Instructor: Ludovic Rheault  
Email: ludovic.rheault@utoronto.ca  
Office Hours: Online (see Quercus page)
Software

Class exercises and demonstrations will be done using the Python programming language. Python is the most popular programming language in the world, and is especially useful for the analysis of textual data. The course begins with an introduction to the language.

Class examples will rely upon Python 3.8 (any version of Python 3.x should be good to go). In-class examples will be provided from the Jupyter notebook, a user-friendly environment for interactive computing.

Python is freely available on all operating systems. It should be installed by default on Mac computers. Students can reproduce exercises and replicate examples on their personal computers during class.

Requirements

This course may be of interest to graduate students using either qualitative or quantitative methods (or both). Although there are no formal requirements for the course, it will involve some advanced concepts in programming and statistics. A background in statistical analysis and/or computing would be useful, at the level of the PhD introductory course POL 2504. The pedagogical approach is tailored to students who may not have had an extended training in mathematics or computing as undergraduate students (as is often the case in the social sciences).

Marking Scheme

Given the intensive pace of a summer semester, the course relies on three assignments (plus a participation mark). The first two assignments are problems sets. The last assignment is a critical evaluation of written work from the field of text as data (5 pages max).

- Written Assignment #1 30% Due May 19, 2021
- Written Assignment #2 30% Due June 14, 2021
- Written Assignment #3 30% Due at the end of term (June 21)
- Participation 10 %

Readings

The readings for this course comprise a collection of chapters from the following set of seminal texts in the field. The readings recommended for each class are helpful to supplement the lecture notes that will be made available to students. All of these books are accessible for free online, either from the authors’ websites or electronically through the UofT Library.

  - An accessible introduction to natural language processing in Python. The book is available online for free.

  - A key reference that covers most of the topics discussed in this course, and more. Online versions are available.
  ○ Another useful reference for exploring some of the topics in more depth. Some chapters are available online for free.
  ○ An older reference that nonetheless covers key basic concepts for this course. The book is available electronically through the UofT Library.
  ○ A useful reference on the particular topic of machine learning. The book is available electronically through the UofT Library.
  ○ This new resource is a short book that covers many of the topics we study in this course. It is available electronically through the UofT Library.

**Evaluations**

The course uses two evaluation formats to help students develop different skills related to scientific research.

**Problem Sets**

The first two written assignments are problem sets designed to evaluate students’ ability to put the methods learned into practice. They involve practicing various types of textual analysis using Python and answering short factual questions about the models and their interpretation.

There is no better way to improve one’s skills than practice. Therefore, those exercises are not only useful as evaluations, but also as a way for students to gain concrete expertise with the subject-matter. Assignments are done individually. They are submitted on Quercus at the due date.

**Critical Review**

The last assignment consists of reading and discussing published work relying on text as data. The list of admissible papers will be posted on Quercus. The goal is to demonstrate that, at the end of the term, the graduate student is able to engage with the literature in a meaningful way, understand the methods and judge their appropriateness, and identify the strengths and limitations of published studies involving automated textual analysis.

This exercise will be valuable in preparing students to read and write their own papers using text as data.
## Class Schedule: Summary

<table>
<thead>
<tr>
<th>Date</th>
<th>Topic</th>
<th>Evaluation</th>
</tr>
</thead>
<tbody>
<tr>
<td>May 3</td>
<td>Computers and language &amp; introduction to Python</td>
<td></td>
</tr>
<tr>
<td>May 5</td>
<td>Introduction to Python (continued)</td>
<td></td>
</tr>
<tr>
<td>May 10</td>
<td>Statistics for textual data I</td>
<td></td>
</tr>
<tr>
<td>May 12</td>
<td>Statistics for textual data II</td>
<td></td>
</tr>
<tr>
<td>May 17</td>
<td>Concepts in computational linguistics</td>
<td></td>
</tr>
<tr>
<td>May 19</td>
<td>Lexicons and dictionaries</td>
<td>Assignment 1 due</td>
</tr>
<tr>
<td>May 24</td>
<td>[Victoria Day - No Classes]</td>
<td></td>
</tr>
<tr>
<td>May 26</td>
<td>Meaning and word embeddings</td>
<td></td>
</tr>
<tr>
<td>May 31</td>
<td>Introduction to machine learning</td>
<td></td>
</tr>
<tr>
<td>June 2</td>
<td>Supervised learning I</td>
<td></td>
</tr>
<tr>
<td>June 7</td>
<td>Supervised learning II</td>
<td></td>
</tr>
<tr>
<td>June 9</td>
<td>Unsupervised learning I</td>
<td></td>
</tr>
<tr>
<td>June 14</td>
<td>Unsupervised learning II</td>
<td>Assignment 2 due</td>
</tr>
<tr>
<td>June 21</td>
<td>[Final assessment period]</td>
<td>Assignment 3 due</td>
</tr>
</tbody>
</table>

*Note: Topics by date are for information only. The schedule above (and the detailed structure in the following pages) may be adjusted during the term due to unforeseen circumstances or to improve the pedagogical benefits to students.*
Class Schedule: Detailed

Topic 1: Computers and Text

May 3: Computers and Language; Introduction to Python

1. Brief history of automated textual analysis.
2. Examples of recent applications.
3. Introduction to Python 3 (beginning).

May 5: Introduction to Python (Continued)

1. Introduction to Python 3 (continued).
2. Data types, lists and dictionaries.
3. Input/Output.
4. Functions and conditional statements.
5. Encoding text.
6. Processing textual data in Python.

Readings

- Hovy (2020), Ch. 1.
- Bird, Klein, and Loper (2009), Ch. 2–4.

Other Useful References

- Aggarwal and Zhai (2012b).
- Manning and Schütze (1999), Ch. 1.
- McKinney (2013), Ch. 1.
- Downey, Elkner, and Meyers (2002), Ch. 1–2.
- D’Orazio et al. (2014).
- Jockers (2014).
- Krippendorff (2013), Ch. 4.
- Grimmer and Stewart (2013).
- Gentzkow, Kelly, and Taddy (2019).
- Benoit (2019).
- Watch a 45-minute introductory video on Python.
Topic 2: Statistics for Textual Data

May 10: Statistics for Textual Data I
1. Document retrieval and indexing.
2. Tokenization, sentence splitting.
3. Word counts and word distributions.
4. Vectorization.
5. Visualization techniques.

May 12: Statistics for Textual Data II
1. Term-frequency/inverse document frequency (tf–idf) weighting.
2. Word co-occurrences/collocations.
3. Comparing texts.
4. Statistical properties of texts.

Readings
- Manning, Raghavan, and Schütze (2009), Ch. 1–2.
- Hovy (2020), Ch. 2–4.
- Manning and Schütze (1999), Ch. 5–6.

Other Useful References
- Bird, Klein, and Loper (2009), Ch. 2–4.
- Jiang (2012).
- Nenkova and McKeown (2012).
- Python Online Documentation.

Examples of Applications
- Laver and Garry (2000).
- Gentzkow and Shapiro (2010).
- Proksch and Slapin (2010).
- Black et al. (2011).
- Däubler et al. (2012).
- Acton and Potts (2014).
- Spirling (2016).
- Blaxill and Beelen (2016).
- Benoit, Munger, and Spirling (2019).
Topic 3: Linguistics and Natural Language Processing

May 17: Concepts in computational linguistics

1. Overview of linguistic theory.
2. Unigrams, bi-grams and n-grams.
3. Part-of-speech tagging.
4. Stemming and lemmatization.
5. Grammar parsing.

May 19: Lexicons and dictionaries

1. Creating and using word lexicons (dictionaries).
2. Summarizing text properties.

May 26: Meaning and word embeddings

1. Meaning representation and latent semantic analysis.
2. Word embeddings.
3. Word similarities and word relations.

Readings

• Bird, Klein, and Loper (2009), Ch. 5.
• Hovy (2020), Ch. 5.
• Jurafsky and Martin (2020), Ch. 20.

Other Usefulness References

• Manning, Raghavan, and Schütze (2009), Ch. 6.
• Miller et al. (1990).
• Turney and Pantel (2010).
• Mikolov et al. (2013).
• Manning et al. (2014).
• Landauer, Foltz, and Laham (1998).
• Python Online Documentation.

Examples of Applications

• Tausczik and Pennebaker (2010).
• Bollen, Mao, and Zeng (2011).
• Bollen, Mao, and Pepe (2011).
• Golder and Macy (2011).
• Michel et al. (2011).
• Young and Soroka (2012).
• Jensen et al. (2012).
• Coviello et al. (2014).
• Gentzkow, Shapiro, and Taddy (2016).
• Rheault et al. (2016).
• Vosoughi et al. (2018).
• Martin and McCrain (2019).
• Gennaro and Ash (2021).

## Topic 4: Machine Learning

### May 31: Introduction to Machine Learning

2. Annotating texts and intercoder reliability.
3. Development, training and testing.
4. An introductory example: sentiment analysis.

### June 2: Supervised Learning I

1. Features and classes.
2. "Bag of words" approach.
3. Feature selection.
5. Nearest Neighbor classifiers.
6. Multi-class problems.

### June 7: Supervised Learning II

1. Evaluating classifiers.
2. Accuracy measures.
3. Ridge regression.
5. Applications in Python.

### June 9: Unsupervised Learning I

1. Unsupervised learning.
3. Clustering analysis.
4. Principal component analysis.

### June 14: Unsupervised Learning II

1. Latent Dirichlet Allocation (LDA).
2. Correlated and dynamic topic models.
3. Non-Negative Matrix Factorization.
Readings

- Hastie, Tibshirani, and Friedman (2009), Ch. 2, 6–7, 12.
- Hovy (2020), Ch. 6–7.

Other Useful References

- Manning, Raghavan, and Schütze (2009), Ch. 15.
- Blei, Ng, and Jordan (2003).
- Blei and Lafferty (2006a).
- Blei and Lafferty (2006b).
- Blei (2012).
- Bird, Klein, and Loper (2009), Ch. 6.
- He and Garcia (2009).
- Steyvers and Griffiths (2011).
- Aggarwal and Zhai (2012a).
- Richert and Coelho (2013).
- Lantz (2013).
- James et al. (2013).
- scikit-learn for Python: Online Documentation.

Examples of Applications

- Mosteller and Wallace (1964).
- Yu, Kaufmann, and Diermeier (2008).
- Hopkins and King (2010).
- Grimmer (2010).
- Diermeier et al. (2012).
- Hirst et al. (2014).
- Roberts et al. (2014).
- D’Orazio et al. (2014).
- Harris (2015).
- Reich et al. (2015).
- Roberts, Stewart, and Airoldi (2016).
- Tingley (2017).
- Barberá et al. (2019).
References


10


He, Haibo, and Edwardo A. Garcia. 2009. “Learning from Imbalanced Data.” IEEE Transactions on Knowledge and Data Engineering 21(9): 1263–1284.


James, Gareth, Daniela Witten, Trevor Hastie, and Robert Tibshirani. 2013. An Introduction to Statistical Learning with Applications in R. New York: Springer.


