CS 175, Project in Artificial Intelligence

Introduction

Padhraic Smyth
Department of Computer Science
Bren School of Information and Computer Sciences
University of California, Irvine
Today’s Lecture

• Discuss class schedule and organization

• Applications of text analysis in the real world

• Ideas for possible class projects

• Assignment 1
Course Description for CS 175 (Winter 2016)

Students in this project class will work in small teams to develop artificial intelligence and machine learning algorithms and apply them to a range of different problems related to text analysis.

These problems will include, for example, document classification and clustering, sentiment analysis, information extraction, word prediction, and text synthesis.

Projects will make use of real-world data from sources such as Twitter, Wikipedia, news articles, product and movie reviews, emails, the US patent database, and more.
Class Organization

• Class Website:  [www.ics.uci.edu/~smyth/courses/cs175](http://www.ics.uci.edu/~smyth/courses/cs175)
  – This is where to find assignments, links to software, project guidelines, etc

• Piazza Website:
  – [https://piazza.com/uci/winter2016/compsci175/home](https://piazza.com/uci/winter2016/compsci175/home)
  – Use this to post questions related to assignments, projects, etc
  – Piazza is where we will post announcements, answers to questions, etc

• Waitlisted students
  – Waitlisted students will be added if/when registered students drop the class
Class Organization (continued)

• Textbook and Reading Materials
  – No official textbook
    • NLTK (Natural Language Toolkit): Python platform for text analysis
    • Online text is a very useful reference
    • Note that there is a print version of this book (from 2009), based on Python 2.7 and a little out of date – please use the online text as the primary reference
  – Class Website contains pointers to background reading that we will refer to in lectures and that will be useful for project work

• Discussion Section
  – The TAs will hold discussion sections in weeks 1 to 3 (while you are working on Assignments 1 and 2) – after that we will decide if discussions are needed.
Teaching Assistants and Office Hours

- Two TAs
  - Kevin Bache
  - Jihyun Park

- Kevin’s weekly office hours
  - 2pm to 4pm Mondays, DBH 4059

- Jihyun’s weekly office hours
  - 3pm to 5pm on Wednesdays, DBH 4013

- My weekly office hours
  - Tuesdays: 10 to 11:30, DBH 4216

- Recommend that you take advantage of these office hours
Discussion Sections

• 1 to 1:50pm, Friday, ICS 174

• This week’s discussion topics
  – Anaconda installation
  – PyCharm setup / common keyboard shortcuts
  – overview of Python syntax
  – numerical Python coding best practices
  – questions about the homework

• Attendance at discussions is not required....but is strongly recommended
Contacting TAs and Professor: use Piazza

- Use Piazza for all offline questions related to the class
  - Assignments, lectures, projects, data sets, ideas, etc

- TAs and professor will monitor and answer questions
  - Students should also feel free to also answer questions
  - If you wish you can use “private mode” to ask questions that only the TAs and/or Professor will see

- TA and Professor Office hours
  - Opportunity to discuss assignments/projects in more depth

- Use direct email only if other options do not work for some reason
Academic Honesty

Academic honesty is taken seriously in this class. **Failure to adhere to the class policy can result in a student receiving a failing grade in the class.**

For individual assignments you are allowed to discuss the assignments verbally with other class members, but under no circumstances can you look at or copy anyone else's written solutions or code. All problem solutions and code submitted must be material you have personally written during this quarter, except for any standard library or utility functions.

For team projects all reports submitted must be written by members of your project team. Code generated for class projects can be a combination of code written by team members and publicly-available code. You should clearly indicate which code was written by members of the team (e.g., with comments in appropriate places) and which code was not written by the team (e.g., by listing the libraries you used in an Appendix in your report).

It is the responsibility of each student to be familiar with **UCI's academic honesty policies**
Content of this Class

• Lectures and Assignments
  – Learn general principles of automated text analysis
  – Emphasis on machine learning for text, e.g., classifying a document
  – Combination of lectures, assignments (two), and background reading

• Team Project: build a prototype software system for text analysis (weeks 4 to 10)
  – Propose an idea and plans for your class project
  – Do background research and reading
  – Develop ideas, implement algorithms, make use of libraries and packages
  – Conduct experiments with real data sets
  – Test and evaluate your system in a systematic manner
  – Communicate your results (presentations and reports)
Projects

- Teams of 3 students – you will form your own teams
  - Project grading will be partly team-based and partly on individual contributions
  - Note that Assignments 1 and 2 are *not* team-based – these will be worked on and submitted individually

- Each team will propose its own project
  - Suggestions for multiple different projects will be provided
  - Extensive use of libraries (in addition to writing some of your own code)

- Projects will be graded based on
  - Initial proposal
  - Intermediate and final reports
  - In-class presentation

[We will discuss all of this in more detail in future lectures]
<table>
<thead>
<tr>
<th>Week</th>
<th>Monday</th>
<th>Wednesday</th>
<th>Deadlines</th>
</tr>
</thead>
<tbody>
<tr>
<td>January 4th</td>
<td>Introduction and course outline</td>
<td>Basic concepts in automated text analysis</td>
<td></td>
</tr>
<tr>
<td>January 11th</td>
<td>Review of classification algorithms</td>
<td>Text classification methods</td>
<td>Assignment 1 (Monday, 10pm)</td>
</tr>
<tr>
<td>January 18th</td>
<td>No class (University holiday)</td>
<td>Ideas for class projects</td>
<td>Assignment 2</td>
</tr>
<tr>
<td>January 25th</td>
<td>Text classification methods</td>
<td>Discussion of project proposals</td>
<td>Project Proposal</td>
</tr>
<tr>
<td>February 1st</td>
<td>Project proposal feedback</td>
<td>Unsupervised learning for text</td>
<td></td>
</tr>
<tr>
<td>February 8th</td>
<td>Document clustering methods</td>
<td>Topic modeling for text</td>
<td></td>
</tr>
<tr>
<td>February 15th</td>
<td>No class (University holiday)</td>
<td>Office hours (no lecture)</td>
<td></td>
</tr>
<tr>
<td>February 29th</td>
<td>Student progress presentations</td>
<td>Student progress presentations</td>
<td>Progress report</td>
</tr>
<tr>
<td>March 1st</td>
<td>Office hours (no lecture)</td>
<td>Evaluation methods for projects</td>
<td></td>
</tr>
<tr>
<td>March 8th</td>
<td>Student final project presentations</td>
<td>Student final project presentations</td>
<td>Upload slides before class</td>
</tr>
<tr>
<td>March 15th</td>
<td></td>
<td></td>
<td>Final Report (Monday)</td>
</tr>
</tbody>
</table>
Software Environment for Assignments and Projects

• Python
  – Python will be the primary language we will use in this class
  – Assume that all students have a working knowledge of Python 3

• Packages and Libraries
  – We will make extensive use of additional packages and libraries in Python
  – NLTK: Natural Language Toolkit
  – Scikit-learn: machine learning library
  – Scientific computing/graphs/etc: matplotlib, numpy, scipy, etc

  We recommend you download and install the Anaconda package: it contains pretty much everything you need for this class (NLTK, numpy, scikit-learn, etc)

• Integrated Development Environment (IDE)
  – You are free to use whatever IDE you prefer (e.g., PyCharm)
  – In class I will use the Spyder IDE for demos (comes with Anaconda)
Screenshot of the Spyder IDE
Natural Language Toolkit

NLTK is a leading platform for building Python programs to work with human language data. It provides easy-to-use interfaces to over 50 corpora and lexical resources such as WordNet, along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning, and an active discussion forum.

Thanks to a hands-on guide introducing programming fundamentals alongside topics in computational linguistics, NLTK is suitable for linguists, engineers, students, educators, researchers, and industry users alike. NLTK is available for Windows, Mac OS X, and Linux. Best of all, NLTK is a free, open source, community-driven project.

NLTK has been called “a wonderful tool for teaching, and working in, computational linguistics using Python,” and “an amazing library to play with natural language.”

Natural Language Processing with Python provides a practical introduction to programming for language processing. Written by the creators of NLTK, it guides the reader through the fundamentals of writing Python programs, working with corpora, categorizing text, analyzing linguistic structure, and more. The book is being updated for Python 3 and NLTK 3. (The original Python 2 version is still available at http://nltk.org/book_1ed.)

Some simple things you can do with NLTK

Tokenize and tag some text:

```python
>>> import nltk
>>> sentence = """"At eight o'clock on Thursday morning
... Arthur didn't feel very good."""
>>> tokens = nltk.word_tokenize(sentence)
>>> tokens
```
Natural Language Processing with Python

– Analyzing Text with the Natural Language Toolkit

Steven Bird, Ewan Klein, and Edward Loper

The NLTK book is currently being updated for Python 3 and NLTK 3. This is work in progress; chapters that still need to be updated are indicated. The first edition of the book, published by O'Reilly, is available at http://nltk.org/book_web/. A second edition of the book is anticipated in early 2016.

0. Preface
1. Language Processing and Python
2. Accessing Text Corpora and Lexical Resources
3. Processing Raw Text
4. Writing Structured Programs
5. Categorizing and Tagging Words (minor fixes still required)
6. Learning to Classify Text
7. Extracting Information from Text
8. Analyzing Sentence Structure
9. Building Feature Based Grammars
10. Analyzing the Meaning of Sentences (minor fixes still required)
11. Managing Linguistic Data (minor fixes still required)
12. Afterword. Facing the Language Challenge

Bibliography
Term Index

This book is made available under the terms of the Creative Commons Attribution Noncommercial No-Derivative-Works 3.0 US License. Please post any questions about the materials to the nltk-users mailing list. Please report any errors on the issue tracker.
scikit-learn
Machine Learning in Python

- Simple and efficient tools for data mining and data analysis
- Accessible to everybody, and reusable in various contexts
- Built on NumPy, SciPy, and matplotlib
- Open source, commercially usable - BSD license

Classification
Identifying to which set of categories a new observation belong to.

- Applications: Spam detection, Image recognition.
- Algorithms: SVM, nearest neighbors, random forest, ...  — Examples

Regression
Predicting a continuous value for a new example.

- Applications: Drug response, Stock prices.
- Algorithms: SVR, ridge regression, Lasso, ...  — Examples

Clustering
Automatic grouping of similar objects into sets.

- Applications: Customer segmentation, Grouping experiment outcomes.
- Algorithms: k-Means, spectral clustering, mean-shift, ...  — Examples

Dimensionality reduction
Reducing the number of random variables to consider.

- Applications: Visualization, Increased efficiency.
- Algorithms: PCA, Isomap, non-negative matrix factorization.  — Examples

Model selection
Comparing, validating and choosing parameters and models.

- Goal: Improved accuracy via parameter tuning
- Modules: grid search, cross validation, metrics.  — Examples

Preprocessing
Feature extraction and normalization.

- Application: Transforming input data such as text for use with machine learning algorithms.
- Modules: pre-processing, feature extraction.  — Examples

News
- On going development: What's new (changelog)

Community
- Questions? See stackoverflow # scikit-learn
- Mailing list: scikit-learn-

Who uses scikit-learn?
Assignment 1

Available on the class Web page

Due Monday January 11\textsuperscript{th}, by 10pm

Outline

– Read Chapter 1 of the online NLTK book
– Install Anaconda/NLTK/…
– Write simple functions in Python for text analysis
  • Extract words from text strings ("tokenization")
  • Remove “stop words”
  • Create “bag of words” representation
  • Plot distributions of word frequency
  • Learn to use NLTK functions
– Submit your code as a single python script via EEE
Basic Concepts and Terminology

• **Document:**
  – A book, a news article, a report, a Web page, an email, a tweet, etc
  – Usually contains both text and metadata.
  – Examples of metadata: author name(s), date, where published, etc

  Note that the definition of a document is flexible
  e.g., a book could be a single document, or ..... each section of a book could be considered a “document”

• **Corpus:** a collection of documents
  – e.g., all news articles from the New York Times since 1990
  – e.g., all Wikipedia Web pages
Basic Concepts and Terminology

• In its raw format a document is one very long string, e.g.,
  ‘Chapter 1: The Beginning. In the beginning, life was tough!.........’

• Tokens are “words” extracted from the text, e.g.,
  – Tokens = {'chapter', '1', 'the', 'beginning', 'in', 'the', 'beginning', 'life', 'was',...}
  – Punctuation and white spaces are usually ignored

• Vocabulary
  – The unique set of tokens (or words) in a document or corpus, e.g.,
    Vocabulary = {'chapter', '1', 'the', 'beginning', 'in', 'life', 'was', 'tough',...}

• Bag of Words
  – The number of times each word (unique token) appears, e.g.,
    Bag of Words = { ['chapter', 1], ['1', 1], ['the' 2], ['beginning', 2], ...}
Automated Text Analysis

• Availability of large amounts of text in digital form has led to a huge increase in automated text analysis techniques and applications

• Examples of large text data sets
  – Web pages
  – Emails and text messages
  – Blogs and microblogs
  – Product reviews
  – Search queries
  – Scientific articles, e.g., 20 million Medline articles
  – News articles about companies and products
  – Collections of digitized books and historical documents
  – ...and many more....
APPLICATIONS OF TEXT ANALYSIS
Who is interested in analyzing such data?

- **Web companies**
  - Google, Facebook, Twitter, Microsoft, Yahoo!, and many more

- **Ecommerce**
  - Automated analysis of product reviews + customer text such as emails, search queries, etc
  - eBay, Amazon, plus many “regular” companies that have a Web presence

- **Financial industry**
  - Automated tracking of news and online blogs about companies and products

- **Law enforcement and intelligence agencies**
  - Text mining of vast amounts of emails, blogs, etc

- **Medical researchers**
  - Automated analysis/summarization of publications on diseases, genes, drugs, etc

- **Social scientists and humanities researchers**
  - Studying history and social science through analysis of large text collections
A Revolution in the Technology of Data

Graphic from Ray Kurzweil, singularity.com
What Happens in an Internet Minute?

- 204 million emails sent
- 2.5 million search queries issued
- 350,000 new tweets

500 million 30-day active users

The Friendship graph

500M users each connect to an average of 130 other users = ~ 60 Billion Edges

Over 30 billion pieces of content shared every month

Over 3 billion photos uploaded each month

Graphics from Lars Backstrom, ESWC 2011
Google Search query = “beer”, over time
Google Search query = “beer”, over time
The Google Books Project

- Google has digitized over 8 million books
  - Books from 40 university libraries around the world
  - 4.5 million in English, rest in other languages. 6% of all books ever published.
  - 500 billion words
  - Spans multiple centuries since 1500’s

- Reading the books manually is impossible
  - Reading only English-language entries since 2000, at the pace of 200 words/minute, with no sleep/food interruptions, would take 80 years!
The Google N-grams Corpus

- **N-grams**
  - 1-gram: a string of characters with no spaces, e.g., “dog”, “NFL”, “3.142”
    - In effect 1grams are “words”
  - N-gram: a sequence of N 1-grams, e.g.,
    - Bigram = sequence of pairs of 1-grams, e.g., “big dog”, “New York”
    - Trigram = sequence of triples of 1 grams, e.g., “New York City”

- Google N-grams corpus allows us to see how individual words (1-grams) and N-grams have changed in usage over the years

- **Usage frequency =**
  
  \[
  \frac{\text{(number of instances of an N-gram per year)}}{\text{(total number of words in the collection that year)}}
  \]
Detecting of Linguistic Usage and Change over Time

Political Analysis: Detection of Possible Censorship

Suppression of mentions of political figures in Russian texts

Differences in mentions of Tiananmen after 1976 and 1989 incidents, in Chinese (blue) and English (green)

Applications of Text Analysis

• Document classification
  – Spam email classification: email text -> {spam, not spam}
  – Sentiment classification: product review text -> {positive, negative}
  – Web page classification: Web page text -> {sports, finance, entertainment, ....}

• Machine translation
  – Automated translation of text from one language to another
  – e.g., for Web pages, for mobile phones

• Web search
  – Ranking of Web pages based on matching queries with content

• Web advertising
  – Matching search queries and Web page content to online advertisements
Each ? represents an “ad slot”

In a fraction of a second, algorithms predict which ads you are most likely to click on (from 1000’s of ads)
The ads that are most likely to lead to a click are selected and displayed.
Applications of Text Analysis (continued)

• Personalization
  – Creating customized Web pages, newspapers, interfaces for individuals

• Autocompletion
  – Predicting words to improve user interfaces on smartphones

• Corpus exploration
  – Developing visualization and search tools for researchers and lawyers exploring millions of patents

• Information extraction
  – Extracting mentions of entities (people, places, companies, ...) from text
    • e.g., “Mr. Obama traveled to London to meet Mr. Cameron
  – Extraction of relations
    • e.g., travel_to(Obama, London), meet(Obama, Cameron)
Jaguar Cars

Car manufacturer

Jaguar Cars is a brand of Jaguar Land Rover, a British multinational car manufacturer headquartered in Whitley, Coventry, England, owned by the Indian company Tata Motors since 2008. [Wikipedia]

Customer service: 1 (800) 452-4827

Headquarters: Coventry, United Kingdom

Founded: September 4, 1922, Blackpool, United Kingdom

Parent organization: Jaguar Land Rover, Tata Motors

Founders: William Walmsley, William Lyons

Latest models

- [2016 Jaguar XF](#)
- [2016 Jaguar XJ](#)
- [2016 Jaguar F-TYPE](#)
- [2017 Jaguar XE](#)
- [2015 Jaguar XKR](#)

People also search for

- BMW
- Mercedes
- Audi
- Aston Martin
- Land Rover

Feedback
Jacksonville Jaguars

3rd in AFC South

Tomorrow, 10:00 AM on CBS
NRG Stadium, Houston, Texas

Jacksonville Jaguars vs. Houston Texans

Schedule and scores

All times are in Pacific Time

Jacksonville Jaguars, Official Site of the Jacksonville Jaguars

www.jaguars.com

The official team site with scores, news items, game schedule, and roster.
Schedule - News - Roster - The Jungle

Jacksonville Jaguars Football News, Schedule, Roster, Stats

www.sbnation.com/nfl/teams/jacksonville-jaguars

Record: 5-10 - After sustaining a foot injury late in last week’s blowout of the Titans, the first pick of the 2014 NFL Draft is unlikely to play in the Texans’ regular...

Jacksonville Jaguars - Wikipedia, the free encyclopedia


The Jacksonville Jaguars are an American professional football franchise based in
Applications of Text Analysis (continued)

• Automated Dialog Agents
  – Bots that can carry on a conversation/dialog with a human via text
  – E.g., applications to answering customer inquiries (e.g., for troubleshooting)

• Text Summarization
  – Automated summaries of text documents
    • In applications such as law, medicine, etc

• Automated Essay Grading
  – E.g., for SAT, AP, GRE exams, or for online courses

• Natural Language Generation (NLG) or Text Synthesis
  – Applications to automated generation of news stories
  – Automatically generating replies to customer emails
Application: Text Synthesis

## Airline Delays

<table>
<thead>
<tr>
<th>airline</th>
<th>airline_short</th>
<th>recent_flights</th>
<th>month_current</th>
</tr>
</thead>
<tbody>
<tr>
<td>American Airlines Inc.</td>
<td>American</td>
<td>354</td>
<td>4</td>
</tr>
<tr>
<td>JetBlue Airways</td>
<td>JetBlue</td>
<td>233</td>
<td>4</td>
</tr>
<tr>
<td>Delta Air Lines Inc.</td>
<td>Delta</td>
<td>446</td>
<td>4</td>
</tr>
<tr>
<td>ExpressJet Airlines Inc.</td>
<td>ExpressJet</td>
<td>242</td>
<td>4</td>
</tr>
<tr>
<td>Frontier Airlines Inc.</td>
<td>Frontier</td>
<td>19</td>
<td>4</td>
</tr>
<tr>
<td>Envoy Air</td>
<td>Envoy</td>
<td>344</td>
<td>4</td>
</tr>
<tr>
<td>SkyWest Airlines Inc.</td>
<td>SkyWest</td>
<td>25</td>
<td>4</td>
</tr>
<tr>
<td>United Air Lines Inc.</td>
<td>United</td>
<td>75</td>
<td>4</td>
</tr>
<tr>
<td>US Airways Inc.</td>
<td>US Air</td>
<td>276</td>
<td>4</td>
</tr>
<tr>
<td>Southwest Airlines Co.</td>
<td>Southwest</td>
<td>844</td>
<td>4</td>
</tr>
<tr>
<td>British Airways</td>
<td>BA</td>
<td>340</td>
<td>4</td>
</tr>
<tr>
<td>Alaskas Airlines</td>
<td>Alaskas</td>
<td>240</td>
<td>4</td>
</tr>
</tbody>
</table>

### American Airlines

American Airlines Inc. ranked 8th in on-time performance at Raleigh-Durham International Airport (RDU) in April with 22.9% of flights arriving at least 15 minutes late, up from No. 9 last month. American saw a slight improvement compared to the prior month's performance in which 25.6% of flights were delayed. The airline also cancelled two flights into Raleigh. Frontier Airlines Inc. ranked first among the 10 airlines flying into RDU in April, while United Air Lines Inc. finished with the worst on-time performance at the airport.

For the last six months, American ranks 7th among airlines flying into RDU with 21.3% of flights delayed. American's delayed flight percentage over that period has been as high as 25.6% in March and as low as 15.8% in February. Delta holds the top spot over that period at 10.8%, while United once again ranks last at 29.5%.

American's 81 delays out of 354 flights in April totaled 73.1 hours, down 1.7% from the previous month. The U.S. Department of Transportation divider delays hit 1,650.

Graphic from: https://automatedinsights.com/examples/
## Application: Text Synthesis

### Credit Card Account Summary

<table>
<thead>
<tr>
<th>Month</th>
<th>Start Period</th>
<th>End Period</th>
<th>Merchandise</th>
<th>Merchandise 12M Avg</th>
<th>Merchandise 12M Mktg</th>
<th>Rest 12M Avg</th>
<th>Rest 12M Mktg</th>
<th>Rest 12M Mktg %</th>
<th>Rest Month</th>
<th>Rest 12M Mktg %</th>
<th>Rest Month</th>
<th>Rest 12M Mktg %</th>
<th>Rest Month</th>
</tr>
</thead>
<tbody>
<tr>
<td>December</td>
<td>12/6/2013</td>
<td>1/5/2014</td>
<td>893.22</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>January</td>
<td>1/6/2014</td>
<td>2/5/2014</td>
<td>104.9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>February</td>
<td>2/6/2014</td>
<td>3/5/2014</td>
<td>90.17</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>March</td>
<td>3/6/2014</td>
<td>4/5/2014</td>
<td>332.74</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>April</td>
<td>4/6/2014</td>
<td>5/5/2014</td>
<td>500.23</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>May</td>
<td>5/6/2014</td>
<td>6/5/2014</td>
<td>325.48</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>June</td>
<td>6/6/2014</td>
<td>7/5/2014</td>
<td>283.92</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>July</td>
<td>7/6/2014</td>
<td>8/5/2014</td>
<td>142.43</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>August</td>
<td>8/6/2014</td>
<td>9/5/2014</td>
<td>300.36</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>September</td>
<td>9/6/2014</td>
<td>10/5/2014</td>
<td>575.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>October</td>
<td>10/6/2014</td>
<td>11/5/2014</td>
<td>192.39</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### May Account Summary

#### Account Summary

For the period between 5/6/15 and 6/5/15, you accumulated $1,944 worth of charges. A payment of $1,800.00 was processed during the month. A credit of $31.99 was issued to your account. Your current balance is $432.25 and a minimum payment of $35 is due on 7/2/15.

#### Breakdown

- Restaurants were where you spent the most money this month, accounting for $572.33. Two categories set 12-month highs this period, restaurants and gasoline. Five categories exceeded their 12-month average with travel and entertainment showing the largest increase at 83%.

#### Rewards

Way to go! You earned $38.63 in Rewards Cash this month.

Graphic from: https://automatedinsights.com/examples/
IDEAS FOR POSSIBLE CLASS PROJECTS
Possible Projects: Document Classification

Original document

\[ \text{(NBA, 7), (Lakers, 3), basket (2),...} \]

Document features (e.g., a “bag of words”)

Label = basketball

Class label
Possible Project: Document Classification

Original document

[ (NBA, 7), (Lakers, 3), basket (2)....]  
Document features  
(e.g., a “bag of words”)  
Label = basketball  
Class label

Tokenization,  
Stemming,  
Part of Speech Tagging, etc  
(Wednesday’s lecture)

Classification model, e.g.,  
- naïve Bayes  
- logistic regression  
- neural network  
(Assignment 2 and next week’s lectures)

Assignment 2 and next week’s lectures
Possible Projects using Document Classification

- Use Wikipedia pages and categories as training data and build a classification algorithm that can classify news articles.

- Build a sentiment classification algorithm that can predict if a product or movie review is positive or negative.

- Develop an algorithm that can automatically classify emails into an appropriate folder (e.g., for Gmail).

- Conduct a systematic study of how document length, sample size, or other factors affect the accuracy of document classifiers on standard data sets.
Possible Projects using Document Clustering

• Clustering of Documents:
  – Takes a set of documents (each represented as a bag of words) and automatically clusters/groups the documents

• Build an algorithm that can cluster news articles so that articles about the same news story end up in the same cluster
  – Note that to do this well may require extraction of information about people and places and time from the articles

• Develop a tool to download an individual’s email history (e.g., from Gmail) and to group emails into clusters on similar topics
Other Ideas for Projects

(this is just a small partial list...there are many other possibilities!)

• Information Extraction:
  – Extract names of products and companies from news articles
  – Extract names of actors and directors from movie reviews

• Change in Language over Time:
  – Develop an algorithm that can automatically identify key topics in US Patent data and track how these topics change over time
Examples of large text data sets that could be used for projects

Text from 4 million Wikipedia articles

PubMed: 20 million abstracts of biomedical research papers

Enron emails: 250,000 company emails

Twitter data: large streams of tweets via Twitter API
The DBpedia Data Set (2015-04)

we are happy to announce the release of DBpedia 2015-04 (also known as: 2015 A). The new release is based on updated Wikipedia dumps dating from February/March 2015 and features an enlarged DBpedia ontology with more infobox to ontology mappings, leading to richer and cleaner data.


The English version of the DBpedia knowledge base currently describes 5.9M things out of which 4.3M resources have abstracts, 452K geo coordinates and 1.46M depictions. In total, 4 million resources are classified in a consistent ontology and consists of 2,06M persons, 682K places (including 455K populated places), 378K creative works (including 92K music albums, 90K films and 17K video games), 188K organizations (including 51K companies and 33K educational institutions), 278K species and 5K diseases. The total number of resources in English DBpedia is 15.3M that, besides the 5.9M resources, includes 1.2M skos concepts (categories), 6.83M redirect pages, 250K disambiguation pages and 1.13M intermediate nodes.

We provide localized versions of DBpedia in 128 languages. All these versions together describe 38.3 million things, out of which 23.8 million are localized descriptions of things that also exist in the English version of DBpedia. The full DBpedia data set features 38 million labels and abstracts in 128 different languages, 25.2 million links to images and 29.8 million links to external web pages; 80.9 million links to Wikipedia categories, and 41.2 million links to YAGO categories. DBpedia is connected with other Linked Datasets by around 50 million RDF links.

In addition we provide DBpedia datasets for Wikimedia Commons and Wikidata.
Assignment 1

Available on the class Web page

Due Monday January 11\textsuperscript{th}, by 10pm

Outline

– Read Chapter 1 of the online NLTK book
– Install Anaconda/NLTK/...
– Write simple functions in Python for text analysis
  • Extract words from text strings (“tokenization”)
  • Remove “stop words”
  • Create “bag of words” representation
  • Plot distributions of word frequency
  • Learn to use NLTK functions
– Submit your code as a single python script via EEE