Democratizing Machine Learning for Interdisciplinary Scholars: Reflections on the NLP+CSS Tutorial Series

Katherine Keith[*]
Williams College
`kak5@williams.edu`

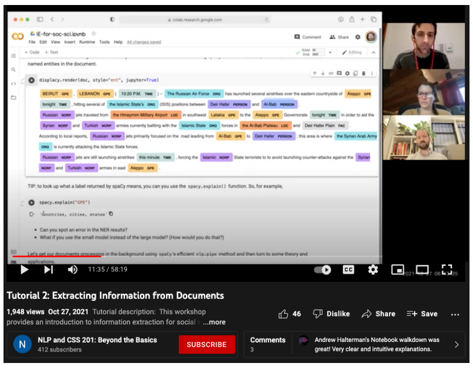
Ian Stewart[*]
Pacific Northwest National Laboratory
`ian.stewart@pnnl.gov`

Abstract

Many scientific fields—including biology, health, education, and the social sciences—use machine learning (ML) to analyze data at an unprecedented scale. However, ML researchers who develop advanced methods rarely provide tutorials showing how to apply these methods. We attempt to democratize the use of ML methods by making them accessible to a broader set of reserachers and practitioners. To that end, we organized a year-long, free, online tutorial series targeted at teaching advanced natural language processing (NLP) methods to computational social science (CSS) scholars. Two organizers worked with fifteen subject matter experts to develop tutorials with hands-on Python code for a range of methods and use cases, from data pre-processing to analyzing temporal language changes. Although live participation was more limited than expected, surveys of participants showed an increase in their perceived knowledge by almost one point on a 7-point Likert scale. Furthermore, participants asked thoughtful questions during tutorials and engaged readily with the content afterwards, as demonstrated by approximately 30K total views of posted tutorial recordings. We distill five principles for democratizing other ML+X tutorials, and we hope that future organizers continue to lower barriers to developing ML skills for researchers of all fields.[1]

1 Introduction

Interest in incorporating machine learning into scientific analyses has exploded in the last two decades. Machine learning (ML)—the process of teaching a machine to predict statistical patterns in data (10)—has gained prominence in biology (9), physics (11), health care (2), and the social sciences (14) inter alia, yielding many successful


TIP: to look up what a label returned by spaCy means, you can you use the `spacy.explain()` function. So, for example,

```
spacy.explain("GPE")
'countries, cities, states'
```

- Can you spot an error in the NER results?
- What if you use the small model instead of the large model? (How would you do that?)

Let's get our documents processing in the background using spaCy's efficient `nlp.pipe` method and then turn to some theory and applications.

11:35 / 58:19
Tutorial 2: Extracting Information from Documents
1,948 views Oct 27, 2021 Tutorial description: This workshop provides an introduction to information extraction for social s ...more
46
Dislike
Share
Save
NLP and CSS 201: Beyond the Basics
412 subscribers
SUBSCRIBE
Comments
3
Andrew Halterman's Notebook walkdown was great! Very clear and intuitive explanations.

Figure 1: Screen shot of recording for *Tutorial 2: Extracting Information from Documents*. As of May 28 2023, this video had 19K views on YouTube.

"ML+X" collaborations. While this potential impact of ML+X is enormous, many researchers unfamiliar with ML methods face barriers to entry, partly because implementing complex methods can be challenging for those without strong mathematical or programming backgrounds (5). Despite the successes of interdisciplinary work such as computational biology, the learning resources for these research areas are often sparse and not well documented, outside of introductory-level material.

Basic ML methods provide a useful starting place, but applied researchers often have more complex problems. For instance, many social scientists want to use ML to develop deep semantic representations of language or to estimate causal effects. Scholars who seek to advance their understanding of ML beyond the basics are often left searching for tutorial-like materials on their own, a difficult and often time-consuming task. On the other hand, well-meaning ML experts may try to share their expertise through media such as blog posts, but they run the risk of "parachuting" into unfamiliar fields with ill-adapted solutions (1; 19). Finally, many formal avenues for sharing knowledge about ML—such as academic conferences—can systematically *exclude* researchers outside of ML via high fees to

[*]Both authors contributed equally to this work.
[1]NLP+CSS Tutorial Website: `https://nlp-css-201-tutorials.github.io/nlp-css-201-tutorials/`

access materials.[2]

We take the position that ML researchers can make their methods more accessible and inclusive to researchers outside the field by creating online instruction explicitly tailored to the fields of subject matter experts. Using the NLP+CSS tutorial series we organized in 2021-2022 as a case study, we argue that these interdisciplinary training sessions should incorporate the following **Principles for Democratizing ML+X Tutorials**:

P.1 Teach machine learning (ML) methods that are relevant and targeted to specific non-ML fields—e.g. biology, health, or the social sciences

P.2 Teach ML methods that are recent and cutting-edge

P.3 Lower start-up costs of programming languages and tooling

P.4 Provide open-source code that is clearly written, in context, and easily adapted to new problems

P.5 Reduce both monetary and time costs for participants

ML+Social Sciences Starting in summer 2021, we put our principles into action and created the *NLP+CSS 201 Online Tutorial Series*. We focused on an applied branch of machine learning to language data—a field called natural language processing (NLP)—aimed at early career researchers in the social sciences. This report reflects on our experience and provides clear takeaways so that others can generalize our NLP + social sciences tutorials to tutorials targeted at other ML+X disciplines.

As we describe in Section 3, we incorporated the principles above into our tutorial series by: (**P.1**&**P.2**) inviting experts in computational social science (CSS) to each lead a tutorial on a cutting edge NLP method; (**P.3**&**P.4**) working with the experts to create a learning experience that is hosted in a self-contained interactive development environment in Python—Google CoLaboratory—and uses real-world social science datasets to provide context for the method; and (**P.5**) hosting our tutorials live via Zoom and posting the recordings on YouTube, while providing all the materials and participation without any monetary costs to participants.

2 Related work

2.1 Interdisciplinary tutorials

Researchers specializing in NLP methods have proposed a variety of interdisciplinary tutorials to address social science questions, which we surveyed before we began planning our tutorial series. However, none satisfied all the principles we listed in Section 1. The tutorials presented at the conferences for the Association for Computational Linguistics (ACL)[3]—one of the premiere venues for NLP research—are on the cutting edge of research (+**P.2**) and often include code (+**P.4**), but the ACL tutorials are also often are geared towards NLP researchers rather than researchers in fields outside of computer science (−**P.1**), contain code that assumes substantial background knowledge (−**P.3**) and cost hundreds of dollars to attend (−**P.5**). Other interdisciplinary conferences such as the International Conference on Computational Social Science (IC2S2)[4] also have tutorials that explain recent NLP methods to computational social scientists (+**P.1**,**P.2**,**P.4**), but often the tutorials are presented with inconsistent formats (−**P.3**) and have high attendance costs (−**P.5**). The Summer Institutes in Computational Social Science (SICSS) (18) provide free (+**P.5**) tutorials on NLP methods for social scientists (+**P.1**) with accompanying code (+**P.3**&**P.4**), but they cover only the basic NLP techniques and not cutting edge methods (−**P.2**), while also limiting their target audience to people already involved with CSS research.[5]

2.2 Online learning

While not without flaws, online learning experiences such as Massive Online Open Courses (MOOCs) have proven useful in higher education when meeting physically is impossible or impractical to due to students' geographic distance (6; 8; 13). Online courses have disrupted traditional education such as in-person college

[2]Among other issues, social science research receives less funding compared to computer science. In 2021, the NSF dispersed $283 million in funding for social sciences, versus $1 billion for computer sciences (from https://www.nsf.gov/about/congress/118/highlights/cu21.jsp, accessed 10 August 2022). This lack of funding can prevent social science researchers from attending ML conferences where new tutorials are presented.

[3]https://www.aclweb.org/portal/acl_sponsored_events

[4]https://iscss.org/ic2s2/conference/

[5]NLP methods include word counting and basic topic modeling: https://sicss.io/curriculum (accessed 11 August 2022).

No.	Tutorial Title	Views
	Fall 2021	
T1	Comparing Word Embedding Models	2732
T2	Extracting Information from Documents	19061
T3	Controlling for Text in Causal Inference with Double Machine Learning	676
T4	Text Analysis with Contextualized Topic Models	1043
T5	BERT for Computational Social Scientists	1402
	Spring 2022	
T6	Moving from Words to Phrases when Doing NLP	864
T7	Analyzing Conversations in Python Using ConvoKit	1093
T8	Preprocessing Social Media Text	1269
T9	Aggregated Classification Pipelines	211
T10	Estimating Causal Effects of Aspects of Language with Noisy Proxies	393
T11	Processing Code-mixed Text	693
T12	Word Embeddings for Descriptive Corpus Analysis	413

Table 1: Tutorial content. Order, title, and number of views of the corresponding recordings on YouTube as of May 28 2023. Full abstracts of each tutorial are provided in the appendix, Table 3.

classes (22), but they may eventually prove most useful as a supplement rather than a replacement to traditional education (21). For one, computer science students have found online learning useful when it incorporates interactive components such as hands-on exercises which may not be possible to execute during a lecture (15; 20). Additionally, while the centralized approach to traditional education can provide useful structure for students new to a domain, the decentralized approach of many online courses can provide room for socialization and creativity in content delivery (23; 24). We intended our tutorial series to fit into the developing paradigm of online education as a decentralized and interactive experience, which would not replace but supplement social science education in machine learning. However, our tutorial series differs from MOOCs in that we limit the time committment for each topic to one hour (+**P.5**) and each tutorial hour is meant to be stand-alone so that researchers can watch only the topics that are relevant to them.

3 Methods for Tutorial Series: Process and Timeline

We describe our process and timeline for creating the tutorial series with the hope that future ML+X tutorial series organizers can copy or build from our experience. Throughout our planning process,

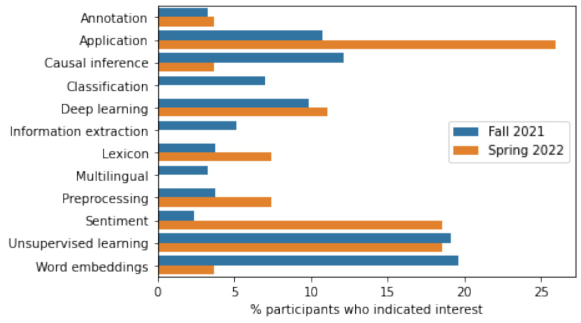


Figure 2: Distribution of interest in NLP methods indicated in initial interest surveys.

we based our decisions on the five principles mentioned earlier (**P.1**-**P.5**). Our tutorial series spanned two semesters: Fall 2021 (August through December) and Spring 2022 (February through May). The tutorial content is summarized in Table 1.

3.1 Interest survey

To select relevant methods for the tutorial series (**P.1**), we distributed a survey via our personal Twitter accounts, via a Google group mailing list that we created at the beginning of the fall 2021 semester, and via topically related mailing lists (e.g. a political methods list-serv). We asked participants to list the methods that they would be most interested in learning about during a tutorial, which we then grouped into categories based on underlying similarities.

The distribution of interest categories is shown in Figure 2. As expected, the responses covered many different NLP applications (17), including data preparation (preprocessing, multilingual), conversion of text to relevant constructs (information extraction, word embeddings, deep learning), and downstream analysis (causal inference, application). Most participants expressed interest in word embeddings, unsupervised learning, and downstream applications of NLP methods, which aligns with the current popularity of such methods.

Lessons learned Since we typically publish in NLP venues, we took a prescriptive approach to choosing the tutorial methods to present, in an attempt to more actively shape the field of computational social science (addressing **P.1**& **P.2**). We used the results of the survey to brainstorm potential topics for each upcoming semester, but did not restrict ourselves to only the most popular methods. While useful, the interest surveys revealed a disconnect between our ideal tutorials, which focused on advanced NLP methods, and the participants' ideal tutorials, e.g. entry-level methods with immediate downstream results. For example, many participants in the Spring 2022 interest survey mentioned sentiment analysis, a well-studied area of NLP (3) that we considered to be more introductory-level and sometimes unreliable (7). This was one source of tension between our expectations and those of the participants, and future tutorial series organizers may want to focus their efforts on highly-requested topics to ensure consistent participation and satisfaction (**P.1**).

3.2 Leader recruitment

Aligning with **P.2**, we recruited other NLP experts who worked on cutting-edge methods to lead each individual tutorial.[6] To ensure **P.3**, we also met with the tutorial hosts to agree on a common format for the programming platform—Google CoLaboratory with Python[7]—and to help them understand the tutorials' objectives. The process involved several meetings: an introduction meeting to scope the tutorial, and at least one planning meeting to review the slides and code to be presented. Normally, this process was guided by a paper or project for which the tutorial leader had code available. For example, the leader of tutorial T4 was able to leverage an extensive code base already tested by her lab.

Lessons learned During the planning process, we were forced to plan the tutorials one at a time due to complicated schedules among the leaders. We spread out the planning meetings during the semester so that the planning meetings would begin roughly two to three weeks before the associated tutorial. We strongly encouraged leaders to provide their code to us at least one week in advance to give us time to review it, but we found this difficult to enforce due to time constraints on the leaders' side (e.g. some leaders had to prioritize other teaching commitments). Future organizers should set up a consistent schedule for contacting leaders in advance and agree with leaders on tutorial code that is relatively new and usable (**P.2**& **P.4**) without presenting an undue burden for the leader, e.g. re-using existing code bases.

[6]We recruited tutorial leaders through our own social networks and through mutual acquaintances. We targeted postdoctoral fellows, early-career professors, and advanced graduate students.

[7]`https://colab.research.google.com/`

3.3 Participant recruitment

Even if we guaranted **P.1**-**P.5** with the content developed, recruiting social science participants was essential to the success of our tutorial series. In September 2021, we set up an official mailing list through Google Groups and advertised it on social media and other methods-related list-servs.[8] The mailing list eventually hosted 396 unique participants. For all tutorials, we set up a RSVP system using Google Forms for participants to sign up, and we provided an RSVP link up to one week before each tutorial. We chose this "walled garden" approach to discourage anti-social activity such as Zoom-bombing which is often made easier by open invitation links (12), and to provide tutorial leaders with a better sense of their participants.

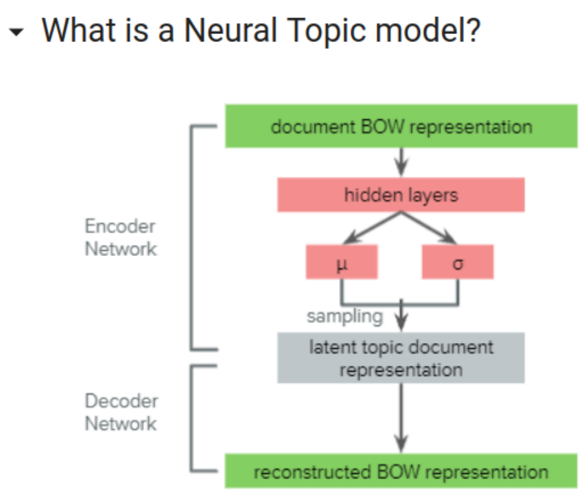
Lesson learned This process revealed significant drop-out: between 10-30% of people who signed up actually attended the tutorial. While the reasons for the drop-out remained unclear, participants may have signed up for the tutorial as a back-up to an existing obligation, under the assumption that the tutorial recording would be available later. Although asynchronous learning can be effective in some cases, the low number of live participants was somewhat discouraging to the tutorial hosts.

3.4 Running the tutorials

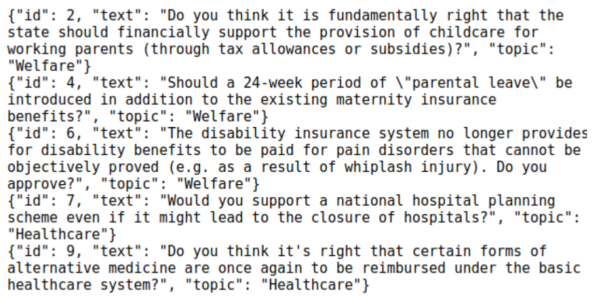
During the tutorials, we wanted to ensure low start-up cost of the programming environment (**P.3**) and well-written code that participants could use immediately after the tutorials (**P.4**). We designed each tutorial to run for slightly under 60 minutes, to account for time required for introductions, transitions, and follow-up questions. The tutorial leader began the session with a presentation to explain the method of interest with minimal math, using worked examples on toy data and examples of prior research that leveraged the method.

After 20-30 minutes of presentation, the tutorial leader switched to showing the code written in a Google CoLaboratory Python notebook (**P.3**), which allows users to run modular blocks of code. The leader would load or generate a simple text dataset, often no more than several hundred documents in size, to illustrate the method's application. Depending on the complexity of the method, the

[8]The Google Group was only accessible to participants with Google Mail accounts, which in retrospect likely discouraged some participants who only use institutional email accounts.

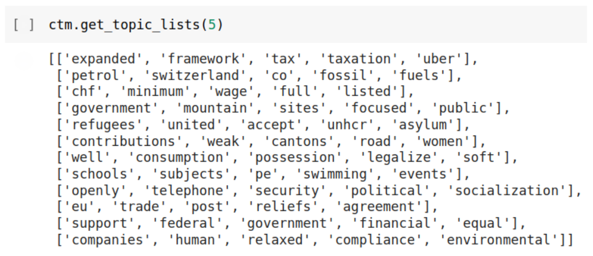

What is a Neural Topic model?
document BOW representation
hidden layers
Encoder Network
μ
σ
sampling
latent topic document representation
Decoder Network
reconstructed BOW representation

(a)



```
{"id": 2, "text": "Do you think it is fundamentally right that the
state should financially support the provision of childcare for
working parents (through tax allowances or subsidies)?", "topic":
"Welfare"}
{"id": 4, "text": "Should a 24-week period of \"parental leave\" be
introduced in addition to the existing maternity insurance
benefits?", "topic": "Welfare"}
{"id": 6, "text": "The disability insurance system no longer provides
for disability benefits to be paid for pain disorders that cannot be
objectively proved (e.g. as a result of whiplash injury). Do you
approve?", "topic": "Welfare"}
{"id": 7, "text": "Would you support a national hospital planning
scheme even if it might lead to the closure of hospitals?", "topic":
"Healthcare"}
{"id": 9, "text": "Do you think it's right that certain forms of
alternative medicine are once again to be reimbursed under the basic
healthcare system?", "topic": "Healthcare"}
```

(b)



```
[ ]  ctm.get_topic_lists(5)

     [['expanded', 'framework', 'tax', 'taxation', 'uber'],
      ['petrol', 'switzerland', 'co', 'fossil', 'fuels'],
      ['chf', 'minimum', 'wage', 'full', 'listed'],
      ['government', 'mountain', 'sites', 'focused', 'public'],
      ['refugees', 'united', 'accept', 'unhcr', 'asylum'],
      ['contributions', 'weak', 'cantons', 'road', 'women'],
      ['well', 'consumption', 'possession', 'legalize', 'soft'],
      ['schools', 'subjects', 'pe', 'swimming', 'events'],
      ['openly', 'telephone', 'security', 'political', 'socialization'],
      ['eu', 'trade', 'post', 'reliefs', 'agreement'],
      ['support', 'federal', 'government', 'financial', 'equal'],
      ['companies', 'human', 'relaxed', 'compliance', 'environmental']]
```

(c)

Figure 3: Excerpts from the tutorial on topic modeling (T4), demonstrating (a) the application of a neural model to (b) text from politicians' interviews, which produces (c) word lists for inductively discovered topics.

leader might start with some basic steps and then show the students increasingly complicated code snippets. In general, the leaders walked the students through separate modules that showed different aspects of the method in question. During the topic modeling session (T4), the leader showed first how to train the topic model, then provided extensive examples of what the topic output looked like and how it should be interpreted (e.g. top words per topic, example documents with high topic probabilities).[9] As a point of comparison, the leader also would often show the output of a simpler "baseline" model to demonstrate the superior performance of the tutorial's more advanced method. We show excerpts from the tutorial notebook on topic modeling in Figure 3, which includes an overview of the topic model, a sample of the text data which relates to politics, and the resulting learned "topics" as lists of words.

Lessons learned To encourage critical thinking, some of the tutorial leaders provided questions or exercises in the Colab notebooks for students to complete at a later time. The leader of the information extraction tutorial (T2) created an exercise for students to parse sentences from news text related to military activity, and then to extract all sentences that described an attack between armies. Some of these exercises posed challenges to participants who lacked experience with the data structures or function calls involved in the code. For future tutorials, leaders should consider simply showing participants how to solve a simple exercise (e.g. live-coding) rather than expecting participants to attack the problem on their own.

3.5 Participation during tutorials

During each tutorial, we—the organizers—acted as facilitators to help the leaders handle questions and manage time effectively. The leaders were often unable to see the live chat while presenting, and we therefore found natural break points in the presentation to answer questions sent to the chat. While we allowed for written and spoken questions, participants preferred to ask questions in the chat, possibly to avoid interrupting the presenter and to allow them to answer asynchronously.

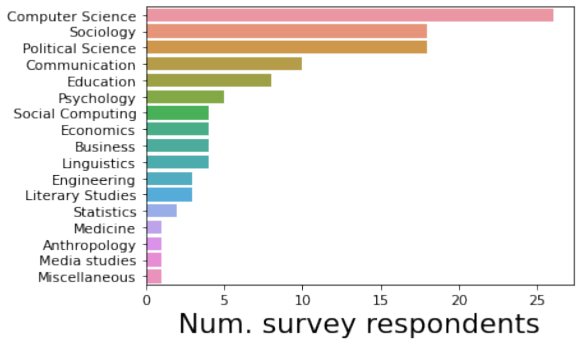
Participants were encouraged to test out the code on their own during the tutorial, and the code was generally written to execute quickly without significant lag for e.g. downloads or model training (**P.3**). This often required the leaders to run some of the code in advance to automate less interesting components of the tutorial, e.g. selecting the optimal number of topics for the topic model.

Lessons learned Based on some of the questions received, participants seemed to engage well with the code and to follow up with some of the methods. Participants asked between 1 and 15 questions per tutorial (median 5). We show example questions from the tutorials with the largest number of questions in Table 2. The questions cover both simple closed-answer questions ("Can the code provide

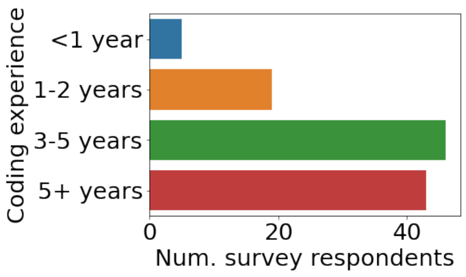
[9]Topic models are used to identify latent groupings for words in a document, e.g. a health-related topic might include "exercise" and "nutrition." (4)

Tutorial	Question sample
T1	Is there a reason that we're using word2vec rather than other models such as fastText? What does Euclidean distance between embeddings mean? Does word2vec work on short "documents" such as Twitter data?
T4	How is the bag of words representation combined with contextualized representation? How should someone choose the model to use for this component? What models does your package support?
T6	Are Phrases mostly Nouns since Nouns are the ones that have multi-words? Have you tried this model in languages other than English? How was the PhraseBert model trained?
T7	How do you keep track of who's responding to what previous utterance? How do you create a conversation corpus from scratch? Can the code provide statistics or summary for each speaker or utterance?
T9	Could you interpret a well-calibrated model as estimating the moral outrage in a post? Do choices in favor of hard modeling an aggregation techniques lead to higher values in outcome measurement? What would you recommend to handle annotator disagreement when the task is to label spans inside the text?

Table 2: Example questions from tutorial sessions. Some wording changed for clarity.



(a)



(b)

	μ	σ
Pre-Q1–Learned from code (1-5)	4.00	0.94
Pre-Q2–Learned from content (1-5)	4.24	0.90
Pre- vs post-survey	E[**Post-Q3**] - E[**Pre-Q3**]	
Knowledge about the topic (1-7)	0.77*	

(c)

Figure 4: Participant responses for the survey sent during the live tutorials (aggregated from T4-T12). Figure (a) indicates participant disciplines **(Pre-Q1)** and (b) coding experience **(Pre-Q2)**. Table (c) shows *(top)* the mean (μ) and standard deviation (σ) on a 5-point Likert scale for **Post-Q1&2** and *(bottom)* for the question about participants' self-rated knowledge about the topic graded on a 7-point Likert scale, the expected value of the post survey minus the expected value of the pre-survey (E[**Post-Q3**] - E[**Pre-Q3**]). *Indicates statistical significance with p-value $< 10^{-5}$ via a two-sided T-test.

statistics") and more complicated open-ended questions ("How should someone choose the model to use"). While the number of questions was relatively low overall, the participants who asked questions were engaged and curious about the limitations and ramifications of the methods being presented. To improve participant engagement via questions, future leaders may find it useful to pose their own questions throughout the code notebook ("what do you think would happen if we applied method X while setting parameter Z=1?") as a way to guide the participants' curiosity.

4 Analysis of Effectiveness

4.1 Pre- and post-surveys during live tutorials

During the live portions of the tutorials, we distributed an optional survey to participants at the beginning and end of the one-hour sessions.[10] The pre-survey consisted of three questions in a Google form: **(Pre-Q1)** *Academic discipline background* in which participants chose one of the given disciplines or wrote their own; **(Pre-Q2)** *How many years of experience in coding/data analysis do you have?* which had four options; and **(Pre-Q3)** *How much do you currently know about the topic?* which was judged on a 7-point Likert scale with 1 described as *I know nothing about the topic*, 4 described as *I could possibly use the methods in my research, but I'd need guidance* and 7 described as *Knowledgeable, I could teach this tutorial.* The post-survey consisted of four questions: **(Post-Q1)** *Code: How much did you learn from the hands-on code aspect of the tutorial?*; **(Post-Q2)** *Content: How much did you learn from the content part of the tutorial?*; **(Post-Q3)** *Now, after the tutorial, how much do you currently know about the*

[10]During T1-T3 we were prototyping the series, so we only distributed the surveys for T4-T12.

topic? and **(Post-Q4)** *Any suggestions or changes we should make for the next tutorial?*. Questions 1 and 2 were judged on a 5-point Likert scale with 1 described as *Learned nothing new* and 5 described as *Learned much more than I could have on my own*. Question 3 was judged on the same 7-point Likert scale as the analogous question in the pre-survey.

Results We report aggregated survey responses in Figure 4. Across the eight tutorials for which we collected data, the pre-surveys had 113 respondents total and the post-surveys had 63 respondents. Figure 4a shows the results of the breakdown by academic discipline or background **(Pre-Q1)**. The three largest areas of participation came from the fields of computer science, sociology, and political science. Figure 4b shows that our participants actually had quite a lot of experience in coding or data analysis **(Pre-Q2)**–78.8% of participants who responded had three or greater years of experience in coding.

Analyzing **Post-Q1** about how much they learned from code, participants responded with $\mu = 4, \sigma = 0.94$. **Post-Q2** about learning from content was similar with $\mu = 4.24, \sigma = 0.9$. Interpreting these results, many participants perceived a high degree of learning from attending the live tutorials. We measure the pre- to post-survey learning by computing the difference between the mean of **Post-Q3** and mean of **Pre-Q3**, and we find a difference of 0.77.[11] We ran a two-sided T-test to see if the pre- versus post-survey differences were greater than zero with statistical significance, which produced a t-value of 4.16 and a p-value less than 10^{-5}. While seemingly small in aggregate, this change represents a consistent growth in perceived knowledge among participants that is surprising considering the relatively short tutorial length of one hour. Manually reading the responses from **(Post-Q4)**, participants described very positive experiences, including ”very good tutorial” “Excellent tutorial!!!” and “very helpful.”

Lessons learned As Figure 4a shows, we were successful in recruiting participants from a wide variety of social science disciplines. However, computer science or data science—top-most bar in Figure 4a—was the most represented field. In reflection, having another organizer who was primarily focused on social science, rather than NLP, would help us recruit more CSS-oriented participants and would align better with **P.1**. Responses from **Post-Q4** also indicated that the tutorials were not long enough for some participants. One participant said “It would be great to make something like this into a multi-part tutorial. It seemed like too much new material for 1 hour.” Some suggestions for future tutorial organizers could be to make the tutorials 2-3 hours long. In the first hour, the tutorial could provide an overview, followed by more advanced topics or practice in hours 2-3. It's difficult to satisfy the trade-offs of (1) audience attention bandwidth and (2) fully explaining a particular method. We also could have improved how we set audience expectations: introducing the tutorials as a crash course and explaining that participants should expect to spend 4-5 hours on their own afterwards to learn the material in depth. Furthermore, future leaders may want to require or strongly encourage participation in the surveys to improve data collection as we had relatively low participation rates.[12]

4.2 Downstream impact

Despite the relatively low synchronous participation (roughly 4-30 participants per session), the views on the tutorial videos posted to YouTube showed consistent growth during the tutorial series and even afterward, culminating in approximately 30K total views. In addition, the tutorial materials were showcased on the website for the Summer Institute for Computational Social Science,[13] and several tutorial leaders presented their tutorials again at an international social science conference, having prepared relevant materials as part of our series (**P.4**). [14] The tutorial series may therefore have the greatest impact not for the synchronous participants but instead for the large and growing audience of researchers who discover the materials after the fact and may not have the re-

[11]Ideally, we would look not at the aggregate participant responses but instead test the pairwise differences for each individual's pre- versus post-survey. However, we found that only 18 participants could be matched from pre- to post-survey due to drop-out, which is too small for pairwise significance testing.

[12]After all the tutorials were presented, we also sent a survey to the mailing list to ask about how much participants had learned from the tutorials and whether they used the material in their own work. We received only five responses total, therefore we do not present statistics here.

[13]Accessed 15 October 2022: `https://sicss.io/overview`.

[14]International Conference on Web and Social Media 2022, accessed 15 October 2022: `https://www.icwsm.org/2022/index.html/#tutorials-schedule`

sources to learn about the methods via traditional methods (**P.5**). The success of the tutorials in other contexts also points to the beginning of a virtuous cycle, in which tutorial leaders test-drive their work in an informal setting and then present a more formal version at an academic conference.

5 Conclusion

Future improvements Reflecting on our organization experience, we suggest the following improvements for future ML+X tutorial organizers:

- Despite the results of the pre-tutorial interest surveys, we made curatorial decisions about the content and we cannot be sure that we satisfied the needs of what participants wanted versus what we thought was important. Future organizers may achieve higher participation by focusing on methods with high public interest, regardless of their lower perceived utility by subject area experts.
- The two co-organizers were both computer scientists, and we largely leveraged a computer science professional network for recruitment. Future renditions would ideally include a social scientist co-organizer to provide better insight into current ML needs and desires among researchers (**P.1**), as well as helping tutorial participants feel more at ease with complicated ML methods.
- We found that participants did not consistently engage in the hands-on coding segments of the tutorials. We recommend that future tutorial leaders either simplify the hands-on coding for short sessions, or follow up on the tutorial with additional "office hours" for interested students to try out the code and ask further questions about the method (**P.3**). Similar to some computer science courses, this approach might have a lecture component and a separate "recitation" session for asking questions about the code.
- In the early stages of the tutorial series, we focused more on executing the tutorials rather than collecting quantitative data about the participants' experience. This makes it difficult to judge some aspects of the tutorials' success, especially how the tutorials were received by participants with different backgrounds and expectations. With more extensive evaluation and participation in surveys, we hope that future organizers will make quicker and more effective improvements during the course of a tutorial series.

Successes Despite these drawbacks, we believe our tutorial series succeeded in its goal—to help social scientists advance their skills beyond introductory NLP methods. We hope other ML+X tutorials can build from our successes:

- We accumulated approximately 30K total views among our public recordings. Thus, we'd encourage future ML+X organizers to put even more effort into the recordings rather than live sessions.
- Although participants came in skilled—78.8% of participants who responded had three or greater years of experience in coding (Figure 4b)–they reported aggregate increase in perceived knowledge of the methods presented—0.77 on a 7-point Likert scale.
- We generated education content for a diverse set of relevant and new NLP methods (**P.1**&**P.2**) that can accelerate social science research. The subject matter experts who led the tutorials were able to translate complicated ML concepts into understandable, step-by-step lessons. We hope future ML+X organizers can take inspiration from these tutorials' choice of content and social organization.
- Our tutorials have produced ready-to-use, modular, and freely available Python code with a low barrier to entry (**P.3**,**P.4**,**P.5**), which will provide "scaffolding" to future students seeking to start their own projects (16). We envision future ML+X organizers using this codebase as a template for releasing code in their own domain.

As machine learning methods become more available and more powerful, scientists may feel encouraged to implement these methods within their own domain-specific research. We believe tutorial series such as the one described in this report will help guide these researchers on their journey. Like the tutorials themselves, we hope that our *Principles for Democratizing ML+X Tutorials* (**P.1**–**P.5**) will be used as springboard toward more open and inclusive learning experiences for all researchers. Rather than wait for top-down solutions, we encourage other ML practitioners to get involved and shape the future of applied science by sharing their knowledge directly with scholars eager to know more.

Acknowledgments

We are deeply grateful for financial assistance from a Social Science Research Council (SSRC)/Summer Institutes in Computational Social Science (SICSS) Research Grant. We thank SIGCSE reviewers and various computer science education experts for their feedback on initial drafts. We thank the fourteen additional organizers who generously donated their expertise and time to making these tutorials possible: Connor Gilroy, Sandeep Soni, Andrew Halterman, Emaad Manzoor, Silvia Terragni, Maria Antoniak, Abe Handler, Shufan Wang, Jonathan Chang, Steve Wilson, Dhanya Sridhar, Monojit Choudhury, Sanad Rizvi, and Neha Kennard.

References

[1] Adame, F. Meaningful collaborations can end "helicopter research". *Nature* (2021).

[2] Beam, A. L., and Kohane, I. S. Big data and machine learning in health care. *Jama 319*, 13 (2018), 1317–1318.

[3] Birjali, M., Kasri, M., and Beni-Hssane, A. A comprehensive survey on sentiment analysis: Approaches, challenges and trends. *Knowledge-Based Systems 226* (2021), 107134.

[4] Blei, D. M., Ng, A. Y., and Jordan, M. I. Latent Dirichlet Allocation. *Journal of machine Learning research 3*, Jan (2003), 993–1022.

[5] Cai, C. J., and Guo, P. J. Software developers learning machine learning: Motivations, hurdles, and desires. In *2019 IEEE symposium on visual languages and human-centric computing (VL/HCC)* (2019), IEEE, pp. 25–34.

[6] De Waard, I., Abajian, S., Gallagher, M. S., Hogue, R., Keskin, N., Koutropoulos, A., and Rodriguez, O. C. Using mLearning and MOOCs to understand chaos, emergence, and complexity in education. *The International Review of Research in Open and Distributed Learning 12*, 7 (2011), 94–115.

[7] Díaz, M., Johnson, I., Lazar, A., Piper, A. M., and Gergle, D. Addressing age-related bias in sentiment analysis. In *Proceedings of the 2018 chi conference on human factors in computing systems* (2018), pp. 1–14.

[8] Harasim, L. Shift happens: Online education as a new paradigm in learning. *The Internet and higher education 3*, 1-2 (2000), 41–61.

[9] Jones, D. T. Setting the standards for machine learning in biology. *Nature Reviews Molecular Cell Biology 20*, 11 (2019), 659–660.

[10] Jordan, M. I., and Mitchell, T. M. Machine learning: Trends, perspectives, and prospects. *Science 349*, 6245 (2015), 255–260.

[11] Karniadakis, G. E., Kevrekidis, I. G., Lu, L., Perdikaris, P., Wang, S., and Yang, L. Physics-informed machine learning. *Nature Reviews Physics 3*, 6 (2021), 422–440.

[12] Ling, C., Balci, U., Blackburn, J., and Stringhini, G. A First Look at Zoombombing. In *2021 IEEE Symposium on Security and Privacy (SP)* (2021), IEEE, pp. 1452–1467.

[13] Marcelino, M. J., Pessoa, T., Vieira, C., Salvador, T., and Mendes, A. J. Learning computational thinking and Scratch at distance. *Computers in Human Behavior 80* (2018), 470–477.

[14] Mason, W., Vaughan, J. W., and Wallach, H. Computational social science and social computing, 2014.

[15] Meerbaum-Salant, O., Armoni, M., and Ben-Ari, M. Learning computer science concepts with scratch. *Computer Science Education 23*, 3 (2013), 239–264.

[16] Nam, D., Kim, Y., and Lee, T. The effects of scaffolding-based courseware for the Scratch programming learning on student problem solving skill. In *Proceedings of the 18th International Conference on Computers in Education* (2010), vol. 723, Asia-Pacific Society for Computers in Education Putrajaya, Malaysia, p. 727.

[17] Nguyen, D., Liakata, M., DeDeo, S., Eisenstein, J., Mimno, D., Tromble, R., and Winters, J. How we do things with words: Analyzing text as social and cultural data. *Frontiers in Artificial Intelligence 3* (2020), 62.

[18] Sharlach, M. Summer institute advances social science in the digital age. *Princeton Office of Engineering Communications* (2019).

[19] Summers, R. M. Artificial intelligence of COVID-19 imaging: a hammer in search of a nail. *Radiology* (2021).

[20] Tang, T., Rixner, S., and Warren, J. An environment for learning interactive programming. In *Proceedings of the 45th ACM technical symposium on Computer science education* (2014), pp. 671–676.

[21] Twigg, C. A. Models for online learning. *Educause review 38* (2003), 28–38.

[22] Vardi, M. Y. Will MOOCs destroy academia? *Communications of the ACM 55*, 11 (2012), 5–5.

[23] Wallace, A. Social learning platforms and the flipped classroom. In *2013 Second International Conference on E-Learning and E-Technologies in Education (ICEEE)* (2013), IEEE, pp. 198–200.

[24] Wiley, D. A., and Edwards, E. K. Online self-organizing social systems: The decentralized future of online learning. *Quarterly review of distance education 3*, 1 (2002), 33–46.

Appendix

We provide the full abstracts of the tutorials in Table 3, which the tutorial leaders wrote in coordination with the organizers.

	Summary
T1	We'll demonstrate an extension of the use of word embedding models by fitting multiple models on a social science corpus (using gensim's word2vec implementation), then aligning and comparing those models. This method is used to explore group variation and temporal change. We'll discuss some tradeoffs and possible extensions of this approach.
T2	This workshop provides an introduction to information extraction for social science–techniques for identifying specific words, phrases, or pieces of information contained within documents. It focuses on two common techniques, named entity recognition and dependency parses, and shows how they can provide useful descriptive data about the civil war in Syria. The workshop uses the Python library spaCy, but no previous experience is needed beyond familiarity with Python.
T3	Establishing causal relationships is a fundamental goal of scientific research. Text plays an increasingly important role in the study of causal relationships across domains especially for observational (non-experimental) data. Specifically, text can serve as a valuable "control" to eliminate the effects of variables that threaten the validity of the causal inference process. But how does one control for text, an unstructured and nebulous quantity? In this tutorial, we will learn about bias from confounding, motivation for using text as a proxy for confounders, apply a "double machine learning" framework that uses text to remove confounding bias, and compare this framework with non-causal text dimensionality reduction alternatives such as topic modeling.
T4	Most topic models still use Bag-Of-Words (BoW) document representations as input. These representations, though, disregard the syntactic and semantic relationships among the words in a document, the two main linguistic avenues to coherent text. Recently, pre-trained contextualized embeddings have enabled exciting new results in several NLP tasks, mapping a sentence to a vector representation. Contextualized Topic Models (CTM) combine contextualized embeddings with neural topic models to increase the quality of the topics. Moreover, using multilingual embeddings allows the model to learn topics in one language and predict them for documents in unseen languages, thus addressing a task of zero-shot cross-lingual topic modeling.
T5	What is BERT? How do you use it? What kinds of computational social science projects would BERT be most useful for? Join for a conceptual overview of this popular natural language processing (NLP) model as well as a hands-on, code-based tutorial that demonstrates how to train and fine-tune a BERT model using HuggingFace's popular Python library.
T6	Most people starting out with NLP think of text in terms of single-word units called "unigrams." But many concepts in documents can't be represented by single words. For instance, the single words "New" and "York" can't really represent the concept "New York." In this tutorial, you'll get hands-on practice using the phrasemachine package and the Phrase-BERT model to 1) extract multi-word expressions from a corpus of U.S. Supreme Court arguments and 2) use such phrases for downstream analysis tasks, such as analyzing the use of phrases among different groups or describing latent topics from a corpus.
T7	ConvoKit is a Python toolkit for analyzing conversational data. It implements a number of conversational analysis methods and algorithms spanning from classical NLP techniques to the latest cutting edge, and also offers a database of conversational corpora in a standardized format. This tutorial will walk through an example of how to use ConvoKit, starting from loading a conversational corpus and building up to running several analyses and visualizations.
T8	hmm howwww should we think about our #NLProc preprocessing pipeline when it comes to informal TEXT written by social media users?!? In this tutorial, we'll discuss some interesting features of social media text data and how we can think about handling them when doing computational text analyses. We will introduce some Python libraries and code that you can use to process text and give you a chance to experiment with some real data from platforms like Twitter and Reddit.
T9	NLP has helped massively scale-up previously small-scale content analyses. Many social scientists train NLP classifiers and then measure social constructs (e.g sentiment) for millions of unlabeled documents which are then used as variables in downstream causal analyses. However, there are many points when one can make hard (non-probabilistic) or soft (probabilistic) assumptions in pipelines that use text classifiers: (a) adjudicating training labels from multiple annotators, (b) training supervised classifiers, and (c) aggregating individual-level classifications at inference time. In practice, propagating these hard versus soft choices down the pipeline can dramatically change the values of final social measurements. In this tutorial, we will walk through data and Python code of a real-world social science research pipeline that uses NLP classifiers to infer many users' aggregate "moral outrage" expression on Twitter. Along the way, we will quantify the sensitivity of our pipeline to these hard versus soft choices.
T10	Does the politeness of an email or a complaint affect how quickly someone responds to it? This question requires a causal inference: how quickly would someone have responded to an email had it not been polite? With observational data, causal inference requires ruling out all the other reasons why polite emails might be correlated with fast responses. To complicate matters, aspects of language such as politeness are not labeled in observed datasets. Instead, we typically use lexicons or trained classifiers to predict these properties for each text, creating a (probably noisy) proxy of the linguistic aspect of interest. In this talk, I'll first review the challenges of causal inference from observational data. Then, I'll use the motivating example of politeness and response times to highlight the specific challenges to causal inference introduced by working with text and noisy proxies. Next, I'll introduce recent results that establish assumptions and a methodology under which valid causal inference is possible. Finally, I'll demonstrate this methodology: we'll use semi-synthetic data and adapt a text representation method to recover causal effect estimates.
T11	Code-mixing, i.e., the mixing of two or more languages in a single utterance or conversation, is an extremely common phenomenon in multilingual societies. It is amply present in user-generated text, especially in social media. Therefore, CSS research that handles such text requires to process code-mixing; there are also interesting CSS and socio-linguistic questions around the phenomenon of code-mixing itself. In this tutorial, we will equip you with some basic tools and techniques for processing code-mixed text, starting with hands-on experiments with word-level language identification, all the way up to methods for building code-mixed text classifiers using massively multilingual language models.
T12	Word embeddings such as word2vec have recently garnered attention as potentially useful tools for analysis in social science. They promise an unsupervised method to quantify the connotations of words, and compare these across time or different subgroups. However, when training or using word embeddings, researchers may find that they don't work as well as expected, or produce unreplicable results. We focus on three subtle issues in their use that could result in misleading observations: (1) indiscriminate use of analogical reasoning, which has been shown to underperform on many types of analogies; (2) the surprising prevalence of polysemous words and distributional similarity of antonyms, both leading to counterintuitive results; and (3) instability in nearest-neighbor distances caused by sensitivity to noise in the training process. Through demonstrations, we will learn how to detect, understand, and most importantly mitigate the effects of these issues.

Table 3: Tutorial abstracts, provided by leaders.