# An Interactive Toolkit for Approachable NLP

**AriaRay Brown[1], Julius Steuer[1], Marius Mosbach[2*], Dietrich Klakow[1]**

[1]Saarland University, [2]Mila Quebec AI Institute,
{arbrown,jsteuer,dklakow}@lsv.uni-saarland.de
marius.mosbach@mila.quebec

## Abstract

We present a novel tool designed for teaching and interfacing the information-theoretic modeling abilities of large language models. The Surprisal Toolkit allows students from diverse linguistic and programming backgrounds to learn about measures of information theory and natural language processing (NLP) through an online interactive tool. In addition, the interface provides a valuable research mechanism for obtaining measures of surprisal. We implement the toolkit as part of a classroom tutorial in three different learning scenarios and discuss the overall receptive student feedback. We suggest this toolkit and similar applications as resourceful supplements to instruction in NLP topics, especially for the purpose of balancing conceptual understanding with technical instruction, grounding abstract topics, and engaging students with varying coding abilities.

## 1 Introduction

The field of information theory has seen intriguing results in the computational modeling of human language processing. Measures of information encoded in linguistic units can be used to predict the processing difficulty, or surprisal, of language (Hale, 2001; Levy, 2008) . The topic of surprisal is relevant both to researchers who want to investigate language using measures of information density, and to students of linguistics and computer science who benefit from learning about the subject.

There is also a need for tactile, communicative, and individualized learning tools. Online tools in particular provide full flexibility for hybrid or online class environments. Visual tools that aid in understanding language model outputs, such as projects from Hoover et al. (2019); Vig (2019), are also supportive of taking steps towards interpreting models (Belinkov and Glass, 2019). Such tools for learning about abstract concepts can provide students with a conceptual intuition that they can build upon and improve.

One existing public tool, OpenAI's Playground (OpenAI, 2024), offers exploratory functionality for interacting with large language models and a modest view of token probabilities. While the Playground showcases an appealing example of a user interface, we have yet to see a toolkit available that is fitting for the goals of research and education in surprisal theory. Notably, this setting calls for a tool with payment-free usage, easily retrievable surprisal calculations, and an extendable offering of publicly available language models, ideally through a simplified user interface. With this in mind, we developed the Surprisal Toolkit as an open-source research and educational tool[1] .

The Surprisal Toolkit was built in part to exist with a suite of language modeling tools for measuring aspects of information density. As part of a larger research aim to share computational tools across related projects, the Toolkit interface enables researchers with or without programming skills to obtain and analyze surprisal. Secondly, students with an interest in learning about measures of information density or examining their importance can interact with the Toolkit as an educational tool.

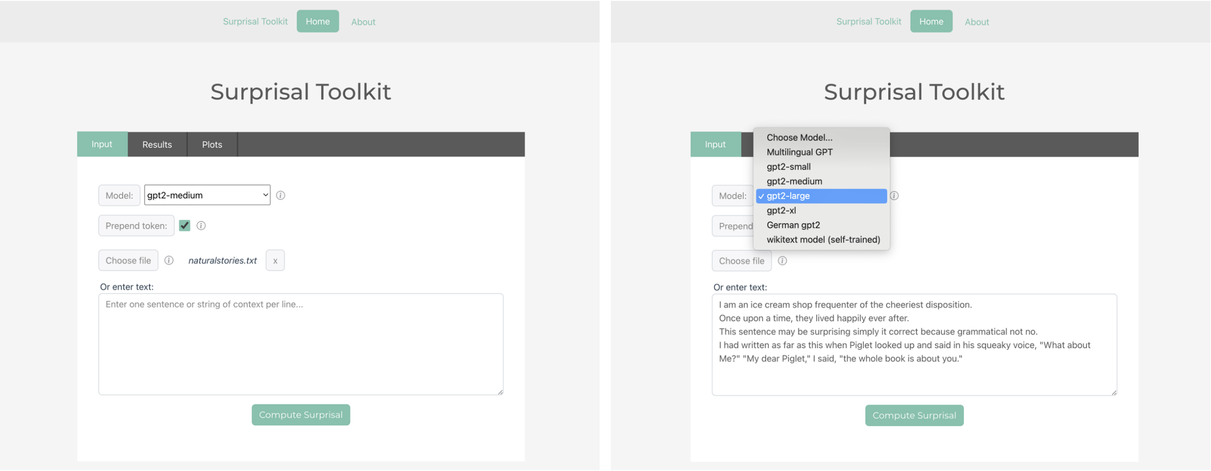
We begin with the measure of surprisal and illustrate in the following sections how the web-based Toolkit supports multiple classroom environments, for a range of student profiles, in sessions taught both in person and online.

This work presents the Surprisal Toolkit, an online interface for interacting with and teaching concepts of surprisal. We demonstrate the usefulness of this tool to encourage educators and developers to consider making use of similar resources in NLP courses.

[*] Work done while at Saarland University.

[1] https://github.com/uds-lsv/surprisal-toolkit

*Proceedings of the Sixth Workshop on Teaching NLP*, pages 119–127
August 15, 2024 ©2024 Association for Computational Linguistics



Model: gpt2-medium

Prepend token: ☑

naturalstories.txt x

Enter one sentence or string of context per line...

Model:

I am an ice cream shop frequenter of the cheeriest disposition.
Once upon a time, they lived happily ever after.
This sentence may be surprising simply it correct because grammatical not no.
I had written as far as this when Piglet looked up and said in his squeaky voice, "What about Me?" "My dear Piglet," I said, "the whole book is about you."

Figure 1: The Surprisal Toolkit web interface with selected user input, e.g., *Model: gpt2-medium, Prepend Token: selected, Chosen file: naturalstories.txt*, in the left window. The right window depicts text box input with 4 unique sentences separated onto new lines. Here, the open *Model* list reveals a starting selection of pretrained and self-trained models sourced from Hugging Face and *languagemodels*, respectively.

## 2 Learning Objectives

The tutorial we teach with the Surprisal Toolkit provides a hands-on approach to analyzing surprisal estimates from large language models. Students directly engage with applications of information theory, calculating surprisal values from model predictions and statistically comparing results in order to evaluate the alignment of machine to human language processing. The learning objectives are as follows:

1. *Learn to calculate surprisal with the Toolkit, becoming familiar with an abstract concept through concrete, visual examples.*
2. *Learn how and why to use surprisal for psycholinguistic research.* Statistically model surprisal as a predictor of human reading times using Linear Mixed Effects (LME) models. Evaluate model fit using log-likelihood and mean squared error (MSE).
3. *Learn to calculate token and word-level surprisal directly with code, machine learning libraries, and language model output.*

## 3 The Surprisal Toolkit

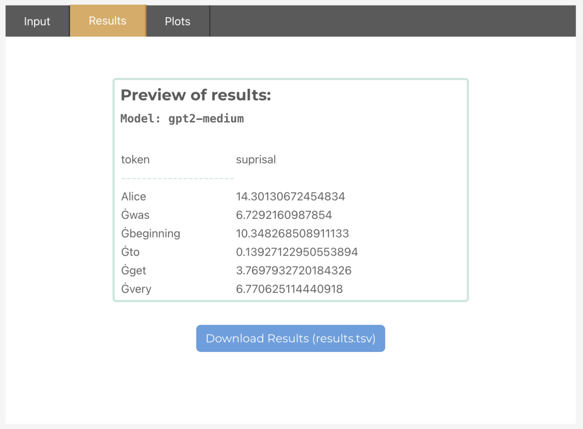
### Application architecture.

The Surprisal Toolkit shown in Figure 1 is a web-based application built to interface with a language-modeling Python library, *languagemodels*, developed for information theoretic research at a large university. The *languagemodels* library supplies custom surprisal functions using PyTorch (Paszke et al., 2019) and serves as a wrapper for functionalities from Hugging Face Transformers (Wolf et al., 2019). Together with access through a browser, the Surprisal Toolkit allows for calculating and visualizing surprisal values from language models, either internally provided or externally accessed through Hugging Face Transformers.

The application was developed using Flask for back-end functionality along with ease of integration with Python in *languagemodels*, and Angular for the front-end design and user interface. We host the application on an existing web domain of the research group to allow students to directly access the Toolkit without the need for each student to build the project locally.

### Purpose and benefits.

As a visual and computational tool, the Surprisal Toolkit serves two main purposes in our learning communities. First, it *simplifies access* to working with language models for students and researchers who lack experience in coding. As a stand-alone tool, it allows users to specify input, quickly obtain surprisal estimates, and allot focus to evaluating results. Thus attention is freed for assessing hypotheses or conceptually grasping adjacent learning objectives. The second purpose is to act as an interactive resource for students to *learn and experiment* with the topic of surprisal from language models. The Toolkit demonstrates both the theoretical topic of surprisal and its technical realization. The Toolkit provides a student-led introduction to the topic as well as a balance of high-level understanding. This allows it to be cohesively combined with subsequent coding instruction for implementing the processes observed in the interface.



Preview of results:

`Model: gpt2-medium`

| token | suprisal |
|---|---|
| Alice | 14.30130672454834 |
| Ġwas | 6.7292160987854 |
| Ġbeginning | 10.348268508911133 |
| Ġto | 0.13927122950553894 |
| Ġget | 3.7697932720184326 |
| Ġvery | 6.770625114440918 |

Figure 2: View of the results file preview in the Surprisal Toolkit web interface. Users can scroll through the window of token and surprisal estimates from the selected model (here, `gpt2-medium`) for the given text input. The complete *results.tsv* file can be downloaded via the button underneath.

## 3.1 Use Cases

We discuss several use cases for the Surprisal Toolkit as a scientific learning and research tool.

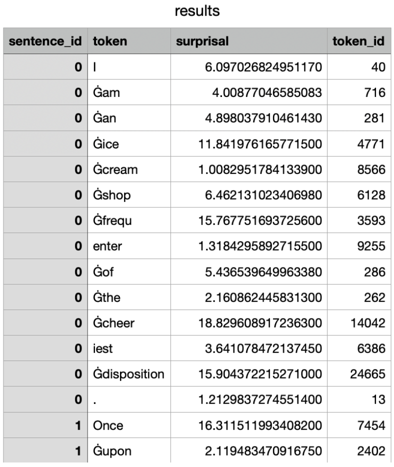
### Case I: Basic usage.

The simple usage of the Toolkit is as a pre-built calculator for computing surprisal from language models across text. Surprisal is calculated on the token level (i.e., words, characters, or subwords), as word-level surprisal can be obtained by summing the retrieved surprisal values.

- Users enter text into a text box or upload a text file, select a pretrained language model, and click *Compute Surprisal*, as portrayed in Figure 1.

- In the *Results* tab of the interface shown in Figure 2, a scrolling preview of tokens and surprisal values per line of context is shown. Below it, a *Download Results (results.tsv)* button allows users to save the complete tab-separated values file with columns for `sentence_id`, `token`, `surprisal`, and `token_id`.

- To visually explore the data in the *Plots* tab (Figure 4), users select a sentence from the input text to display a plot of log base 2 surprisal across all tokens. Plot views can be adjusted by panning the window, zooming in and out, and fitting automatically. Helpful views may then be downloaded as PNG image files.



results

| sentence_id | token | surprisal | token_id |
|---|---|---|---|
| 0 | I | 6.097026824951170 | 40 |
| 0 | Ġam | 4.00877046585083 | 716 |
| 0 | Ġan | 4.898037910461430 | 281 |
| 0 | Ġice | 11.841976165771500 | 4771 |
| 0 | Ġcream | 1.0082951784133900 | 8566 |
| 0 | Ġshop | 6.462131023406980 | 6128 |
| 0 | Ġfrequ | 15.767751693725600 | 3593 |
| 0 | enter | 1.3184295892715500 | 9255 |
| 0 | Ġof | 5.436539649963380 | 286 |
| 0 | Ġthe | 2.160862445831300 | 262 |
| 0 | Ġcheer | 18.829608917236300 | 14042 |
| 0 | iest | 3.641078472137450 | 6386 |
| 0 | Ġdisposition | 15.904372215271000 | 24665 |
| 0 | . | 1.2129837274551400 | 13 |
| 1 | Once | 16.311511993408200 | 7454 |
| 1 | Ġupon | 2.119483470916750 | 2402 |

Figure 3: A cropped example of a downloaded *results.tsv* file.

### Case II: Comparing language models.

The Toolkit provides an ongoing list of pretrained language models from Hugging Face of varying parameter sizes. Thus, results can be compared between them. Users may enter text or upload a text file of the same input while selecting differing models to calculate surprisal. By visualizing results in the *Plots* tab of the Toolkit, or downloading results in the *Results* tab, measures of surprisal from each model can be quickly viewed or saved for further comparison. In psycholinguistic investigations such as those in Oh and Schuler (2023); Kuribayashi et al. (2023), for example, surprisal estimates are used to compare varying models' abilities to predict human reading behavior.

### Case III: Computing surprisal over text.

Surprisal values can be computed over plain text in the context of sentences, stories, or documents. Results can then be used for subsequent processing or as data to investigate research questions. As an example, in the work of Wilcox et al. (2023) the authors derive surprisal estimates for 11 languages using multilingual models such as mGPT, a variant of GPT-3 pretrained on 61 languages (Shliazhko et al., 2022), in order to assess surprisal theory cross-linguistically. mGPT, available as a pretrained model through Hugging Face, can also be selected as input in the Surprisal Toolkit.

In addition to plain text, CoNLL-U[2] formatted text files, a revised version of the CoNLL-X format (Buchholz and Marsi, 2006), can be processed through the Toolkit. Downloaded result files will include an updated CoNLL-U text file with an additional column holding surprisal values.

**Case IV: Visualizing surprisal estimates.**

In the *Plots* tab (see Figure 4), users can visualize surprisal values across tokens in a sentence or line of context. By comparing surprisal modulations across sentences, users can investigate hypotheses or gather quick insights. This use case is especially helpful for demonstrating to students how surprisal increases, decreases, or persists across token values.

## 4 Classroom Implementation

The tutorial was experienced in several learning formats for a range of student backgrounds. We describe each scenario in the following subsections with a highlight of how using the Surprisal Toolkit addressed the specific needs of the class environment.

In all classroom implementations, we presented the tutorial through online materials[3]. We began with a presentation shared on a large screen, communicating either the details of the session, as in settings 4.1 and 4.2, or a brief introduction to the topic, as in setting 4.3. In hybrid settings, online participants joined through a video meeting in which the screen was also shared. The tutorial took place over a span of 90 minutes, segmented by an introduction, group question answering, and checkpoints throughout students' self-paced learning.

The format of the tutorial was comprised of a Python Jupyter Notebook, a computational notebook with interactive code blocks for sequential documenting and visualizing of code, and a web browser for accessing the Surprisal Toolkit. The Jupyter Notebook provided four sections for students to work through independently or in pairs. Section 1 documented a guided *Surprisal Toolkit Warm-Up* in which students interacted with the web interface to familiarize with calculating surprisal over tokens. In Section 3, students used the Toolkit to quickly gather surprisal results over the Natural Stories Corpus (Futrell et al., 2020) based on probability estimates from three different GPT-2 model sizes.

The premise of the notebook was to reproduce part of the experiments in (Oh and Schuler, 2023) in order to compare statistical models of human reading time data with and without LLM surprisal values as a predictor. In other words, students were given the problem of human and neural language processing in order to explore the degree to which larger language models might be worse at predicting reading times, and why. By providing a simple interface, or input-output mechanism, for obtaining the surprisal data through the Toolkit, students were able to focus in this section on the type of research questions that could be investigated using the results. This exercise was meant in part to showcase *how* the measure of surprisal could be used in psycholinguistic research and in the analysis of large language models, thus motivating a reason to learn its calculation.

The main coding focus of the notebook was then to calculate surprisal from language model outputs, without using the Surprisal Toolkit. This section explores the technical implementation and draws attention to insights for programming directly with Transformer language models.

### 4.1 Course Tutorial

As part of a seminar on information theory offered at a large university, the tutorial was presented to a group of 10-15 students as a practical session. Here, students were given the opportunity to apply the information they learned in an earlier lecture about neural language models.

**Student demographic.**

The information theory course was offered particularly for graduate students, i.e. master's and doctoral students, as well as postdoctoral researchers with a related research focus. Thus, students were expected to be moderately informed on the subject of natural language processing, while in the midst of learning about information theory, neural language processing, and psycholinguistic research. Programming experience was not required, but students were familiar enough with running Python Jupyter Notebooks, installing dependencies, and coding functions to be able to work through technical aspects of the tutorial. The majority of students

[2] See https://universaldependencies.org/format.html for CoNLL-U documentation.

[3] Materials are shared as an example at: https://github.com/uds-lsv/surprisal-toolkit-teaching-materials

were present in person, while 3-5 engaged in the tutorial through an online meeting. Students were motivated to work and learn independently, especially as many shared the goal of being able to directly use the skills from the tutorial.

**Toolkit impact:** ***individualized examples and theory in practice.***

For students in this scenario, the Surprisal Toolkit was useful for grounding the concept of surprisal. After learning about its calculations and implications in psycholinguistic research, students were given this tool to explore surprisal directly. This was achievable through self-guided experiments in which students expressed their hypotheses as input and inspected results through the Toolkit output. Students observed important points in the process of model selection, tokenization, and the modulation of surprisal values across a string of context.

In conjunction with lectures on the theoretical aspects of surprisal and a written tutorial describing technical implementations, the Toolkit supplemented students' learning with a hands-on approach to interacting with the concept. For example, students were able to see a range in surprisal values calculated across a sentence as in Figure 4. By clicking between sentences, the model's processing of language could be compared. When students were curious about the results they saw, they were easily able to modify the input text or model selection to gather more feedback for their question.

Each run of the Toolkit supplied an *individualized example*, of interest to the student who chose it. This was especially important for teaching the concept in a way that was relatable and accessible to every student.

## 4.2 Pop-up Tutorial

We next taught a tutorial on information theory, entitled "Surprisal from Large Language Models". The tutorial was open to all master's students in the university's department of Language Science and Technology who had an interest in learning more about gathering and analyzing surprisal estimates from large language models. We organized the tutorial independently of any courses in order to offer it as a distinct learning module to all interested students. For those who registered, we provided a few external readings as optional background knowledge in preparation. These included the first few pages of an introduction to information theory (Stone, 2015) and two recently published papers on the relation of surprisal from larger large language models (Oh and Schuler, 2023), as well as those from instruction-tuned models (Kuribayashi et al., 2023), and human reading behavior.

**Student demographic.**

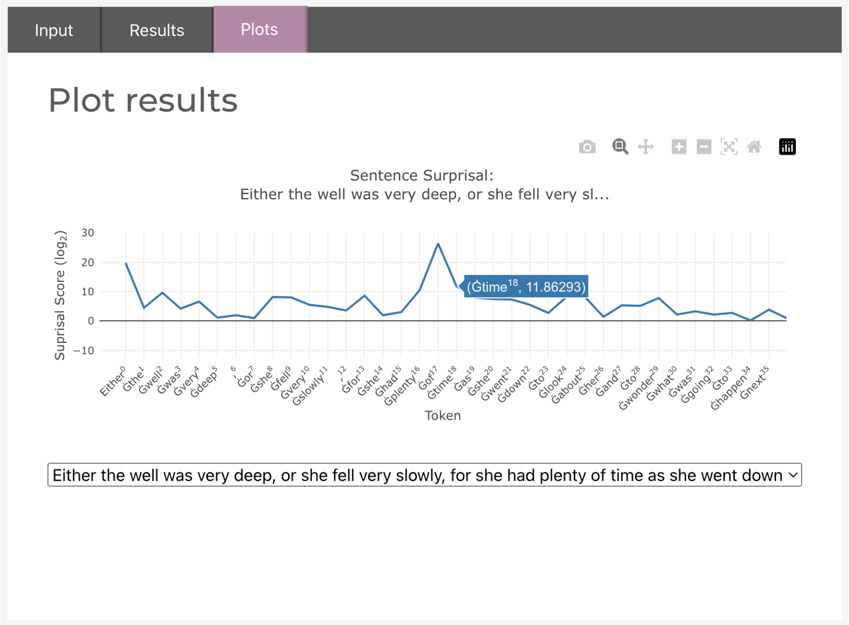
A focused class size of six master's students joined the tutorial, with two participating through an online video meeting. Due to the optional nature of the session, we assume that those who took part in it were students with a special interest in learning about the topic. Most had used Python at least once previously in their courses. Background knowledge on information theory ranged from students having no formal instruction to students being generally familiar with the concept after exposure to it in courses on computational psycholinguistics. All students were either in the beginning or middle of a degree in Language Science and Technology.

**Toolkit impact:** ***introducing and demonstrating main concepts.***

In this setting, the Surprisal Toolkit served a similar purpose of bolstering learning engagement as described in 4.1. Since students did not receive a formal lecture of instruction prior to the tutorial tasks, the Toolkit served as both an *introduction* and a *demonstration of the main concepts*. The tutorial began with a brief discussion of the learning objectives, a definition of surprisal as it relates to language processing, and research findings in the use of language model estimates to predict human reading times. Students, both online and in person, were eager to test the capabilities of the Toolkit[4].

An example interaction illustrates the benefit of having the Toolkit in this setting: While students were exploring the Toolkit, they shared several questions relating to (i) *how* the application was able to produce surprisal estimates, (ii) *what steps* were taken by the selected language model for estimation, and (iii) *what meaning* could be interpreted from the values displayed in the plot of tokens across a sentence. Essentially, all three of these questions would be answered while working through the code and instructions in the tutorial Jupyter notebook. The Toolkit thus became a precursor to the more technical and theoretical investigations within the written tutorial. Engaging with

[4] We elaborate on how to prepare for enthusiastic memory usage of web-based tools in Section 7 but ultimately were able to support the engagement.



Either the well was very deep, or she fell very slowly, for she had plenty of time as she went down

Figure 4: Plot of surprisal values from an excerpt of sentences from *Alice's Adventures in Wonderland* by Lewis Carroll. On the x-axis is the model-tokenized sentence or line of text. The Ġ symbol represents white space and, when prefixed to a token, indicates the start of an orthographic word. On the y-axis is the log base 2 surprisal score for each token. Below the plot is a drop-down menu for selecting a sentence or line of text from the input for viewing.

the Toolkit effectively directed students' attention to the information they would receive during the remaining tutorial.

### 4.3 Workshop

We taught the tutorial again as a workshop during a three-day computational linguistics conference designed for bachelor's and master's students of related fields. The conference was aimed at fostering knowledge exchange between students and served as both an educational and community forming event. Participants were able to attend keynote lectures given by university professors in the field of natural language processing. Lecture topics were loosely related to our tutorial topic of information theory, and may have provided interest or slight context to students who participated in the workshop. Of the day's events, students could choose to attend our workshop or opt for other talks occurring simultaneously.

**Student demographic.**

A group of 9 students, mostly master's, elected to participate in the workshop. We expected most participants to have little to no background knowledge on the subject of information theory, natural language processing, or computer programming. In this setting, we also could not expect participants to have prepared background knowledge outside of the classroom. Instead, we prepared a short introduction with key points for situating the tutorial. Since participants would be selecting from a full day of events to meet their intellectual interests, we aimed to create a learning environment that was direct, concise, salient, and enjoyable. Here we targeted learning through active engagement more so than through self-directed coding challenges, as offered in prior tutorials.

**Toolkit impact:** ***engaging students and prompting student-led experiments.***

The Toolkit became a focal point for grounding concepts and engaging students in this scenario. After presenting a short series of slides with information on surprisal theory and its calculations through language models, we introduced the Surprisal Toolkit in an interactive group warm-up.

Through student-led input, we aimed to exemplify and experiment with the notions just introduced. We prompted students to come up with hypotheses that might lead to changes in surprisal values across tokens in a line of context. By comparing visualizations of plotted results, students were able to assess possible answers to their questions and experiment with further insights gained. For example, students were asked to come up with sentences that might lead to peaks or drops in surprisal values; compare output from different models; and assess the reliability of model output, along with contributing factors thereof, given their own intuitions about language. The discussion gained from using such a tool raised several interesting observations from students, whether about influences on the measure of surprisal or the processes implemented behind the interface.

## 5 Hybrid Classroom

One benefit of web-based learning tools is their functionality in both online and in-person settings. With the usability of the Surprisal Toolkit online, for example, we have been able to adapt our tutorial for hybrid learning with little modification. An additional advantage we observe through having taught with an interactive tool is its ability to engage remote students who are unable to physically immerse in the classroom environment.

## 6 Student Feedback

At the end of each tutorial, we collected optional student feedback in the form of a ten-question survey. In an effort to ensure some feedback over none, we kept questions to a minimum and included only one free response. The first question asked for confirmation that students were able to use the Surprisal Toolkit in the tutorial. Questions 2-9 were opinion questions on a 5-point Likert scale, with 5 indicating the most positive opinion. Question 10 was open-ended, allowing for optional comments or suggestions regarding the Toolkit. Students were free to fill out the survey on paper or online through a printed QR code. Those who completed the survey did so anonymously and confirmed their consent to having their answers contribute towards future research on teaching NLP. In Table 1 we present the results from the 12 student responses collected across all three tutorial sessions[5].

Overall, student feedback was positive towards using the Surprisal Toolkit as a learning tool. The majority of students gave scores of 5 in each question. All questions except for one saw scores at 3 or above. Only one question, *Did using the toolkit help you to understand more about language models or evaluating surprisal?* received a rating of 2 from a student who attended the pop-up tutorial (4.2). One possible explanation for this response could be related to reaching the memory capacity for the Toolkit server during the warm-up of this tutorial session. The experience revealed the need to manage high usage through further application development, or to carefully plan classroom scenarios to best distribute simultaneous interactions with the tool.

When asked about satisfaction with using the Toolkit as a resource and, later, satisfaction with the tutorial overall, ratings decreased slightly, with two student scores (16.7%) reducing from 4 to 3. At a minimum, we can interpret that the Toolkit did not detract from the learning environment. Even with room for improvement in the tutorial, the Toolkit provided a mostly satisfying component.

The highest-scoring question, *How interested would you be in seeing similar applications for interacting with language models in your courses?* received almost unanimous ranking of 5 for "Very interested". This result is promising, as it suggests that students enjoyed using the Toolkit enough in this instance to look positively towards further implementations of such tools in their learning environments. As educators, researchers, and developers we may be encouraged to build and share more interactive NLP tools with students who welcome the resources.

## 7 Discussion and Suggestions

One important consideration when developing web-based learning applications is to ensure sufficient server memory is available to handle multiple simultaneous requests. We suggest two methods for addressing this need depending on the stage of development of the tool. First, in the most ideal case, function calls to large Hugging Face models should be implemented in a way that minimizes redundant memory and allows for shared resources among user requests. Second, in the case that memory is limited, an option is to use the application in group-led activities, specifically at points in the lesson dedicated mainly to exploring the application. Shared usage, where students still have the opportunity to direct the interaction with the tool, is one way to counteract memory limitations that can be just as effective for engagement. In the workshop setting described in 4.3, we found this to positively be the case.

We suggest continuously iterating over the appli-

[5] This number represents 53% of all students who attended the tutorial sessions and stayed for the full duration of the class. Of all student responses, 16.67% originated from the course tutorial, 33.33% from the pop-up tutorial, and 50.00% from the workshop.

| # | Student Feedback Question | Rating Distribution |
|---|---|---|
| 1 | How much did using the toolkit affect your learning engagement during the tutorial? *I didn't feel at all engaged...I felt very engaged* | **5** (1:0, 2:0, 3:3, 4:1, 5:8) |
| 2 | How satisfied were you with using the toolkit as a resource? *Not at all satisfied...Extremely satisfied* | **4.5** (1:0, 2:0, 3:1, 4:5, 5:6) |
| 3 | How easily were you able to interact with the toolkit? *Not at all easily...Very easily* | **5** (1:0, 2:0, 3:1, 4:2, 5:9) |
| 4 | Did using the toolkit help you to understand more about language models or evaluating surprisal? *No, strongly disagree...Yes, strongly agree* | **4.5** (1:0, 2:1, 3:2, 4:3, 5:6) |
| 5 | Did using the toolkit bring up questions about language models or surprisal that you would like to explore further? *No, strongly disagree...Yes, strongly agree* | **5** (1:0, 2:0, 3:1, 4:3, 5:8) |
| 6 | How interested would you be in using the toolkit again for a similar task? *Not at all interested...Very interested* | **5** (1:0, 2:0, 3:0, 4:5, 5:7) |
| 7 | How interested would you be in seeing similar applications for interacting with language models in your courses? *Not at all interested...Very interested* | **5** (1:0, 2:0, 3:1, 4:0, 5:11) |
| 8 | How satisfied were you with today's tutorial overall? *Not at all satisfied...Extremely satisfied* | **4.5** (1:0, 2:0, 3:3, 4:3, 5:6) |

Table 1: Student feedback (N=12): distribution of responses to opinion questions on learning with the Surprisal Toolkit. Response ratings are from 1-5, with 5 being the most positive assessment. Counts are given to the right of each rating. In bold is the median response and in blue text is the most frequent.

cation of a toolkit interface, as students bring some of the most meaningful feedback for highlighting where important features can be implemented to improve learning.

A few points of interest were inquiries about (1) why `Prepend Token` was necessary when processing text with GPT-2-based language models, (2) how much context was considered when calculating token probabilities, and (3) where more information could be found about the details of the language models themselves. We addressed these points in the user interface by adding tooltips with further information to relevant areas.

The aim was not to remove the class discussion of these facets, but to reinforce the Toolkit for use by students who rely on a reiteration of the answers or may prefer independent discovery. We expect that continual integration of student feedback would bring an ongoing interchange of more informative pedagogical research tools and more empowered student users.

Based on student feedback, we also see valuable uses for making the Toolkit open-source. This could represent an additional learning opportunity and motivation for inquisitive students to investigate the concept further, and is yet to be explored in future courses.

Ongoing work on the Toolkit can provide additional features for presenting important concepts. For example, the relationship between surprisal and perplexity might be explored through the ability to calculate and compare both measures of information density.

## 8 Conclusion

We find the Surprisal Toolkit to balance conceptual understanding with technical implementation, providing opportunities for visual learning and efficient solutions when needed. In classroom settings, the Toolkit was able to demonstrate to students that an implementation of surprisal was possible prior to building the coding calculation themselves. The

Toolkit as a method for grounding abstract concepts provides a scaffolding for adjusting programming course content to a broader range of knowledge backgrounds. A benefit of a web-based tool is that it readily integrates into online or hybrid teaching environments. Therefore we recommend implementing such tools in further topics and courses in NLP.

## Acknowledgments

Many thanks are owed to the motivated students who took part in the tutorials and offered their feedback and suggestions. We thank the organizers of the 33rd TaCoS for serendipitously contacting us with an opportunity to teach. Thank you to Zaynab Reza for helping to continue development on the Surprisal Toolkit. We also thank the reviewers for their constructive feedback. This research was supported by funding from the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation), Project ID 232722074 – SFB 1102.

## References

Yonatan Belinkov and James Glass. 2019. Analysis Methods in Neural Language Processing: A Survey. *Transactions of the Association for Computational Linguistics*, 7:49–72.

Sabine Buchholz and Erwin Marsi. 2006. CoNLL-X shared task on multilingual dependency parsing. In *Proceedings of the Tenth Conference on Computational Natural Language Learning (CoNLL-X)*, pages 149–164, New York City. Association for Computational Linguistics.

Richard Futrell, Edward Gibson, Harry J Tily, Idan Blank, Anastasia Vishnevetsky, Steven T Piantadosi, and Evelina Fedorenko. 2020. The natural stories corpus: a reading-time corpus of english texts containing rare syntactic constructions. *Language resources and evaluation, 55(1), 63–77*.

John Hale. 2001. A probabilistic Earley parser as a psycholinguistic model. In *Second Meeting of the North American Chapter of the Association for Computational Linguistics*.

Benjamin Hoover, Hendrik Strobelt, and Sebastian Gehrmann. 2019. exbert: A visual analysis tool to explore learned representations in transformers models. *arXiv preprint arXiv:1910.05276*.

Tatsuki Kuribayashi, Yohei Oseki, and Timothy Baldwin. 2023. Psychometric predictive power of large language models. *arXiv preprint arXiv:2311.07484*.

Roger Levy. 2008. Expectation-based syntactic comprehension. *Cognition*, 106(3):1126–1177.

Byung-Doh Oh and William Schuler. 2023. Why does surprisal from larger transformer-based language models provide a poorer fit to human reading times? *Transactions of the Association for Computational Linguistics*, 11:336–350.

OpenAI. 2024. Playground. `https://platform.openai.com/playground/complete`. Accessed: 2024-07-01.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. *Preprint*, arXiv:1912.01703.

Oleh Shliazhko, Alena Fenogenova, Maria Tikhonova, Vladislav Mikhailov, Anastasia Kozlova, and Tatiana Shavrina. 2022. mgpt: Few-shot learners go multilingual. *arXiv preprint arXiv:2204.07580*.

James V. Stone. 2015. Information theory: A tutorial introduction. *ArXiv*, abs/1802.05968.

Jesse Vig. 2019. A multiscale visualization of attention in the transformer model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 37–42, Florence, Italy. Association for Computational Linguistics.

Ethan G. Wilcox, Tiago Pimentel, Clara Meister, Ryan Cotterell, and Roger P. Levy. 2023. Testing the Predictions of Surprisal Theory in 11 Languages. *Transactions of the Association for Computational Linguistics*, 11:1451–1470.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.