PS9594A: Computational Text Analysis

Department of Political Science – Western University, Winter 2026
Monday 1:00pm-4:00pm, SSC 7200

Instructor: Dr. Sebastián Vallejo Vera (sebastian.vallejo@uwo.ca)
Office hours: Monday 12:00pm-1:00pm or by appointment... mostly by appointment (SSC 7221)

Course description

One of the most abundant sources of data available to social and political scientists today is text. Recent advances in Natural Language Processing (NLP) have spearheaded a text-as-data revolution, which has led social scientists to seek out new means of analyzing text data at scale. In this course, we will learn the intuition behind—and how to implement—different computational methods to process, analyze, and classify text. The course will cover Bag-of-Words (BoW) approaches, unsupervised methods, supervised and semi-supervised methods, and LLMs that use text as data, as well as how we can interpret the results obtained from applying these methods.

Course objectives

In this course, students will:

- learn how to use text as data;
- understand the potential and the limitations of using text as data;
- get training on how to use computational text analysis techniques;
- learn how to obtain and process text data.

Acknowledgments

The organization of the first part of this course (Weeks 1 - 5) and the format of the assignments are borrowed from Christopher Barrie's excellent course on "Computational Text Analysis", a syllabus from the prolific Tiago Ventura, and Grimmer, Roberts, and Stewart's excellent book, "Text as data: A new framework for machine learning and the social sciences". The code used throughout the course is a patchwork of my own code, but my own code borrows heavily from the internet (but that's true for all code). I try my best to give credit to the original authors of the code (when and if possible).

Readings and Slides

The main external platform we will use in this class is Perusall (www.perusall.com). Perusall is a free collaborative annotation tool that allows you to analyze texts collaboratively. All the required texts for the class, the most important supplementary readings, and the class slides are available on Perusall.

You can access Perusall through the Content tab in the OWL page for the course.

Code

All the code for the class will be posted at `www.COMING_SOON.com`.

Course assessment

Students will be assessed as follows:

- **Homework (30%)**: There will be three worksheets (Exercise #N) that will walk you through the implementation of different text analysis techniques. At the end of each worksheets, you will find a set of questions. You should partner up with someone else in your class and go through these together.[1] In the next class, I will pick on a pair at random to answer each one of that worksheet's questions, and walk us through your code. This is not a punitive exercise, but rather a space for collaborative learning. More often then not, the obstacles encountered by one person are also encountered by many other. Furthermore, there are many ways to arrive to the same solution, and being exposed to different frameworks is beneficial to all. All that matters to me is that you **try** and, eventually, learn. (Same goes for those who only have to turn in the assignment).

  The assignments will be graded in accuracy and quality of the programming style. The following are elements I will be looking at when grading:

  - all code must run;
  - solutions should be readable:
    * Code should be thoroughly commented (I should be able to understand the codes purpose by reading the comment),
    * Coding solutions should be broken up into individual code chunks in Jupyter/R Markdown notebooks, not clumped together into one large code chunk,
    * Each student defined function must contain a doc string explaining what the function does, each input argument, and what the function returns (OPTIONAL);
  - Commentary, responses, and/or solutions should all be written in Markdown and explain sufficiently the outputs.

- **Replication Exercise (30%)**: Replication exercises are adapted from Gary King's work here and here. At its core, replication is the process of re-running a study using the authors' original data and code, and checking whether you can recover the published results (and, when relevant, bringing new data into the conversation). Replicability matters because it is how scientific knowledge becomes credible, cumulative, and actually useful.

  For our purposes, replication is primarily an educational tool. A common saying in science is that you only really learn a method once you use it in your own work. Since we do not really have the time to write three full papers in one semester, we will instead take advantage of published research with openly available datasets and code, and treat the replication as a structured, step-by-step way of learning by doing. Here is what this exercise will look like:

[1] From Barrie: "This is called pair programming and there's a reason we do this. Firstly, coding can be an isolating and difficult thing–it's good to bring a friend along for the ride! Secondly, if there's something you don't know, maybe your partner will. This saves you both time. Thirdly, your partner can check your code as you write it, and vice versa. Again, this means both of you are working together to produce and check something as you go along."

1. **Step 1: Choosing a paper to replicate** By the end of Week 3, you should select one article from the syllabus that you will replicate.
2. **Step 2: Acquiring the data and code** Many journals now have open data and open materials expectations, which means authors will often make their replication files available (commonly on GitHub or the Harvard Dataverse). Your first task is to locate the data and code associated with your chosen article.

   If the article does *not* have replication materials publicly available, you should:
   - Politely contact the authors and request the replication materials; and
   - If you do not receive a response within a reasonable amount of time, select a different article.
3. **Step 3: Presentation** In Week 11, you will present your replication efforts. Your presentation should include:
   - **Introduction**: a short summary of the article and its main claims;
   - **Methods**: a description of the data and empirical strategy used in the article;
   - **Results**: what you were able to replicate (tables, figures, key quantities);
   - **Differences**: any differences between your results and the authors' results;
   - **Replication autopsy**: what worked, what did not, and where things got stuck (this is often the most informative part);
   - **Extension**: if you were writing this paper today, what would you do differently? Where would you innovate?
4. **Step 4: Replication repository and report** By Friday end-of-day of the replication week, you should share your replication materials with me and your classmates. Your replication package should include:
   - A GitHub repository with a clear, well-documented `README`. (A model is available here.)
   - Your presentation as a PDF;
   - The code used in your replication as a reproducible notebook (R Markdown or Jupyter);
   - A short written report (maximum 5 pages; fewer is totally fine) summarizing the replication process, with emphasis on four parts of your presentation: **Results**, **Differences**, **Replication autopsy**, and **Extension**.

In addition to following the requirements above, the replication exercises will also be graded in accuracy and quality of the programming style. For instance, I will be looking at:

- all code must run;
- solutions should be readable:
  - Code should be thoroughly commented (I should be able to understand the codes purpose by reading the comment),
  - Coding solutions should be broken up into individual code chunks in Jupyter/R Markdown notebooks, not clumped together into one large code chunk,
  - Each student defined function must contain a doc string explaining what the function does, each input argument, and what the function returns (OPTIONAL);
- Commentary, responses, and/or solutions should all be written in Markdown and explain sufficiently the outputs.

- **Final Project (40%)**: A 4000-word **max** essay. Further instructions are at the end of the syllabus.

Class Expectations

1. **Always be respectful and mindful of your classmates.**
2. The class starts at 1:00 PM. Please, be on time and awake, or somewhat awake, or faking being awake.
3. I will start the class at 1:00 PM with whoever is in the room. Arriving late? No problem. Just enter discreetly and quietly, take your seat, and we are all good. 1:15 PM is not the time to greet, chat, wave vigorously to your friends in the room. When you do this, you distract those that were on time and you distract me (it is also disrespectful, see point 1).
4. If you are going to be taking notes in your laptop/iPad, close all other tabs that might distract you from the lecture. The secret is to hang to my every word.
5. I cannot make you pay attention and participate. But I can ask you to avoid distracting the rest of the class. Remember: I already know the material. The important part is for you to learn it.
6. If you are going to be watching TikTok during class anyways, at least drop the links to the really funny ones.
7. I do not care if you are wearing pajamas, but please come to class. Worst case scenario, the material presence of your being might allow you to learn through osmosis.

A quick yet important note on cellphones: Our class is 180 minutes long. Most things in life can wait three hours to be resolved/answered/liked/swiped-right/retweeted/watched/poked/high-fived/instagramed/swiped-left/live-streamed. There is no need for you to have your cellphone out and about (yes, I notice when you are in your phone even when you try to hide it under your desk). If, for some reason, you need to have your cellphone out, please let me know before class (you know, as a courtesy).

Children in Class

I applaud all of you who go to school with children! It is difficult to balance academia, work, and family commitments, and I want you to succeed. Here are my policies regarding children in class:

1. All breastfeeding babies are welcome in class as often as needed. If your baby requires your attention, you can step outside and tend to them.
2. Non-nursing babies and older children are welcome as well. As a parent of two school-age children, I understand that babysitters fall through, partners have conflicting schedules, children get sick, and other issues like a global pandemic arise that leave parents with few other options. If you child requires your attention, you can step outside and tend to them.
3. All students are expected to join me in creating a welcoming environment that is respectful of your classmates who bring children to class.

I understand that sleep deprivation and exhaustion are among the most difficult aspects of parenting young children. The struggle of balancing school, work, childcare, and high inflation is tiring, and I will do my best to accommodate any such issues while maintaining the same high expectations for all students enrolled in the class. Please do not hesitate to contact me with any questions or concerns.

Late Work Policy

Legally defined adults are late with things ALL THE TIME (myself included).

That said, deadlines serve their purpose. They can create an external structure to help you plan your workload and prevent everything from piling up on you. Furthermore, we live (and learn) in a community. When I take longer to submit a paper revision to a journal, I make the editor's job more complicated. If many of you turn in your work late, it makes planning the material we need to cover more challenging for me. Finally, there are deadlines that are more absolute than others. If the plane closes its doors at 10:00 AM and you arrive at 10:15 AM, there's no earthly power that can reopen them.

In this class, there are two types of deadlines: 1) the fatal ones, which are deadlines that cannot be postponed, and 2) the non-fatal ones, which are suggestions and planning guides (rather than arbitrary and punitive dates meant to generate anxiety). The fatal deadlines are those that are immovable for practical reasons. For example, any work submitted to me after the deadline I must submit grades will not be considered because, well, I will have already submitted grades. Similarly, due to their nature, the Final Exam must be submitted within the agreed-upon times.

The non-fatal deadlines are more flexible. While I **strongly recommend** that you keep up with the class schedule, I also acknowledge that things happen (e.g., global pandemics, climate crises, life events). Since I don't want your assignments to pile up and I also don't want you to feel like you must disappear if you submit something late, for the rest of the deadlines (e.g., homework), I have adopted a more "liberal" policy with extensions. The only thing I ask is that you proactively communicate with me to find solutions for any delays that will allow you to successfully complete the course. Note that, even if there is no penalty for late submission, if you submit an assignment late, you might also get late feedback, which might lead to knowledge gaps during lectures.

Finally, remember that I also have a life outside the classroom, and it is partly scheduled around important course dates. If you submit an assignment late, there's a good chance it will take me longer to return it corrected.

Course Structure

IMPORTANT NOTE: I have only listed the required readings for the course. The exercises (and additional optional readings) will be provided throughout the semester.

Week #1 (January 5): Course Introduction / Why (Computational) Text Analysis?

Topics: Review of syllabus and class organization. Introduction to computational text analysis and natural language processing (NLP).

READINGS:

1. Grimmer, Roberts, and Stewart - Ch. 2.
2. Wilkerson, J., & Casas, A. (2017). Large-scale computerized text analysis in political science: Opportunities and challenges. Annual Review of Political Science, 20, 529-544;
3. Macanovic, A. (2022). Text mining for social science–The state and the future of computational text analysis in sociology. Social Science Research, 108, 102784;
4. Barberá, P., & Rivero, G. (2015). Understanding the political representativeness of Twitter users. Social Science Computer Review, 33(6), 712-729;
5. Michalopoulos, S., & Xue, M. M. (2021). Folklore. The Quarterly Journal of Economics, 136(4), 1993-2046.

Week #2 (January 12): Tokenization and Word Frequency

Topics: What is a Bag of Words? What are tokens? Why should we care about tokens?

READINGS:

1. Grimmer, Roberts, and Stewart - Ch. 5;
2. Ban, P., Fouirnaies, A., Hall, A. B., & Snyder, J. M. (2019). How newspapers reveal political power. Political Science Research and Methods, 7(4), 661-678;
3. Michel, J.B., et al. 2011. "Quantitative Analysis of Culture Using Millions of Digitized Books." Science 331 (6014): 176–82. `https://doi.org/10.1126/science.1199644`;
4. Bollen, J., et al. (2021). Historical language records reveal a surge of cognitive distortions in recent decades. Proceedings of the National Academy of Sciences, 118(30), e2102061118.

Week #3 (January 19): Dictionary-Based Techniques

Topics: What are dictionaries? Why/when are they useful? What are their limitations?

READINGS:

1. Grimmer, Roberts, and Stewart - Ch. 15-16;

2. Young, L., and Soroka, S. (2012). Affective news: The automated coding of sentiment in political texts. Political Communication, 29(2), 205-231;

3. Martins, M. D. J. D., & Baumard, N. (2020). The rise of prosociality in fiction preceded democratic revolutions in Early Modern Europe. Proceedings of the National Academy of Sciences, 117(46), 28684-28691;

4. Ventura, T., Munger, K., McCabe, K., & Chang, K. C. (2021). Connective effervescence and streaming chat during political debates. Journal of Quantitative Description: Digital Media, 1.

Week #4 (January 26): Natural Language, Complexity, and Similarity

Topics: How do we evaluate complexity in text? Why should we care about complexity in text? How do we evaluate similarity in text? Why is this useful?

READINGS:

1. Grimmer, Roberts, and Stewart - Ch. 6 and Ch. 7;

2. Spirling, A. (2016). Democratization and linguistic complexity: The effect of franchise extension on parliamentary discourse, 1832–1915. The Journal of Politics, 78(1), 120-136.

3. Urman, A., Makhortykh, M., & Ulloa, R. (2022). The matter of chance: Auditing web search results related to the 2020 US presidential primary elections across six search engines. Social science computer review, 40(5), 1323-1339;

4. Schoonvelde, M., Brosius, A., Schumacher, G., & Bakker, B. N. (2019). Liberals lecture, conservatives communicate: Analyzing complexity and ideology in 381,609 political speeches. PloS one, 14(2), e0208450.

Week #5 (February 2): Scaling Techniques (Unsupervised Learning I)

Topics: What is unsupervised learning? What are scaling models and what can they tell us?

READINGS:

1. Grimmer, Roberts, and Stewart - Ch. 12-13;

2. Slapin, J. B., & Proksch, S. O. (2008). A scaling model for estimating time-series party positions from texts. American Journal of Political Science, 52(3), 705-722.

3. Denny, M. J., & Spirling, A. (2018). Text preprocessing for unsupervised learning: Why it matters, when it misleads, and what to do about it. Political Analysis, 26(2), 168-189.

Week #6 (February 9): Topic Modeling and Clustering (Unsupervised Learning II)

Topics: What is topic modeling and what can it tell us?

READINGS:

1. Grimmer, Roberts, and Stewart - Ch. 12-13;

2. Roberts, M. E., et al. (2014). Structural topic models for open-ended survey responses. American journal of political science, 58(4), 1064-1082.

3. Motolinia, L. (2021). Electoral accountability and particularistic legislation: evidence from an electoral reform in Mexico. American Political Science Review, 115(1), 97-113.

(February 16): Spring reading week. Enjoy the break!

Week #7 (February 23): A Primer on Supervised Learning

Topics: What is supervised learning? We will study the framework to train supervised models, and when to use them. We will learn how Support Vector Machine (SVM) and Bidirectional Long-Short Term Memory (Bi-LSTM) models work.

READINGS:

1. Grimmer, Roberts, and Stewart - Ch. 17-20;

2. Siegel, A. A., et al. (2021). Trumping hate on Twitter? Online hate speech in the 2016 US election campaign and its aftermath. Quarterly Journal of Political Science, 16(1), 71-104.

3. Barberá, P., et al. (2021). Automated text classification of news articles: A practical guide. Political Analysis, 29(1), 19-42.

4. Laver, M., Benoit, K., & Garry, J. (2003). Extracting policy positions from political texts using words as data. American political science review, 97(2), 311-331.

5. Benoit, K., et al. (2016). Crowd-sourced text analysis: Reproducible and agile production of political data. American Political Science Review, 110(2), 278-295.

Week #8 (March 2): Introduction to Deep Learning and Word Embeddings

Topics: How can we capture the meaning of words? Using Deep Learning models to represent text.

READINGS:

1. Grimmer, Roberts, and Stewart - Ch. 8;

2. Lin, G., & Lucas, C. (2023). An Introduction to Neural Networks for the Social Sciences.

3. Meyer, D. (2016). How exactly does word2vec work?. Uoregon. Edu, Brocade. Com, 1-18;

4. Jay Alammar, "The Illustrated Word2vec";

5. Rodriguez, P. L., & Spirling, A. (2022). Word embeddings: What works, what doesn't, and how to tell the difference for applied research. The Journal of Politics, 84(1), 101-115;

6. Kozlowski, A. C., Taddy, M., & Evans, J. A. (2019). The geometry of culture: Analyzing the meanings of class through word embeddings. American Sociological Review, 84(5), 905-949.

Week #9 (March 9): The Transformers Architecture

Topics: We will learn about the Transformers architecture, attention, and the encoder-decoder infrastructure.

READINGS:

1. Jay Alammar. 2018. "The Illustrated Transformer";
2. Vaswani, A., et al. (2017). Attention is all you need. Advances in neural information processing systems, 30;
3. Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805;
4. Timoneda, J.C., and S. Vallejo Vera (2025). BERT, RoBERTa or DeBERTa? Comparing Performance Across Transformer Models in Political Science Text. Journal of Politics.

Week #10 (March 16): Encoder-Only LLMs

Topics: We will take a deep dive into encoder-only LLMs, and all that we can do with them.

READINGS:

1. Taylor, W. L. (1953). "Cloze procedure": A new tool for measuring readability. Journalism quarterly, 30(4), 415-433;
2. Dávila Gordillo, D., J.C. Timoneda, and S. Vallejo Vera. Machines Do See Color: A Guideline to Classify Different Forms of Racist Discourse in Large Corpora. Sociological Methods and Research (Forthcoming).

Week #11 (March 23): Decoder-Only LLMs

Topics: Decoder-only LLMs, also known as generative LLMs, are all the rage now. We will study how they work, what they can do, what are their limitations, and how we can use them in our work (writ large).

READINGS:

1. Lee, K., Paci, S., Park, J., You, H. Y., & Zheng, S. (2024). Applications of GPT in Political Science Research. PS: Political Science and Politics;
2. Gilardi, F., Alizadeh, M., & Kubli, M. (2023). ChatGPT outperforms crowd workers for text-annotation tasks. Proceedings of the National Academy of Sciences, 120(30), e2305016120;
3. Heseltine, M., & Clemm von Hohenberg, B. (2024). Large language models as a substitute for human experts in annotating political text. Research & Politics, 11(1), 20531680241236239;
4. Vallejo Vera, S., & Driggers, H. (2025). LLMs as annotators: the effect of party cues on labelling decisions by large language models. Humanities and Social Sciences Communications, 12(1), 1-11.

5. Walker, C. P., & Timoneda, J. C. (2025). Is ChatGPT conservative or liberal? A novel approach to assess ideological stances and biases in generative LLMs. Political Science Research and Methods, 1-15.

Week #12 (March 30): Presentations: Replications / Catch-up

Topics: Students will be presenting their replications. We will also probably be somewhat behind at this point. We will use the remaining time in this week to catch-up. I will also answer questions related to the final assignment.

READINGS:

1. Time to catch-up on all the readings.

Week #13 (April 6): Presentation: Final Projects / Concluding Remarks

Topics: Students will be presenting their final projects. I will also answer any open questions about the material or the final submission of the final project. We will close the course with some concluding remarks.

READINGS:

1. Time to catch-up on all the readings.

Final Project Instructions

The objective of this activity is for you to write a 4000-word **max** essay on a subject previously approved by me. Think of it as a research note for a journal (see, for example, the Letters at APSR). I will provide a range of data sources to choose from, but you are welcome (encouraged) to suggest your own.

General instructions

Political science is an applied field. For that reason, it is essential that you learn how to conduct a *complete* analysis–from collecting data, to cleaning and analyzing it, to presenting your findings in a way that other people can understand (and reproduce).

Because of this, a substantial portion of your final grade (**40%**) will come from an independent text-as-data project where you apply the concepts and tools we cover throughout the semester to a political science-related question.

Project components

The project has three parts:

- a **2-page project proposal** (to be discussed with, and approved by, me),
- an **in-class presentation**, and
- a **4000-word essay**.

Deadlines and grading breakdown

Requirement	Due	Length	Percentage
Project Proposal	EOD Friday, Week 8	2 pages (max)	5%
Presentation	Week 13	10–15 minutes	10%
Project essay	EOD April 13th	4000 words	25%

Project proposal

The project proposal asks you to sketch out a general **2-page (max)** (single-spaced; 12pt font) plan for your project. The goal here is not to have everything figured out. The goal is to have a *coherent plan* that we can refine, and to make sure you have a feasible project before you sink time into it.

Your proposal should include:

- a high-level statement of the problem you intend to address (or the analysis you aim to generate);
- the data source(s) you intend to use;
- your plan to obtain that data;
- the text-as-data method(s) you plan to use; and
- a definition of what *"success"* means for your project.

In your own words: what would a successful project look like? What is the key result you want to be able to show? What would count as a genuinely surprising finding for you?

Presentation

You will have the opportunity to present your final project in class. As political scientists, your presentation skills are as important as your methods skills. You should prepare a **10–12 minute** presentation that covers:

- your motivation / problem statement,
- data collection,
- methods,
- results, and
- lessons learned and next steps.

Project essay

The essay is the complete description of your project's analysis and results. It should be **4000-words max** (single-spaced; 12pt font). The front page and references do *not* count toward the word limit.

Below I outline what you should aim to cover in each section. Note that your paper should read as a cohesive report (not as a set of disconnected answers to prompts):

- **Introduction**
  - summarize your motivation and briefly discuss prior work related to your question,
  - state your research question, and
  - provide a roadmap of the report.
- **Data and Methods**
  - Where does the data come from?
  - What is the unit of observation?
  - What are the variables of interest?
  - What steps did you take to wrangle the data?
- **Analysis**
  - Describe the methods/tools you explored in your project and how you implemented them.
- **Results**
  - Provide a detailed summary of your results.
  - Present results clearly and concisely.
  - Use visualizations instead of tables whenever possible.
- **Discussion**
  - Re-introduce your main results,
  - state your contributions, and
  - explain where you want to take this project next.

Important note (about literature reviews). There is *no* standalone literature review section for this report. You should absolutely use the literature to motivate your work and situate your question, but you do not need a full section that tries to summarize an entire field. If you want examples of this style, read papers in general-interest journals: many excellent articles do not have long, self-contained literature review sections.

Submission of the final project

The end product should be a **GitHub repository** that contains:

- the raw source data used for the project. If the data are too large for GitHub, talk to me and we will find a solution;
- your proposal;
- a `README` that documents the repository. For *each* file, the README should clearly describe:
  - **inputs** to the file (e.g., raw data; credentials needed to access an API),
  - **what the file does** (major transformations), and
  - **outputs** produced by the file (e.g., a cleaned dataset; a figure).
- the code files that transform the raw data into a form usable to answer your question; and
- your final **4000-word essay** (I will share a template later in the semester).

Of course, **no commits after the deadline** will be considered in the assessment.

Templates for writing

Below are a few templates you can use for writing your report. In my experience, writing data-heavy political science essays in LaTeX is a productivity boost in the long run: it makes formatting predictable, citations painless, and reproducibility easier.

If you want to experiment with LaTeX via Overleaf, you can start with the PNAS template (just remember to switch to a one-column format):

- PNAS template

You can also use the *Journal of Quantitative Description* templates (Word or LaTeX):

- JQD Word
- JQD LaTeX

Finally, you can use templates from Quarto and write your entire project using Quarto (or Markdown) files. This is a perfectly valid workflow for this course.