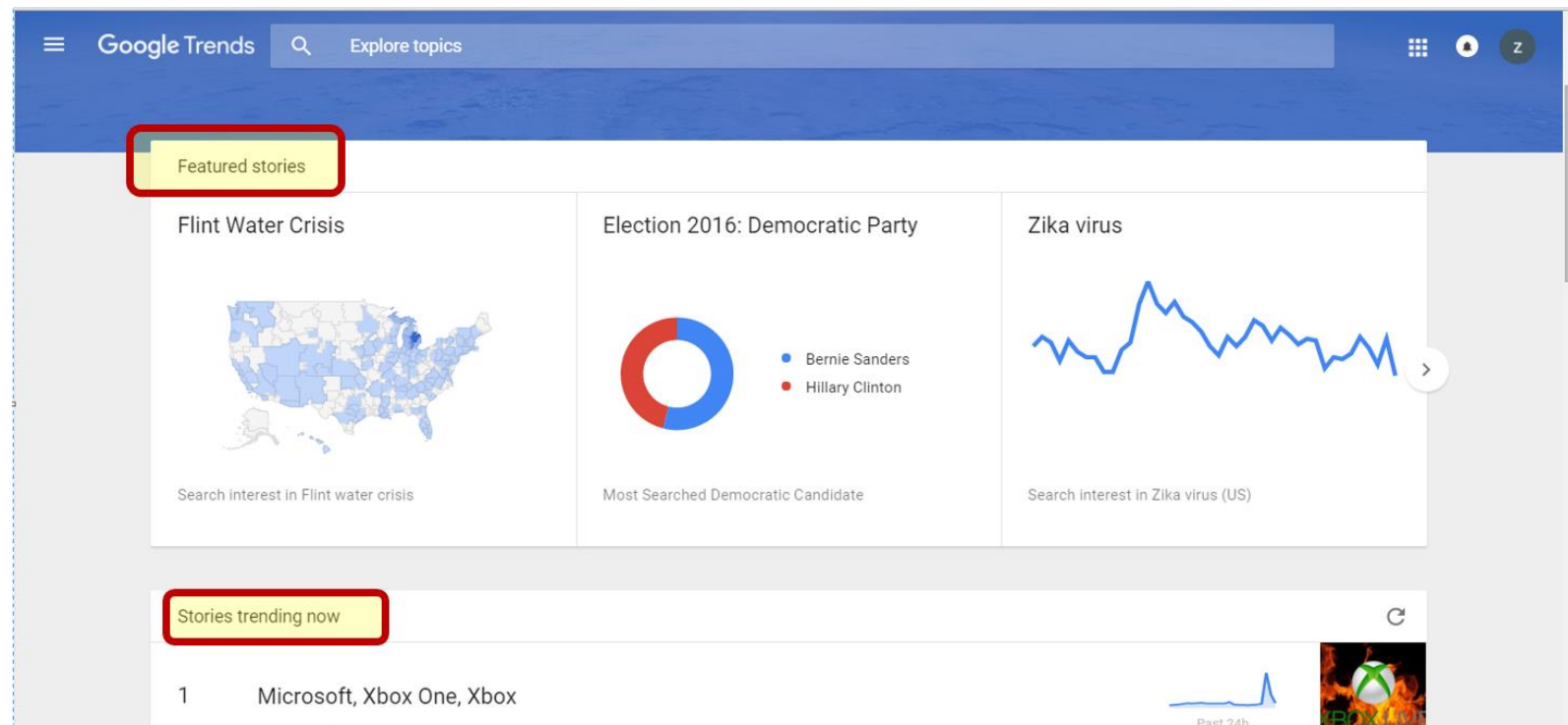
WEB ANALYTICS AND MINING

ZLATKO VASILKOSKI

CLASS 3

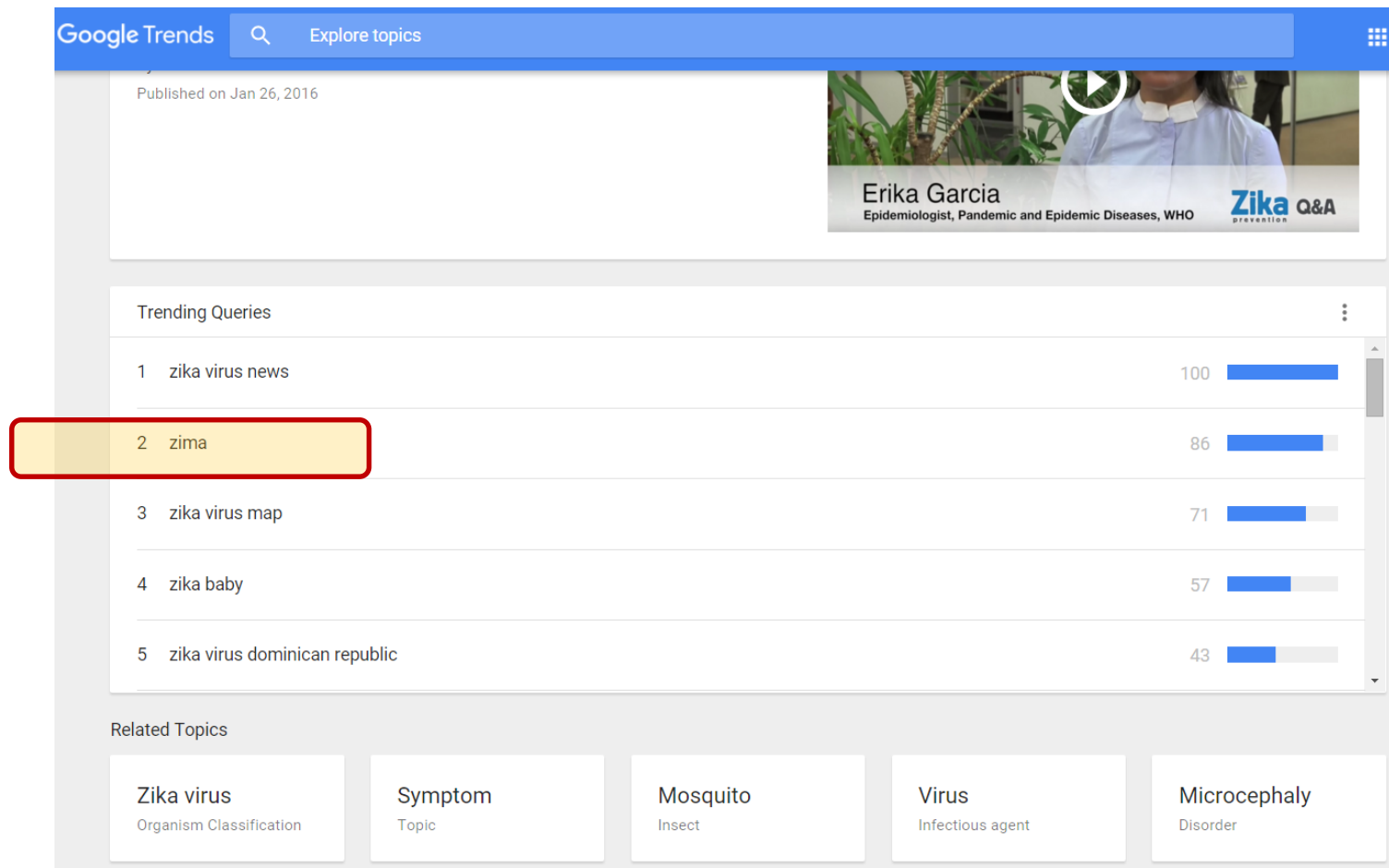
Google Trends

- **Google Trends** shows the ups-and-downs of the public's interest in a particular topic.
- This website <https://www.google.com/trends/> contains featured stories that you can select from, such as the 2016 Elections or the US search interest in Zika virus, as illustrated here in the past, as well as many new trending stories at the moment.



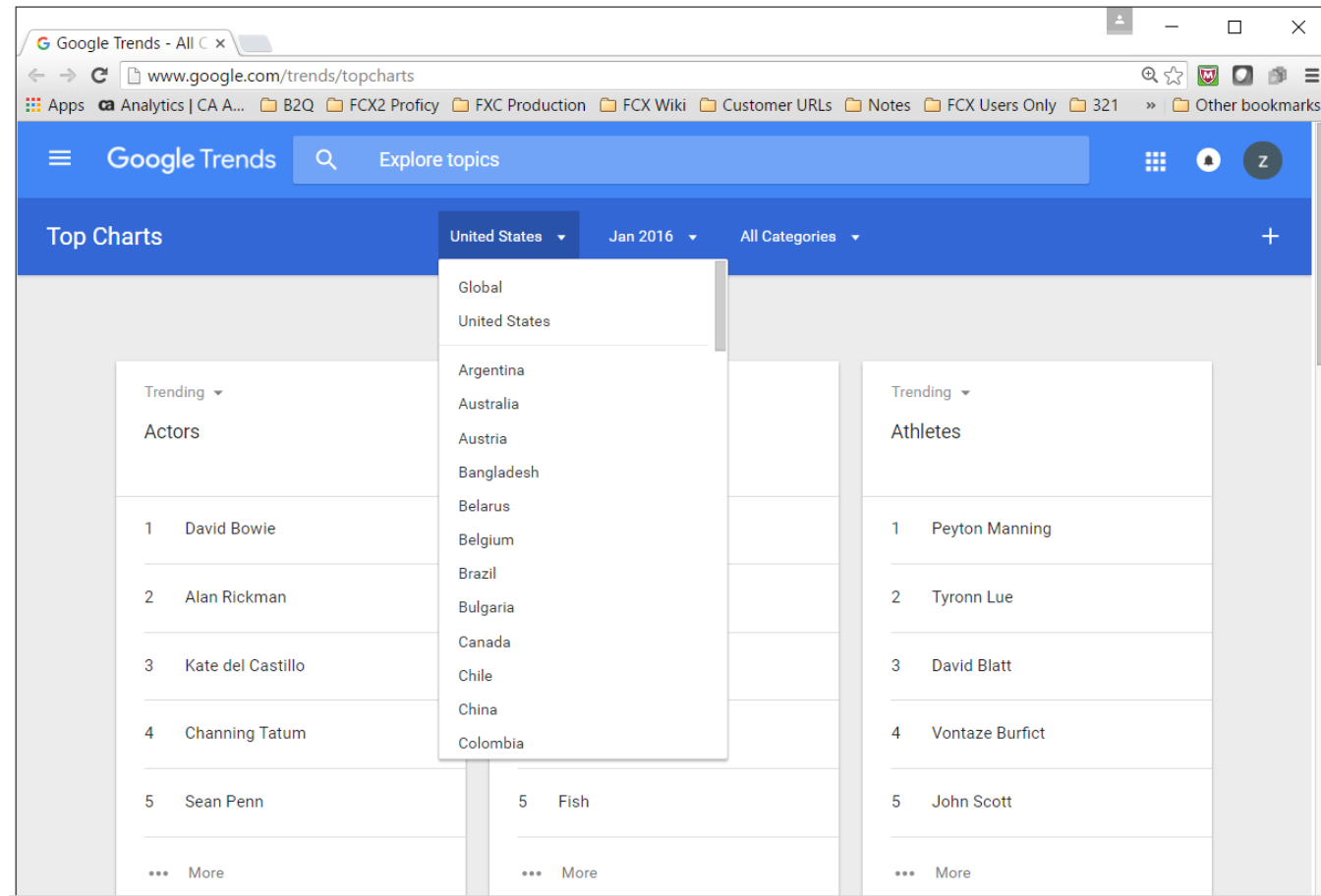
Google Trends

- The same page also includes the trending queries people used related to this particular topic. Note #2!

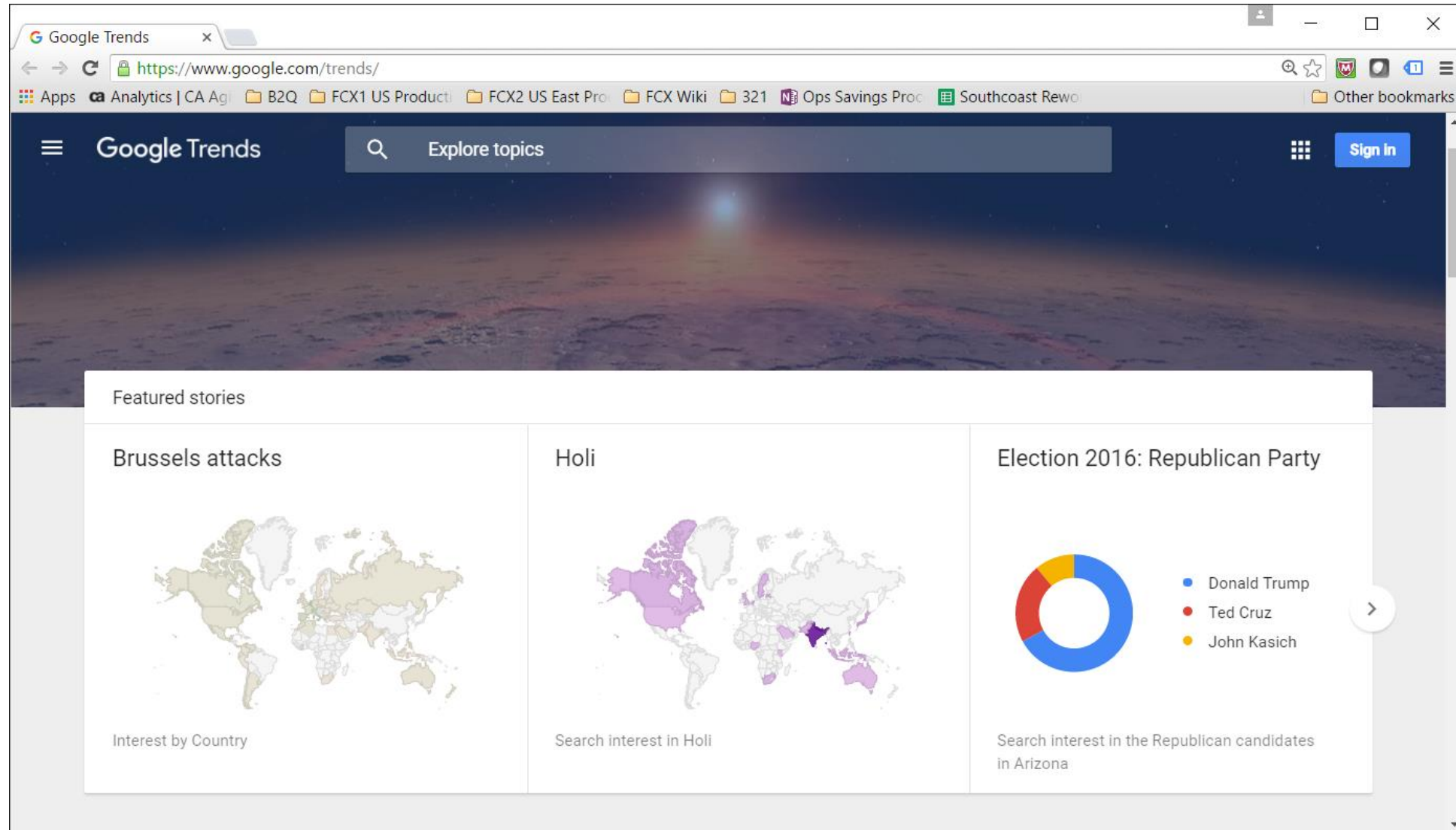


Google Trends

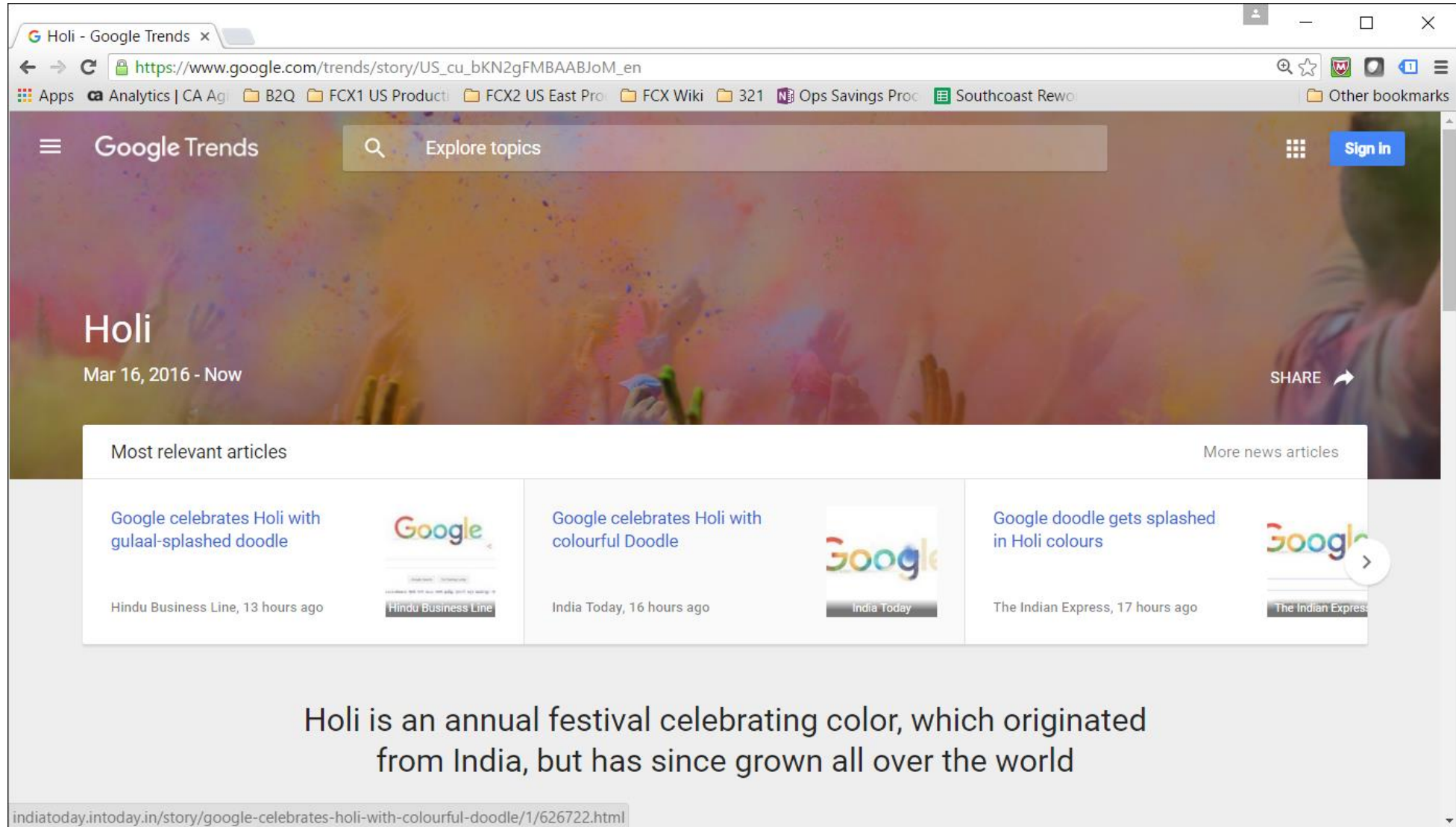
- Trends can be searched and trending keywords per category shown.



Google Trends – What is Holi?



Interesting to learn what is Holi



The screenshot shows the Google Trends interface for the search term "Holi". The page features a large background image of people celebrating Holi with colorful powder. The title "Holi" is prominently displayed, along with the date range "Mar 16, 2016 - Now". A "Sign in" button is visible in the top right corner. Below the main header, there is a section titled "Most relevant articles" which lists three articles from Hindu Business Line, India Today, and The Indian Express, each accompanied by a thumbnail image of the Google Holi doodle. The URL in the address bar is https://www.google.com/trends/story/US_cu_bKN2gFMBAABJoM_en. The browser's bookmark bar shows various folders and links, including "Apps", "Analytics | CA Ag", "B2Q", "FCX1 US Producti", "FCX2 US East Pro", "FCX Wiki", "321", "Ops Savings Proc", "Southcoast Rewo", and "Other bookmarks".

Holi

Mar 16, 2016 - Now

Most relevant articles

Google celebrates Holi with gulaal-splashed doodle

Hindu Business Line, 13 hours ago

Google celebrates Holi with colourful Doodle

India Today, 16 hours ago

Google doodle gets splashed in Holi colours

The Indian Express, 17 hours ago

Holi is an annual festival celebrating color, which originated from India, but has since grown all over the world

https://www.google.com/trends/story/US_cu_bKN2gFMBAABJoM_en

Google Correlate

Google introduced in 2011, inverse form Google Trends.

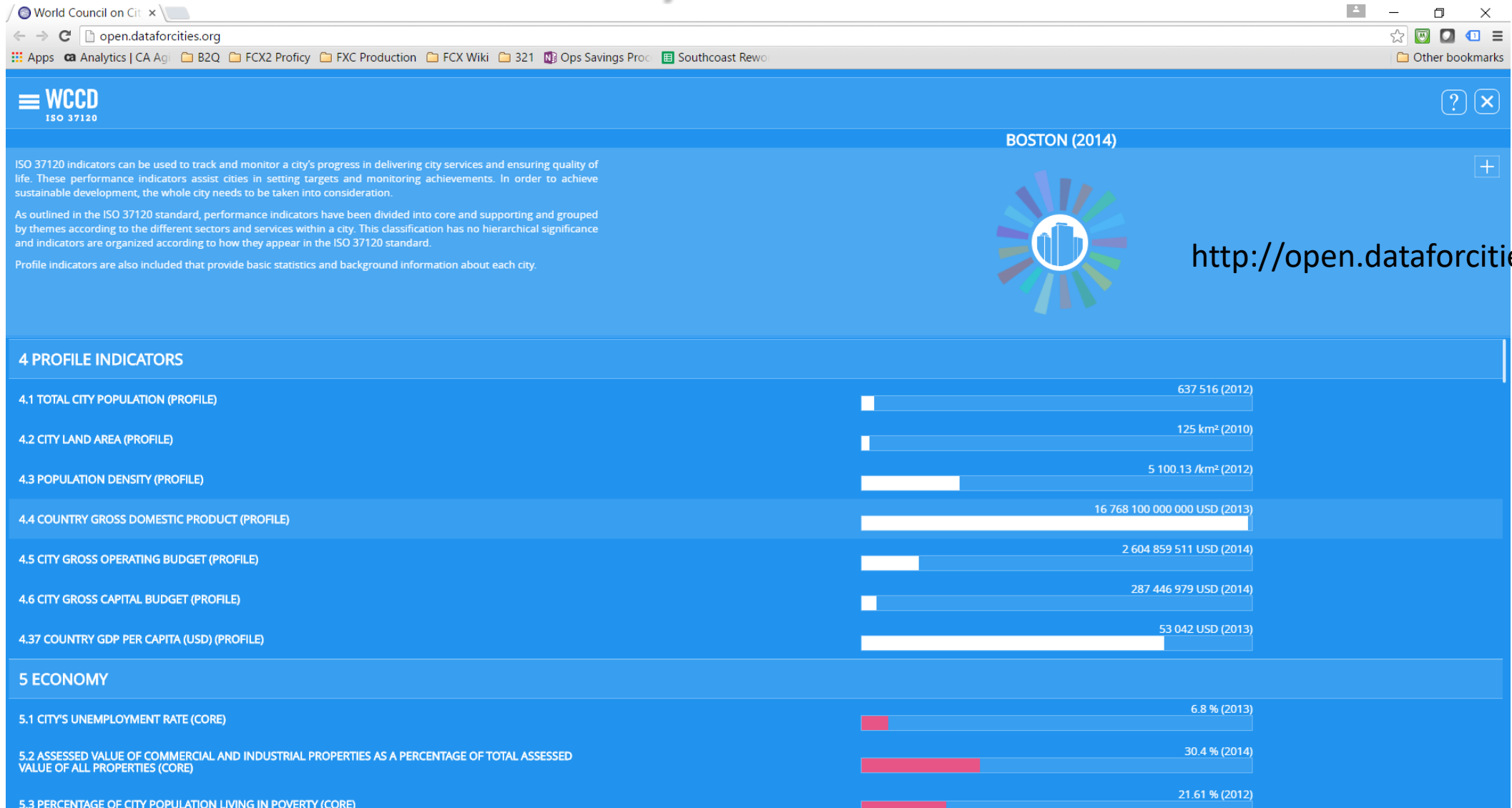
1. Compare US States – terms popularity per state

- Gives you the best correlated terms and the state distribution
- Marketing campaigns and content strategies can be built around this

2. Time Series – terms popularity change over time

- Seasonal changes, holidays etc. in trending pattern
- Interesting to notice how a US holiday such as Halloween becomes trendier in other countries such as Portugal.

Useful for City Indicators studies



Lab project: Google Correlate

- Select what do you want to use:
 1. **Compare US States** – terms popularity per state
 - Analyze in which US state is most appropriate to advertise your type of business based on your key words.
 2. **Time Series** – terms popularity change over time
 - Create a seasonal business marketing campaign for a foreign country
- Submit your result (screenshots & description of your small business) as PowerPoint. Show
 - Screenshots of your term search (including the other most correlated terms)
 - Screenshots of the geographical area
 - Marketing campaign conclusion

Lab project: Google Correlate - Compare US States

- Task: Create small company (startup) marketing campaign based on data analysis from Google Correlate. Focus your marketing efforts on particular geographic areas and find out where to start with the introduction of your new type of product.
 - Invent a small business (Type, and distinctive product)
 - Select (1 or 2) terms (key words) that distinct your product from the others (Note these common terms are the ones people typically do Google search for other reasons than your company)
 - Compare US States – terms popularity per state
 - Analyze in which US state is most appropriate to advertise your type of business based on your key words.
- Submit your result (screenshots & description of your small business) as PowerPoint. Show
 - Screenshots of your term search (including the other most correlated terms)
 - Screenshots of the geographical area
 - Marketing campaign conclusion

Lab project: Google Correlate - Time Series

- Task: Create small company (startup) marketing campaign based on data analysis from Google Correlate. Focus your marketing efforts on particular geographic areas and find out where to start with the introduction of your new type of product.
 - Invent a small business (Type, and distinctive product)
 - Select (1 or 2) terms (key words) that distinct your product from the others (Note these common terms are the ones people typically do Google search for other reasons than your company)
 - Time Series – terms popularity change over time
 - Create a seasonal business marketing campaign for a foreign country
- Submit your result (screenshots & description of your small business) as PowerPoint. Show
 - Screenshots of your term search (including an offset of few weeks before and after)
 - Marketing campaign conclusion

Discussion - Motivation

Please have a look at the following topics and related questions. Please provide your comments on Blackboard E-Discovery Discussion.

Motivation: Electronic discovery or e-discovery refers to discovery in litigation or government investigations which deals with the exchange of information in electronic format. In April 2012, a state judge in Virginia issued the first state court ruling allowing the use of predictive coding in e-discovery in the case Global Aerospace, Inc., et al. v. Landow Aviation, LP, et al. The Global Aerospace case pertained to an accident that occurred during a winter storm in 2010, in which several hangars collapsed at the Dulles Jet Center.

Tasks:

- Describe the most popular legal analysis (e-discovery) platforms available with focus on their types of legal data (digital text, OCR, audio, video) that they can deal with text mining and text analytics features of these platforms.

In particular focus your attention on these key features:

- visualization of their data analytics
- learning technology (predictive coding) to cluster conceptually related documents
- how much data they can review
- how fast is their timeline effectiveness (dealing with large data in very short timeline)
- how complex customer reviews these platforms allow for
- compare professional platform features to open source solutions

Based on all of these features what e-discovery platform would you recommend to a

- smaller size law firm
- large law firm

Provide formal arguments and citation for your response if necessary. Please have a look at other people's comments too.

Text Mining

Content:

- Preprocessing and content extraction
- Searching and fuzzy string matching
- Clustering text
- Classification, categorization, and tagging
- Question answering systems

The general approach is to represent the text analytics tasks into a mathematical ones and apply standard (100 years old) math techniques implemented as a computer algorithm.

Note: that the math notes in this class are for your reference only. They refer to the use in various algorithms and techniques used throughout the modules. It is not expected that you need to understand all of the math mentioned. It is intended to give more of an idea what is behind some of the more complicated concepts such as document matching in context space for example, if some of you have additional interest.

An illustrative example

- Consider an application that allows a user to enter query to search all the files on your computer that may contain the keyword in the text.
- The first thing you would need to do is to transform all these files into a common format so that you only have to worry about one format internally.
- This process of standardizing the many different file types to a common text-based representation is called preprocessing. Preprocessing includes any steps that must be undertaken before inputting data into the library or application for its intended use.
- As a primary example of preprocessing, we'll look at extracting content from common file formats. We'll discuss the importance of preprocessing and introduce an open source framework for extracting content and metadata from common file formats such as MS Word and Adobe PDF.

Text Mining (Text Analytics)

Most of today's data is unstructured (in a form of mixed media)

- text, images, audio data, etc.

The information in the data eventually is retrieved in a textual form – the way we communicate and express ourselves.

- The most optimal, most compact way of keeping the content of the information we use on a daily basis.
- The goal of text mining (text analytics) is to obtain (retrieve) the essential information contained in the text data by using statistical pattern learning.
 - Goal - content management and information organization
 - Tools - applying the knowledge discovery to tasks such as
 - topic detection
 - phrase extraction
 - document grouping, etc.

Text Mining (Text Analytics)

- Typical steps involved in information retrieval
 - Structuring the input text
 - Preprocessing, parsing, searching for some linguistic features and the removal of others.
 - Obtaining and matching patterns within the structured data (text).
 - Linear algebra, Machine Learning (regression, classification, clustering).
 - Evaluation and interpretation of the results.
 - Statistics (Precision, Recall, F score etc.).

Introduction

Example:

- We have a large set of digital text documents (books, newspapers, emails, etc.).
- We want to extract useful information from this massive amounts of text quickly.
- To summarize the main themes and identify those of most interest to us or to particular people (clients).
- Using automated algorithms, we can achieve this much quicker than a human could.
- To this we refer to as knowledge distillation:
 - Text classification/categorization
 - Document clustering—finding groups of similar documents
 - Information extraction
 - Summarization

Text processing

- Why Is Processing Text Important?
 - All of our digital communication, language, documents, storage of knowledge etc. is in a textual form.
 - Understanding the information content of this data part of our daily job.
 - Increase productivity, find relevant information quicker etc.
- Levels of text processing, by increasing complexity:
 - Characters
 - Words
 - Multiword text and sentences
 - Multi sentence text and paragraphs
 - Documents
 - Multi document text and corpora

What Makes Text Processing Difficult?

- Challenges at many levels, from handling character encodings to inferring meaning.
- On a small scale, much easier to deal with:
 - Search the text data based on user input and return the relevant passage (a paragraph, for example).
 - Split the passage into sentences.
 - Extract “interesting” things from the text, for example the names of people.
- On a large scale, much harder to deal with:
 - Linguistics as syntax (grammar), understanding the rules about categories of words and word groupings.
 - Language translation. Have you tried Google translate?
 - “Understanding” the text content like people do.
- We are far from the famous Turing Test—a test to determine whether a machine's intelligent behavior is distinguishable from that of a human.

Searching Through Text Data

- Simplest possible text analytics task
 - Search digital text documents for a content of a particular word.
- Typical approach
 - A linear scan (Grepping - half a century old way since 1970, UNIX, “sed”, “awk”):
 - Going through all of the text and checking for the appearance of that word by matching the exact pattern of letters contained in the query word with every single word in the text we search.
- No better way for a small set of documents.
- Nowadays impractical since the data accumulated online is so large.

Commonly Used Terms in Text Mining

- **Index** – This refers to cataloging (indexing) the documents in advance, to avoid linearly scanning the texts for each query.
- **Terms** – Terms are the indexed units (words) that we are searching for.
- **Term-Document Incidence Matrix** – This is a (binary) table that relates the terms and the documents. The transposed version is called **Document-Term Incidence Matrix**.
- **The Boolean Retrieval Model** – This is one of the basic models used to retrieve information from the term-document incidence matrix by
 - Uses term-document incidence matrix so it avoids linear search of all documents.
 - Uses binary operations which are the fastest possible.

Commonly Used Terms in Text Mining

- **Indexed Notation of a Matrix** – TDM is sparse, more memory economical (thus faster) to keep just the nonzero entries.
- **Dictionary (Vocabulary or Lexicon)** – This keeps all the unique terms.
- **Postings (Posting Lists)** – Relates dictionary (unique) terms with document ID (the way we numbered documents).
- **Ranked Retrieval Model** – Contrasting the Boolean model are ranked retrieval models such as the vector space model. These models use free text queries (more complicated math SVD), and the system decides which documents best satisfy the queries.

Incidence Matrix

Documents	Antony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth	...
Terms	Antony	1	1	0	0	0	1
Brutus	1	1	0	1	0	0	
Caesar	1	1	0	1	1	1	
Calpurnia	0	1	0	0	0	0	
Cleopatra	1	0	0	0	0	0	
mercy	1	0	1	1	1	1	
worser	1	0	1	1	1	0	
...							

- The result is a binary term-document incidence matrix, as shown above. The rows of the term-document incidence matrix are made up of terms (words), and the columns of the term-document incidence matrix are made up of documents (plays in this case).
- In a term-document incidence matrix an element (t, d) is 1 if the play (the document) in column d contains the word (term) in the row t , and is 0 otherwise.
- We can see that the document "Julius Caesar" does not contain the term "Cleopatra" thus the entry "0" in the term-document incidence matrix.

Commonly Used Terms in Text Mining

- **The Boolean Retrieval Model** – This is one of the basic models used to retrieve information from the term-document incidence matrix.
 - Uses term-document incidence matrix so it avoids linear search of all documents.
 - Uses bitwise (binary) logical operations which are the fastest possible.

Illustration:

To answer the query Brutus AND Caesar AND NOT Calpurnia, we take the vectors for Brutus, Caesar and Calpurnia, complement (negate) the binary vector for Calpurnia, and then do a bitwise AND:

$$110100 \text{ AND } 110111 \text{ AND } 101111 = 100100$$

- So this query is contained in the first and the fourth document (see previous slides).

Commonly Used Terms in Text Mining

- **The Boolean Retrieval Model** – This is one of the basic models used to retrieve information from the term-document incidence matrix.
 - Uses term-document incidence matrix so it avoids linear search of all documents.
 - Uses bitwise (binary) logical operations which are the fastest possible.

Illustration:

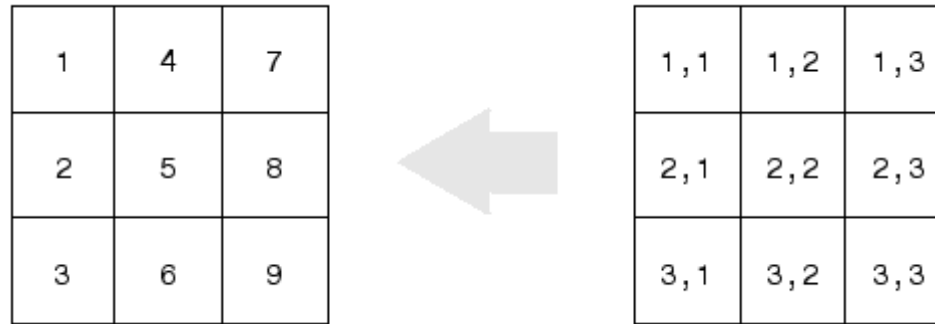
- Suppose we have $N = 1$ million documents (or a collection - units we use to build an information retrieval system from).
- Suppose each document is about $W=1000$ words long (2–3 book pages). Assuming an average of 6 bytes per word including spaces and punctuation, then this is a document collection of about 6 GB in size, or about $M = 500,000$ distinct terms.
- This gives a matrix of zeros and ones of size $M*N=5 \cdot 10^{11}$, which is too big to keep and analyze.
- What can we do about this?
- Note that the DTM-matrix is extremely sparse (only few nonzero entries). The maximum number of ones is $W*N=10^9$, which is much smaller than $M*N=5 \cdot 10^{11}$ (total number of 1's and 0's). Or at least 99.8% in $M*N$ will be 0's.
- What is the best way to deal with sparse matrices?
 - Indexing!

Commonly Used Terms in Text Mining

- **Indexed Notation of a Matrix** – TDM is sparse, more memory economical (thus faster) to keep just the nonzero entries.

Illustration:

- Better representation is to record only the position of 1's. Example of a mapping from subscript (right) to linear indexes notation (left) for a 3-by-3 matrix.

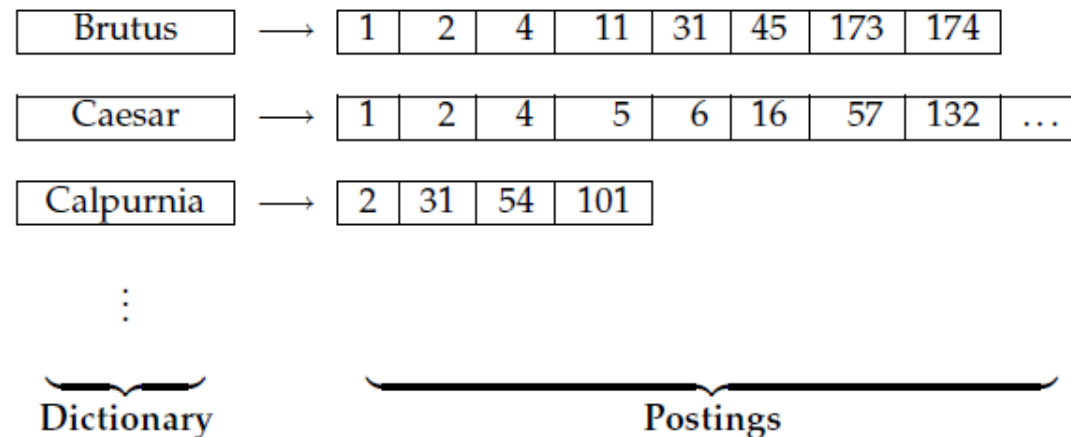


- In the case of having a large sparse matrix the advantage of the index notation is in the fact that only the indices of the nonzero elements need to be kept.
- How many indices do you need to keep track of the nonzero entries in the matrix A?

$$A = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}$$

Commonly Used Terms in Text Mining

- **Dictionary (Vocabulary or Lexicon)** – This keeps all the unique terms. For each term we keep a list (vector) in which document this term occurs, such as Brutus appears in documents 1, 2, 4....
- **Postings (Posting Lists)** – Relates dictionary (unique) terms with document ID (the way we numbered documents). Lists are typically sorted alphabetically and each POSTINGS LIST (matrix) is sorted by document ID. This is very useful but there are also alternatives to doing this. An example of this is illustrated below. The dictionary has been sorted alphabetically and each postings list is sorted by document ID.



Commonly Used Terms in Text Mining

Building an INDEX by sorting and grouping. There are four major steps

1. Collect the documents to be indexed:
 - Friends, Romans, countrymen. So let it be with Caesar . . .
2. Tokenize the text, turning each document into a list of tokens:
 - Friends Romans countrymen So . . .
3. Do linguistic preprocessing, producing a list of normalized tokens, which are the indexing terms: friend roman countryman so . . .
4. Index the documents that each term occurs in by creating an index, consisting of a dictionary and postings as the index built on next slide.

Then queries are processed by

- Locating terms in the dictionary
- Retrieve its postings (vectors of document indices)
- Intersect the two postings lists

Query optimization is the process of selecting how to organize the work of answering a query so that the least total amount of work needs to be done by the system.

Doc 1

I did enact Julius Caesar: I was killed
i' the Capitol; Brutus killed me.

Doc 2

So let it be with Caesar. The noble Brutus
hath told you Caesar was ambitious:

term	docID		term	docID					
I	1		ambitious	2		term	doc. freq.	→	postings lists
did	1		be	2		ambitious	1	→	2
enact	1		brutus	1		be	1	→	2
julius	1		brutus	2		brutus	2	→	1 → 2
caesar	1		capitol	1		capitol	1	→	1
I	1		caesar	1		caesar	2	→	1 → 2
was	1		caesar	2		did	1	→	1
killed	1		caesar	2		enact	1	→	1
i'	1		did	1		hath	1	→	2
the	1		enact	1		I	1	→	1
capitol	1		hath	1		i'	1	→	1
brutus	1		I	1		it	1	→	2
killed	1		I	1		julius	1	→	1
me	1	⇒	i'	1	⇒	killed	1	→	1
so	2		it	2		let	1	→	2
let	2		julius	1		me	1	→	1
it	2		killed	1		noble	1	→	2
be	2		killed	1		so	1	→	2
with	2		let	2		the	2	→	1 → 2
caesar	2		me	1		told	1	→	2
the	2		noble	2		you	1	→	2
noble	2		so	2		was	2	→	1 → 2
brutus	2		the	1		with	1	→	2
hath	2		the	2					
told	2		told	2					
you	2		you	2					
caesar	2		was	1					
was	2		was	2					
ambitious	2		with	2					

Commonly Used Terms in Text Mining

- **Ranked Retrieval Model** – Contrasting the Boolean model are ranked retrieval models such as the vector space model. These models use free text queries (more complicated math SVD), and the system decides which documents best satisfy the queries.

Commonly Used Terms in Text Mining

Typical goals in processing text are:

- Searching and matching
- Extracting information
- Grouping information
- Building QA system

Common tools for processing digital text:

- String manipulation tools.
 - Most programming languages contain libraries for doing basic operations like concatenation, splitting, substring search, and a variety of methods for comparing two strings.
- Tokens and tokenization
 - The first step after extracting content from a file is almost always to break the content up into small, usable chunks of text, called tokens. In English this is best done by occurrence of whitespace such as spaces and line breaks.
- Part of speech assignment
 - Identifying whether a word is a noun, verb, or adjective. Using part of speech can help determine what the important keywords are in a document. Commonly used to enhance the quality of results in digital text processing. There are many readily available, trainable part of speech taggers available in the open source community.
- Stemming
 - Stemming is the process of reducing a word to a root, or simpler form, which isn't necessarily a word on its own. An example is searching for the word bank to retrieve an documents on banking.
- Sentence detection
 - Computing sentence boundaries can help reduce erroneous phrase matches as well as provide a means to identify structural relationships between words and phrases and sentences to other sentences.

Commonly Used Terms in Text Mining

Text processing is also a hard problem on a small scale. Few tasks that come up time and time again in text applications:

- Search the text based on user input and return the relevant passage (a paragraph for example).
- Split the passage into sentences.
- Extract “interesting” things from the text, like the names of people and/or their email addresses for example.

Regular string matching

These operations (regular and fuzzy string matching) most commonly refer to strings from the indexed text documents.

- Regular String Matching
 - Please refer to lecture notes for the common packages for regular string manipulations in R.
 - Some low level regular string manipulations include:
 - Splitting a String
 - Counting the Number of Characters in a String
 - Detecting a Pattern in a String; Detecting the Presence of a Substring

Fuzzy string matching

- Fuzzy String Matching
 - Strings comparison process – similarity between strings, such as in a spell checker.
 - Algorithms reduce to linear algebra concepts such as similarity between vectors (dot product and cosine similarity).
 - Three measures:
 - Character-overlap measures (Jaccard measure, Jaro-Winkler etc.)
 - Edit-distance measures (the minimum operations needed to transform one string into another)
 - N-gram edit distance (similar to the previous, transforms q-characters instead of letters).
- All of these are already implemented in R, you just need to be familiar with them.

Using Trie to Find Fuzzy Prefix Matches

- Used in finding prefix string matches.
 - Predictive text or auto complete dictionary, such as is found in a spell check or on a mobile telephone.
- The trie, or prefix tree, stores strings by decomposing them into characters.
- Retrieving words is done by traversing the tree structure to the node that represents the prefix being queried.
- A search ends if a node has a non null value.

