SyntaxGym: An Online Platform for Targeted Evaluation of Language Models

Jon Gauthier[1], **Jennifer Hu**[1], **Ethan Wilcox**[2], **Peng Qian**[1], and **Roger Levy**[1]
[1] Department of Brain and Cognitive Sciences, Massachusetts Institute of Technology
[2] Department of Linguistics, Harvard University
jon@gauthiers.net, wilcoxeg@g.harvard.edu
{jennhu,pqian,rplevy}@mit.edu

Abstract

Targeted syntactic evaluations have yielded insights into the generalizations learned by neural network language models. However, this line of research requires an uncommon confluence of skills: both the theoretical knowledge needed to design controlled psycholinguistic experiments, and the technical proficiency needed to train and deploy large-scale language models. We present SyntaxGym, an online platform designed to make targeted evaluations **accessible** to both experts in NLP and linguistics, **reproducible** across computing environments, and **standardized** following the norms of psycholinguistic experimental design. This paper releases two tools of independent value for the computational linguistics community:

1. A website, **syntaxgym.org**, which centralizes the process of targeted syntactic evaluation and provides easy tools for analysis and visualization;
2. Two command-line tools, `syntaxgym` and `lm-zoo`, which allow any user to reproduce targeted syntactic evaluations and general language model inference on their own machine.

1 Introduction

Recent work in evaluating neural network language models focuses on investigating models' fine-grained prediction behavior on carefully designed examples. Unlike broad-coverage language modeling metrics such as perplexity, these evaluations are targeted to reveal whether models have learned specific knowledge about the syntactic structure of language (see e.g. Warstadt et al., 2020; Futrell et al., 2019; Marvin and Linzen, 2018).

Research in this line of work requires an uncommon intersection of skills: a) the engineering strength of NLP researchers necessary to train and deploy large-scale neural network language models, and b) the linguistic knowledge of language scientists necessary to design controlled, theoretically interesting psycholinguistic experiments.

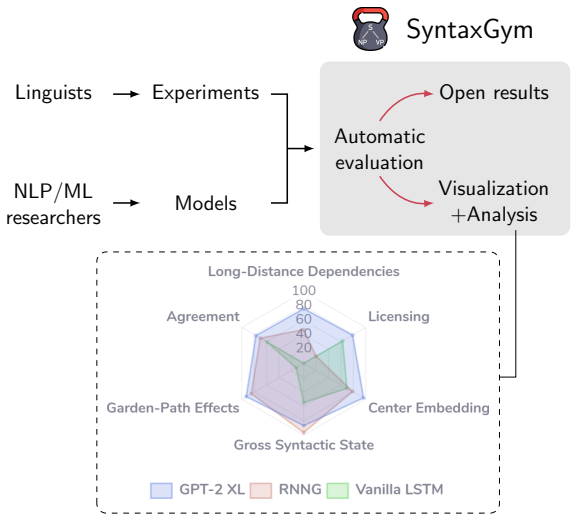

Figure 1: SyntaxGym allows linguists to easily design and run controlled experiments on the syntactic knowledge of language models, and allows NLP experts to test their own models against these standards. Users submit targeted syntactic evaluation experiments to the site, and they are automatically evaluated on language models available in the Gym. SyntaxGym analyzes and visualizes these evaluation results.

In this paper, we introduce **SyntaxGym**: an online platform and open-source framework that makes targeted syntactic evaluations more accessible to experts in NLP and linguistics (Figure 1). The core of SyntaxGym is a website, **syntaxgym.org**, that automates the entire evaluation pipeline: collecting tests and models, running evaluations, and displaying results through interactive visualizations. Language scientists can use the site to design and submit targeted syntactic evaluations, testing whether language models have derived human-like syntactic knowledge. Indepen-

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 70–76

dently, NLP experts can submit their own language models for evaluation on these assays. By separating the tasks performed by these two user groups, the SyntaxGym site lowers the barrier to entry for the broader community of language researchers.

While SyntaxGym will serve as a centralized repository of syntactic evaluations and language models, we also release a set of command-line tools that allow users to reproduce the site's evaluations offline. The computation underlying the SyntaxGym site is structured around a command-line tool `syntaxgym`, which allows any user to run targeted syntactic evaluations on their own computer. We accomplish this by developing a new standard API for interacting with state-of-the-art neural network language models, operationalized in a second tool `lm-zoo`.

Taken together, these tools create a platform that makes the process of targeted syntactic evaluation more standardized, reproducible, and accessible to the broader communities of NLP experts and language scientists. Our goal is for SyntaxGym to facilitate the advancement of language model evaluation, leading to the development of models with more human-like linguistic knowledge.

2 Background

Before presenting the SyntaxGym framework, we briefly introduce the targeted syntactic evaluation paradigm as a way to assess the quality of neural language models.

2.1 Perplexity

Standard left-to-right language models are trained to predict the next token given a context of previous tokens. Language models are typically assessed by their *perplexity*, the inverse geometric mean of the joint probability of words $w_1, \ldots, w_N$ in a held-out test corpus:

$$\text{PPL}(C) = p(w_1, w_2, \ldots w_N)^{-\frac{1}{N}} \qquad (1)$$

However, a broad-coverage metric such as perplexity may not be ideal for assessing whether a language model has human-like syntactic knowledge. Recent empirical results suggest that models with similar perplexity measures can still exhibit substantial variance in syntactic knowledge (Hu et al., 2020; van Schijndel et al., 2019), according to evaluation paradigms described in the next section.

2.2 Targeted tests for syntactic generalization

Alternatively, a language model can be evaluated on its ability to make human-like generalizations for specific syntactic phenomena. The targeted syntactic evaluation paradigm (Linzen et al., 2016; Lau et al., 2017; Gulordava et al., 2018; Marvin and Linzen, 2018; Futrell et al., 2019; Warstadt et al., 2020) incorporates methods from psycholinguistic experiments, designing sentences which hold most lexical and syntactic features of each sentence constant while minimally varying features that determine grammaticality or surprise characteristics of the sentence. For example, the following minimal-pair sentences differ in subject–verb agreement:

(1) The farmer near the clerks knows many people.

(2) *The farmer near the clerks know many people.

A model that has learned the proper subject–verb number agreement rules for English should assign a higher probability to the grammatical plural verb in the first sentence than to the ungrammatical singular verb in the second (Linzen et al., 2016).

3 SyntaxGym

The targeted syntactic evaluation paradigm allows us to focus on highly specific measures of language modeling performance, which more directly distinguish models with human-like representations of syntactic structure. SyntaxGym was designed to serve as a central repository for these evaluations, and to make the evaluations reproducible and accessible for users without the necessary technical skills or computational resources.

Section 3.1 first describes the standards we designed for specifying and executing these targeted syntactic evaluations. Section 3.2 then offers a tour of the SyntaxGym site, which is built around these standards.

3.1 Standardizing targeted syntactic evaluation

We represent targeted syntactic evaluations as *test suites*, visualized in Figure 2. These test suites are the core component of psycholinguistic assessment, and should be familiar to those experienced in psycholinguistic experimental design. We will present the structure of a test suite using the running example of subject–verb agreement, introduced in the previous section. We describe the components of a test suite from bottom-up:

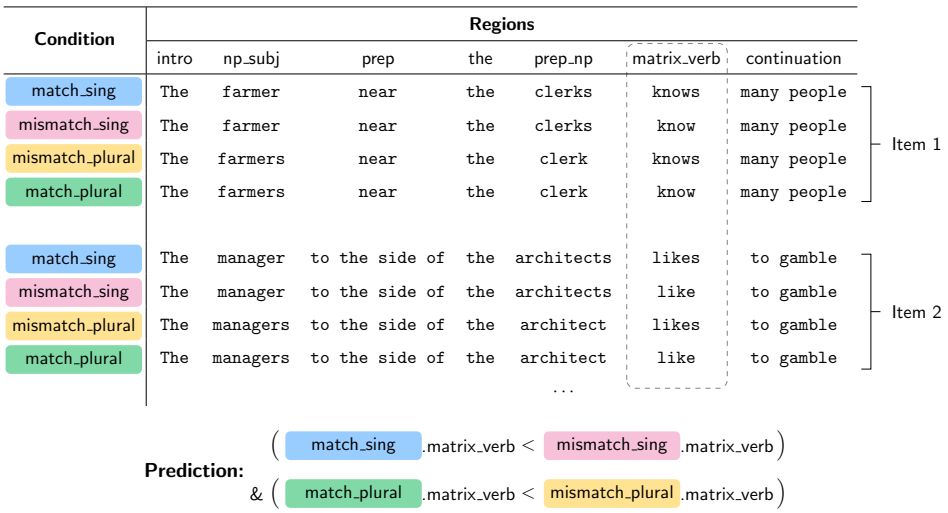

Figure 2: SyntaxGym test suites evaluate predictions about language models' surprisal values (negative log-probabilities) within regions (columns above) across experimental conditions (leftmost column). A prediction can assert the conjunction of multiple inequalities across conditions. Prediction results are aggregated across items (vertical blocks above) to yield overall accuracy estimates.

Regions The atomic unit of a test suite is a region: a (possibly empty) string, such as the `matrix_verb` region in Figure 2. Regions can be concatenated to form full sentences.

Conditions Regions vary systematically across experimental conditions, shown as colored pill shapes in Figure 2. Here the `matrix_verb` and `np_subj` regions vary between their respective singular and plural forms, as described by the condition.

Items Items are groups of related sentences which vary across experimental conditions. An item is characterized by its lexical content and takes different forms across conditions. For example, *The farmer near the clerk knows* and **The farmer near the clerk know* are different sentences under two conditions of the same item.

Predictions Test suites are designed with a hypothesis in mind: if a model has correctly learned some relevant syntactic generalization, then it should assign higher probability to grammatical continuations of sentences. Test suite predictions operationalize these hypotheses as expected inequalities between total model surprisal values in different experimental conditions (i.e., between rows within item blocks in Figure 2). The SyntaxGym standard allows for arbitrarily complex disjunctions and conjunctions of such inequalities. Figure 2 shows a prediction with two inequalities between model surprisals at `matrix_verb` across two pairs of conditions.

We designed a standard JSON schema for describing the structure and content of test suites using the above concepts. Interested readers can find the full schema and documentation on the SyntaxGym site.[1]

3.1.1 A standard API for language models

Reproducing research results with modern neural network architectures can be notoriously difficult, due to variance in computing environments and due to each individual project's tangled web of package dependencies. In addition, inconsistencies in data preprocessing — for example, in tokenization practices and the management of out-of-vocabulary items — often make it difficult to evaluate even the same model on different datasets. In order to address these difficulties, we designed a standardized API for interacting with trained language models, built to solve these reproducibility issues and allow for highly portable computing with state-of-the-art language models. Users can easily connect with this API through the `lm-zoo` command-line tool, described later in Section 4.

The standard is built around the Docker containerization system. We expect each language model to be wrapped in a Docker image, including a thin API exposing a set of standardized binary commands: `tokenize`, which preprocesses natural-language sentences exactly as a language

[1] http://docs.syntaxgym.org

model expects; `get-surprisals`, which computes per-token language model surprisals on natural language input; and `unkify`, which indicates exactly which tokens in an input text file are invocabulary for the language model.

Language model creators or third-party maintainers can produce such Docker images wrapping language model code. At present, this API is designed to mainly serve the needs of the SyntaxGym evaluation process. In the future, however, we plan to extend the API for other common uses of language models: for example, to extract the next-word predictive distributions from the model, and to extract the model's internal word and sentence representations. This standard is documented in full at `cpllab.github.io/lm-zoo`.

3.2 The SyntaxGym website

The SyntaxGym website provides a centralized domain for collecting targeted syntactic evaluations and evaluating them on state-of-the-art language models. It provides intuitive, user-friendly tools for visualizing the behavior of any language model on any syntactic test suite, and also exposes all of the resulting raw data to interested advanced users. This section presents a brief tour through the major features of the SyntaxGym site.

Create test suites Non-technical users can use SyntaxGym's browser-based interface to design and submit their own psycholinguistic test suites (Figure 3). Separately, the site supports uploading pre-made test suites as a JSON-formatted file. This functionality may be useful for advanced users who prefer to automatically generate test suites.[2]

Figure 3: Non-technical users can design their own test suites with a browser-based form.

Submit language models Users interested in evaluating their own language models first create a public Docker image conforming to the API specified by the SyntaxGym standard (Section 3.1.1). After users submit these language models on the SyntaxGym site, the models are automatically validated for conformity to the API by the SyntaxGym backend. Valid models are added to the SyntaxGym collection, and will be evaluated on all past and future available test suites in the Gym.

Automatic evaluation Whenever novel test suites or language models are submitted, SyntaxGym automatically evaluates the relevant suites and models in the cloud. For each test suite and model, the evaluation yields a prediction accuracy — the number of items for which the prediction holds. These prediction accuracies, along with the raw surprisal data, are stored in the SyntaxGym database and made available in visualizations such as Figure 4b.

Visualization and data analysis The site provides a variety of interactive charts that allow users to visualize results at different levels of granularity. On the coarsest level, users can compare aggregate performance across language models and groups of theoretically related test suites called *tags* (see Figure 1). Users can also compare accuracy across models on a single test suite (Figure 4a), across tags for a single model, and across test suites within a single tag. On the finest level, users can view raw region-by-region surprisal values to analyze in-depth performance of a particular language model on a particular test suite (Figure 4b).

3.3 Seed data and results

We have seeded the SyntaxGym website with a collection of test suites and language models by aggregating prior research. These materials and relevant evaluation results are separately presented in Hu et al. (2020). Here we provide only a brief summary in order to illustrate the features of the SyntaxGym website.

1. We wrapped 8 modern neural network language models (summarized in Table 1) to be compatible with the `lm-zoo` standard, using open-source research code or standard Python frameworks such as Hugging Face Transformers (Wolf et al., 2019).

2. We aggregated past research on targeted syntactic evaluation into 33 test suites, each probing language models' performance on distinct grammatical phenomena.

[2] In a future release, we will also allow users to import test suites from spreadsheets as CSV-formatted files.

(a) The site automatically evaluates language models on test suites and visualizes summary results (here, for a subject–verb number agreement test). (click to see page)

(b) Users can also view all of the raw language model data behind these analyses (here, average per-region surprisal values of GPT-2 for a subject–verb number agreement test) and download the raw data for further analysis. (click to see page)

Figure 4: Screenshots of example visualizations from the SyntaxGym website.

Model	Reference	Training data (# tokens)
GPT-2	Radford et al. (2019)	WebText (∼8B)
GPT-2 XL	Radford et al. (2019)	WebText (∼8B)
Transformer XL	Dai et al. (2019)	WikiText-103 (103M)
JRNN	Jozefowicz et al. (2016)	1B Benchmark (1.04B)
GRNN	Gulordava et al. (2018)	Wikipedia (90M)
Ordered Neurons	Shen et al. (2019)	BLLIP (42M)
LSTM	Hochreiter and Schmidhuber (1997)	BLLIP (42M)
RNNG	Dyer et al. (2016)	BLLIP (42M)

Table 1: Language models currently supported in the SyntaxGym framework.

Interested readers can find more details on these test suites and language models, along with the evaluation results and visualizations, on the SyntaxGym site.

4 Command-line tools

While the SyntaxGym website allows for easy centralization of test suites and public access to results, all of its underlying infrastructure is also available independently for researchers to use. We release two command-line tools, available to any user with Python and Docker installed.

4.1 `lm-zoo`: black-box access to SOTA language models

We first designed a general command-line tool for interacting with state-of-the-art neural language models, called `lm-zoo`. Figure 5b demonstrates how this tool can be used to easily extract prediction data from an arbitrary language model. Full documentation and installation instructions are available at `cpllab.github.io/lm-zoo`.

4.2 `syntaxgym`: targeted syntactic evaluation

Users can completely reproduce the targeted syntactic evaluation paradigm of SyntaxGym outside of the website using a second command-line tool, `syntaxgym`, shown in Figure 5a. This tool does the work of converting test suites into actual natural-language sentences appropriately formatted for a particular language model, executing the model, and mapping the results back to a SyntaxGym-friendly format ready for analysis. It deals with the wide variation in tokenization and out-of-vocabulary token handling across models. Full documentation and installation instructions are available at `syntaxgym.org/cli`.

5 Related work

Marvin and Linzen (2018) release a dataset of minimal-pair sentences designed to test language models' syntactic generalization capabilities. However, the syntactic coverage of the dataset is limited to a small set of phenomena: subject-verb agreement, reflexive anaphor licensing, and negative polarity items.

Warstadt et al. (2020) release a large dataset aggregating a broad collection of targeted syntactic evaluations from prior research, known as BLiMP. Like the Marvin and Linzen dataset, BLiMP consists of a collection of minimal-pair sentences which contrast in grammaticality, following the standard shown in Examples (1) and (2). The BLiMP evaluation requires that language models assign a higher total probability to the grammatical (1) than the ungrammatical (2). The authors design abstract templates which specify grammatical–

```
$ syntaxgym list models
gpt-2, gpt-2-xl, transformer-xl, ...

$ syntaxgym list suites
number-orc, number-src, mvrr, ...

# Evaluate model "gpt-2" on suite "mvrr"
$ syntaxgym evaluate gpt-2 mvrr
Accuracy: 0.7857 (22/28 correct)

# Evaluate arbitrary model on custom suite
$ syntaxgym evaluate \
>   docker://me/my-model my-suite.json
Accuracy: 0.575 (23/40 correct)
```

(a) The `syntaxgym` tool allows users to evaluate language models on test suites — both models and suites hosted by SyntaxGym, and models and suites created by the user.

```
$ echo "This is a sentence." > foo.txt

$ lm-zoo list models
gpt-2, gpt-2-xl, transformer-xl, ...

$ lm-zoo tokenize transformer-xl foo.txt
This is a sentence .

$ lm-zoo get-surprisals transformer-xl foo.txt
sentence_id  token_id  token  surprisal
1            1         This   0.0000
1            1         is     4.1239
1            1         a      1.0126
...
```

(b) The `lm-zoo` tool provides lower-level access to SyntaxGym-hosted language models, allowing users to retrieve models' predictions, tokenization choices, and more.

Figure 5: We built SyntaxGym around command-line tools for probing and evaluating neural network language models, which can be used independently of the SyntaxGym site.

ungrammatical pairs for many linguistic phenomena, and then generate example sentences based on these templates.

While BLiMP and SyntaxGym are similarly motivated, they differ slightly in methodology. First, BLiMP requires models to satisfy only a single inequality between sentence probabilities. While the SyntaxGym system can support such predictions, it is designed to support much stricter tests of language models, such as the conjunction of inequalities across multiple conditions (see Figure 2). Second, BLiMP compares model judgments about total sentence probabilities. In contrast, SyntaxGym is designed to compare model judgments only in critical test regions, which allows us to more fairly evaluate language model predictions only in prespecified spans of interest. Finally, the BLiMP sentences are automatically generated from abstract grammars exemplifying syntactic phenomena of interest. Since automatic methods can easily yield a large number of sentences, they can help us control for other possible sources of noise in test materials. However, many grammatical phenomena of interest are fiendishly difficult to capture in abstract grammars, and require careful design by native speakers.[3] This BLiMP data is thus complementary to the hand-designed test suites currently presented on the SyntaxGym site. We plan to adapt such large-scale test suites on SyntaxGym in the future.

[3]For example, one such phenomenon is the garden-path disambiguation effect (Futrell et al., 2019), which is highly sensitive to nuanced lexical and world-knowledge features of sentences.

6 Conclusion

This paper presented SyntaxGym, an online platform and open-source framework for targeted syntactic evaluation of neural network language models. SyntaxGym promises to advance the progress of language model evaluation by uniting the theoretical expertise of linguists with the technical skills of NLP researchers. The site is fully functional at `syntaxgym.org`, and the entire framework is available as open-source code.

SyntaxGym is continually evolving: we plan to add new features to the site, and to develop further in response to user feedback. In particular, we plan to incorporate *human performance* as a reference metric, integrating psycholinguistic experimental results and supporting easy experimental design starting from the test suite format.

We also plan to further incorporate language models into the `lm-zoo` tool, allowing broader access to state-of-the-art language models in general. We welcome open-source contributions to the website and to the general framework, and especially encourage the NLP community to contribute their models to the `lm-zoo` repository.

Acknowledgments

J.G. is supported by an Open Philanthropy AI Fellowship. J.H. is supported by the NIH under award number T32NS105587 and an NSF Graduate Research Fellowship. R.L. is supported by a Google Faculty Research Award. This work was also supported by the MIT-IBM Watson AI Lab.

References

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime G Carbonell, Quoc Le, and Ruslan Salakhutdinov. 2019. Transformer-xl: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2978–2988.

Chris Dyer, Adhiguna Kuncoro, Miguel Ballesteros, and Noah A Smith. 2016. Recurrent neural network grammars. In *Proceedings of NAACL-HLT*, pages 199–209.

Richard Futrell, Ethan Wilcox, Takashi Morita, Peng Qian, Miguel Ballesteros, and Roger Levy. 2019. Neural language models as psycholinguistic subjects: Representations of syntactic state. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 32–42, Minneapolis, Minnesota. Association for Computational Linguistics.

Kristina Gulordava, Piotr Bojanowski, Edouard Grave, Tal Linzen, and Marco Baroni. 2018. Colorless green recurrent networks dream hierarchically. In *Proceedings of NAACL-HLT*, pages 1195–1205.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Jennifer Hu, Jon Gauthier, Peng Qian, Ethan Wilcox, and Roger Levy. 2020. A systematic assessment of syntactic generalization in neural language models. In *Proceedings of the Association of Computational Linguistics*.

Rafal Jozefowicz, Oriol Vinyals, Mike Schuster, Noam Shazeer, and Yonghui Wu. 2016. Exploring the limits of language modeling. *arXiv preprint arXiv:1602.02410*.

Jey Han Lau, Alexander Clark, and Shalom Lappin. 2017. Grammaticality, acceptability, and probability: A probabilistic view of linguistic knowledge. *Cognitive Science*, 5:1202–1247.

Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. 2016. Assessing the ability of LSTMs to learn syntax-sensitive dependencies. In *Transactions of the Association for Computational Linguistics*, volume 4, pages 521–535.

Rebecca Marvin and Tal Linzen. 2018. Targeted syntactic evaluation of language models. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1192–1202, Brussels, Belgium. Association for Computational Linguistics.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9.

Marten van Schijndel, Aaron Mueller, and Tal Linzen. 2019. Quantity doesn't buy quality syntax with neural language models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5835–5841.

Yikang Shen, Shawn Tan, Alessandro Sordoni, and Aaron Courville. 2019. Ordered neurons: Integrating tree structures into recurrent neural networks. In *International Conference on Learning Representations*.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R Bowman. 2020. BLiMP: A benchmark of linguistic minimal pairs for English. In *Proceedings of the Society for Computation in Linguistics*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.