

Unstructured Data

CS102
Spring 2020

Data Tools and Techniques

- **Basic Data Manipulation and Analysis**

Performing well-defined computations or asking well-defined questions (“queries”)

- **Data Mining**

Looking for patterns in data

Over
“unstructured” data
(text, images, video)

- **Machine Learning**

Using data to make inferences or predictions

- **Data Visualization**

Graphical depiction of data

- **Data Collection and Preparation**

Analyzing Text

- Much of the world's data is in the form of free-text
- Different sizes of text elements
 - Small - tweets
 - Medium - emails, product reviews
 - Large - documents
 - Very large - books
- Different sizes of text corpuses
 - Ultimate text corpus: the web

Types of Text Processing

1. Search “Text analytics”, “Text mining”

- For specific words, phrases, patterns
- Find common patterns, phrase completions

2. Understand “Natural language processing/understanding”

- Parts of speech, parse trees
- Entity recognition - people, places, organizations
- Disambiguation - “ford”, “jaguar”
- Sentiment analysis - happy, sad, angry, ...

Applications of Text Analytics / NLP

- Search engines
- Spam classification
- News feed management
- Document summarization
- Language translation
- Speech-to-text
- Many, many more ...

“Information Retrieval”

- Long-standing subfield of computer science
 - Since the 1950's!
- Goal: retrieve information (typically text) that matches a search (typically words or phrases)
 1. Text corpus usually too large to scan
 2. Return most relevant answers first
- Some techniques and tools
 - Inverted indices
 - N-grams
 - TF-IDF
 - Regular expressions

Seem
familiar?

Inverted Indices

Goal: Find all documents in a corpus containing a specific word (e.g., web search)

- Impossible to scan all documents for the word
- Scan the corpus in advance to build an “inverted index”; keep it up-to-date as content changes

Word	Document IDs
the	1, 2, 3, 5, 7, 8, 9, ...
quick	2, 5, 7, 13, ...
brown	1, 2, 7, 16, 17, ...
fox	7, 9, 11, 13, ...
jumped	3, 4, 6, 7, ...
...	...

N-grams

Goal: Find all documents in a corpus containing a specific *phrase* (e.g., web search)

- Cannot create inverted index for all phrases
- Create indexes of “n-grams” - phrases of length n

3-gram	Document IDs
the quick brown	2, 7, 15, 23, ...
fox jumped over	7, 23, 35 ...
the lazy dog	23, 35, 87, 88, ...
over the lazy	5, 9, 23, 88, ...
...	...

- Note: index on previous slide was 1-grams
- Also helps with longer phrases

N-grams

N-gram frequency counts - typically over entire text corpus or language, useful for:

- Search auto-completion, spelling correction
- Summarization
- Speech recognition, machine translation

3-gram	Frequency
quick brown fox	1500
quick brown dog	800
...	...
the lazy dog	2500
the lazy person	50
...	...

TF-IDF

Goal: estimate how important a given keyword is to a document (e.g., for keyword search ranking)

1. Use word frequency in the document:

“tree” occurs very frequently

“cheetah” occurs much less frequently

➤ “tree” is an important word, “cheetah” less so

2. Incorporate word specialization:

TF-IDF = Term Frequency / Document Frequency

(IDF = “Inverse Document Frequency”)

TF-IDF (slightly simplified)

TF-IDF = Term Frequency / Document Frequency

Term frequency # occurrences in document

Document frequency # documents with at least one occurrence

- Ex: “tree” occurs 50 times in document, occurs in 1000 documents in corpus: TF-IDF = .05
- Ex: “cheetah” occurs 10 times in document, occurs in 30 documents in corpus: TF-IDF = .33

Words with very high document and term frequency are typically “stopwords” (*and, at, but, if, that, the, ...*), often excluded from text analysis

Regular Expressions

Goal: Search for text that matches a certain pattern
Sample of regular expression constructs:

regular expression	matches
.	any single character
$exp ?$	zero or one instances of exp
$exp *$	zero or more instances of exp
$exp ^+$	one or more instances of exp
$(exp_1 exp_2)$	either exp_1 or exp_2

May be composed for arbitrarily complex expressions

Regular Expressions

regular expression	matches
.	any single character
$exp ?$	zero or one instances of exp
$exp *$	zero or more instances of exp
$exp ^+$	one or more instances of exp
$exp_1 exp_2$	either exp_1 or exp_2

((Prof | Dr | Ms)‘. ’)? Widom

(.+)@(.+)‘. ’(edu | com)

http://(.*)stanford(.*)

Precision and Recall

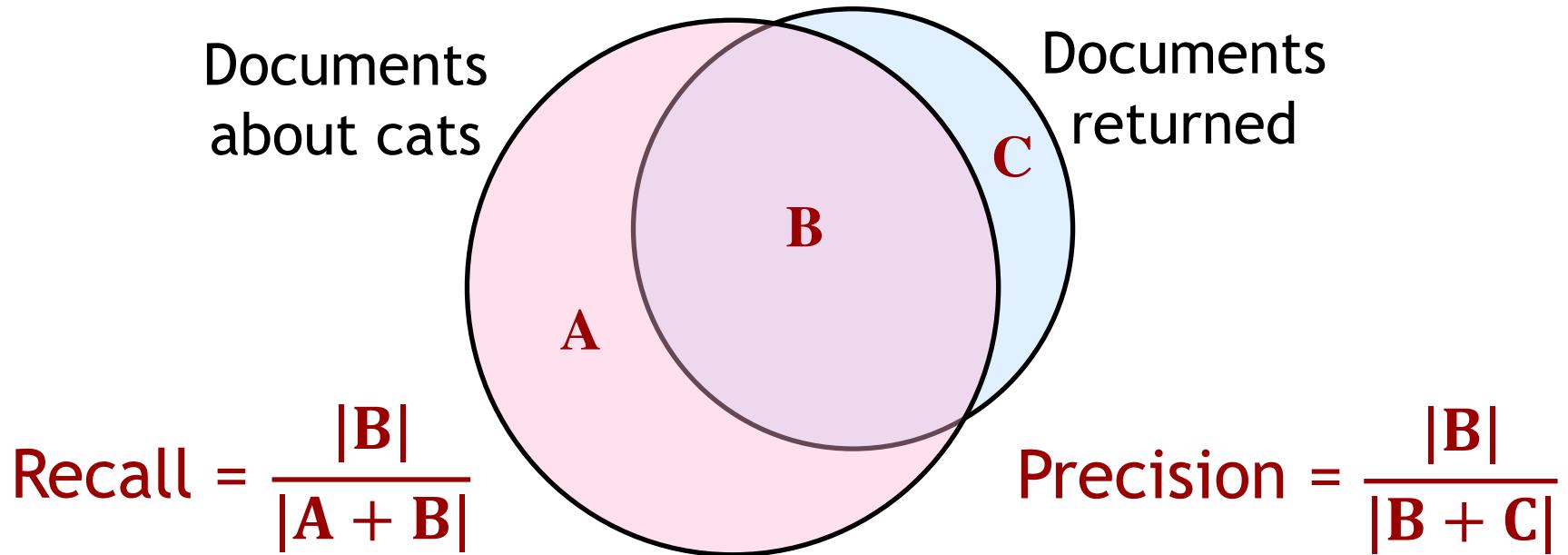
Measures of correctness or quality of results:

- Basic data analysis → expect the correct answer
- Data mining → results based on support and confidence thresholds
- Machine learning → accuracy of predicted values or labels (average error, percent correct)
- Text and image analysis → precision and recall

Precision and Recall

- Example
 - Search for documents about cats
 - Assume “ground truth” - every document is either about cats or it’s not
- *Precision*: Fraction of returned documents that are actually about cats
- *Recall*: Fraction of documents about cats in the corpus that are returned by the search

Precision and Recall



- Return one correct document → 100% precision
- Return all documents → 100% recall
- Challenge is to achieve high values for both
- More complex when results are ranked

Hands-On Text Analysis

- Dataset
 - Wine descriptions
(200 in class, 10K in assignment)
- Python packages
 - Regular expressions (re)
 - Natural Language Toolkit (nltk)
- Text-scanning style only - no indices

Image Analysis

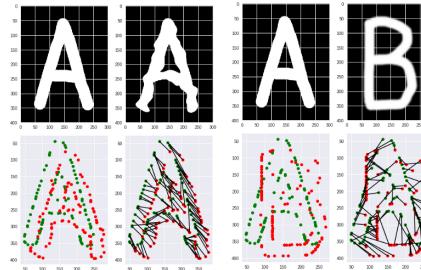
- Find images in a corpus based on specification
E.g., color, shape, object, focus
- Rank images based on specification
- Classify or label images (machine learning)

Image Analysis: Progression

First approach
➤ human labeling



Second approach
➤ shape recognition



Third approach
➤ machine learning



Video Analysis

- Video = series of images + audio stream
- Video search can take different forms
 - Like image search (colors, shapes, objects) but with added time element
 - Like text search over audio stream
 - Combine the two

Find all videos with people in more than 50% of the frames and occurrences of both “Stanford” and “MIT” in the audio track

Hands-on Image Analysis

- Dataset
 - 206 country flags
- Simple image-by-image color analysis
 - All images are a grid of pixels - e.g., 200 x 200
 - Each pixel is one color
 - Color represented as RGB [0-255,0-255,0-255]
- Python package
 - Python Imaging Library (PIL)
 - Newer version “Pillow”

