Transformer-Maze: An Accessible Incremental Processing Measurement Tool

Annika Heuser (aheuser@sas.upenn.edu)
Department of Linguistics, University of Pennsylvania
Philadelphia, PA, USA

Edward Gibson (egibson@mit.edu)
Department of Brain & Cognitive Sciences, Massachusetts Institute of Technology
Cambridge, MA, USA

Abstract

The lesser known G(rammatical)-Maze task (Forster, Guerrera, & Elliot, 2009) is arguably a better choice than self-paced reading (Mitchell, 2004) for detecting difficulty from word to word in online sentence processing over crowdsourcing platforms. In G-Maze, a participant must choose between each successive word in a sentence and a distractor word that does not make sense based on the preceding context. If a participant chooses the distractor as opposed to the actual word, then the trial ends and they may not complete the sentence. Thus, G-Maze automatically filters out data from inattentive participants, and more effectively localizes differences in processing difficulty. Still, the effort required to pick contextually inappropriate distractors for hundreds of words might cause an experimenter to hesitate before picking this method. To save experimenters this time and effort, Boyce, Futrell, and Levy (2020) developed A(uto)-Maze, a tool that automatically generates distractors using a computational language model. We now introduce the next generation of A-Maze: T(ransformer)-Maze. Transformer models are the current state of the art in natural language processing, and thousands, pretrained in a variety of languages, are freely available on the internet, specifically through Huggingface's Transformers package (Wolf et al., 2020). In our validation experiment, T-Maze proves itself to be as effective as G-Maze with handmade materials, run in a lab. This tool thus allows psycholinguists to easily gather high-quality online sentence processing data in many different languages.

Keywords: Online sentence processing; Methods; Machine-learned language models; Syntactic ambiguity

Introduction

Structure, meaning, attention, and memory are critical to online sentence processing and production. Therefore, plausible theories based on their observation in speakers of various languages can have far-reaching implications for linguistics and cognitive science. Characteristics of sentence comprehension make it easier to study than sentence production. Sentence comprehension is incremental: new linguistic information, whether the next phoneme or the next word, must be integrated into our understanding from the previous time step. Due to limited computational resources, the integration cost differs based on the context and properties of the new information. An increased integration cost is typically paid with more time, at the millisecond scale. By measuring how reading times change from word to word and sentence to sentence, researchers capture concise snapshots of online language comprehension and its computational constraints (Gibson & Pearlmutter, 1998; Tanenhaus & Trueswell, 1995; Bartek, Lewis, Vasishth, & Smith, 2011).

The most prevalent methods for collecting reading measures of processing difficulty are eye tracking (Rayner, 1998) and moving-window self-paced reading (SPR; Mitchell, 2004). In eye-tracking, as the name suggests, an infrared camera *tracks* the movements of a participants eyes while they read a sentence projected onto a screen. It is relatively expensive because of the specialized equipment it requires. However, it results in high-quality, though complex, data, with several dependent measures to analyze. SPR, on the other hand, only has one dependent measure: a word's reading time (RT). In SPR, all but one word in a sentence are masked and the participant presses a button to re-mask the current word and reveal the next one. The time between button presses, during which a word is legible, is that word's RT. In an attempt to ensure that a participant does not mentally check out while clicking through a sentence, each sentence is typically followed by a comprehension question. However, these questions can often be answered based on world knowledge rather than the content of the sentence, so that an inattentive participant can often guess correctly. Additionally, analyses of SPR data often reveal what are known as "spillover effects": greater than expected RTs immediately following the anticipated source of the processing difficulty. Nonetheless, SPR's greatest advantage only became clear over the last decade: that researchers can run SPR experiments over crowdsourcing platforms (Enochson & Culbertson, 2015) like Amazon's Mechanical Turk (Paolacci, Chandler, & Ipeirotis, 2010) and Prolific (Palan & Schitter, 2018). Researchers can cheaply and quickly recruit much more diverse participant pools using crowdsourcing platforms.

But a potential problem is that unsupervised participants paid per task are likely to optimize for speed, which means skimming for SPR. Because the comprehension question accuracy might not allow researchers to effectively filter out skimming participants, SPR can lose power over crowdsourcing platforms. In fact, a study confirmed this: the estimated power based on an SPR experiment run in-lab was greater for 2 out of 3 effects than the estimated power based on the same experiment run on Mechanical Turk (Boyce et al., 2020). While an experimenter could then simply recruit more participants for an online study, this raises the cost of running the experiment. More crucially, if we have no estimate of effect size for a potential processing phenomena, we do not know how many more online participants we would need to recruit

In M. Goldwater, F. K. Anggoro, B. K. Hayes, & D. C. Ong (Eds.), *Proceedings of the 45th Annual Conference of the Cognitive Science Society*.

to capture the potential effect. If we recruit too few for this initial experiment, we would then either need to run a second, more expensive pilot experiment or we may incorrectly conclude that there is no such processing effect. Increasing the power of online language processing experimental methods would then reduce experimental costs and increase the field's overall research output.

The Maze task (Forster et al., 2009) is another method for measuring incremental processing time differences that can be run on a crowdsourcing platform. A participant must choose between each successive word in a sentence and a distractor word. In L(exical)-Maze, the distractor words are nonwords while in G(rammatical)-Maze, they are real words that do not make sense in the preceding context of the sentence. The G-Maze task, especially, is harder for an inattentive participant to complete. This is because a skimming participant is much more likely to pick a distractor, after which they are not allowed to finish the sentence. The experiments of Witzel, Witzel, and Forster (2012) confirms this. They found that on the same experiment conducted in-lab, G-Maze more effectively detected an effect of greater processing difficulty at the critical region than L-Maze. Witzel et al. (2012) also compared L-Maze and G-Maze to eye-tracking and SPR and found both versions of Maze to produce localized (i.e. fewer spill-over effects) and robust effects relative to eye-tracking and SPR.

While G-Maze enables researchers to collect higher quality data online, choosing good distractors that obviously do not fit the context can be difficult and time-consuming. Boyce, Futrell, and Levy (2020) therefore developed A(uto)-Maze to automatically generate distractors. A-Maze uses a computational language model to determine which words are the least likely given the sentence's preceding context. Boyce et al. (2020) tested two versions of A-Maze, both based on recurrent neural network (RNN) models, on Mechanical Turk. Both versions of A-Maze were better than SPR at detecting differences in processing difficulty at the expected sentence region.

A-Maze produced materials in English for the validation experiments. Producing materials with A-Maze in other languages requires RNNs trained in those languages. If an RNN trained on the appropriate language to a sufficiently high standard is not available on the internet, and a researcher otherwise does not have access to one, then they will have to train one themselves. Achieving high performance requires substantial computational resources as well as expertise. Training an RNN is therefore not an option for every researcher that would be interested in using A-Maze. Moreover, RNNs have been supplanted by Transformer models as the state of the art in natural language processing. Thousands of pretrained Transformer models, trained in a variety of languages, are freely available on the internet. They are more accessible to researchers with limited computational resources.

We therefore developed T(ransformer)-Maze, the next generation of A-Maze, to generate distractors based on Transformer models. Any researcher can adapt T-Maze to a new language, assuming it is one of the hundreds of languages represented by the models in HuggingFace's Transformers (Wolf et al., 2020) package. We are excited for this tool to promote online sentence processing research in a variety of languages.

Transformer-Maze

To pair a word with a distractor, a set of candidates that approximately match the word in length and frequency are compiled. The user defines how large this set is, allowing them to adjust performance and computational load to their experiment and system setup. The set of potential distractors are quickly collected from a pre-made language-specific dictionary of words sorted by frequency. This dictionary can be adjusted to the user's experimental materials. For example, because our sentences did not contain any abbreviations or interjections, a participant might come to recognize them as distractors without needing to consider the preceding context of the sentence. We therefore removed abbreviations and interjections from the dictionary of potential distractors before generating our materials.

Next, the set of potential distractors are scored. A traditional language model, like the RNNs with which A-Maze was tested, scores a potential distractor by assigning it a conditional probability, given the sentence's preceding context. We pick the distractor with the lowest conditional probability, because it, by definition, is the least likely to occur after the sentence's preceding context. A participant, having read the preceding context, should then be able to easily tell it apart from the actual next word, which almost certainly has a higher conditional probability.

Many high performing Transformer models are not traditional language models. Rather, they are masked language models (MLMs). As opposed to predicting the next word based just on the preceding context, an MLM predicts a word based on the words following it in the sentence as well. Computing a score for how likely a given word is given the preceding context is much less straightforward with an MLM. We employ a Python package by Salazar, Liang, Nguyen, and Kirchhoff (2019) to assign potential distractors pseudo-log-likelihood (PLL) scores based on an MLM. Salazar et al. (2019) found that PLL scores could effectively predict which of two sentences is more acceptable according to human judgements. The MLMs they tested all outperformed GPT-2 (a traditional language modeling Transformer) on the Benchmark of Linguistic Minimal Pairs (BLiMP) (Warstadt et al., 2020). Therefore, even researchers who have access to a traditional language modeling Transformer trained on their language of interest, may find that plugging a masked language modeling Transformer into T-Maze will give them better distractors.

Because the scores from different types of Transformer models are not comparable, our algorithm does not have a threshold score, above which a potential distractor is sufficiently unlikely in the context, to automatically choose it as

Table 1: Example stimuli for each condition. The disambiguating words are italicized.

Relative clause - Low attachment: (1a) The niece of the fisherman who got *himself* a sailboat learned to sail.
Relative clause - High attachment: (1b) The niece of the fisherman who got *herself* a sailboat learned to sail.
Adverb clause - Low attachment: (2a) Robert will meet the friend he phoned *yesterday*, but he doesn't want to.
Adverb clause - High attachment: (2b) Robert will meet the friend he phoned *tomorrow*, but he doesn't want to.
Sentence vs noun phrase (S v NP) coordination - With comma: (3a) The crowd cheered for the model, and the designer *took* a bow after the show
Sentence vs noun phrase (S v NP) coordination - No comma: (3b) The crowd cheered for the model and the designer *took* a bow after the show.

the distractor. Instead, we must evaluate every potential distractor in the set and then choose the one with the best score. Therefore, on average, our algorithm will run longer than A-Maze, which does have a threshold. However, by evaluating every distractor, we can find the best, second best, third best, etc. distractor in the whole set of potential distractors. We allow the user to specify how many top distractors they would like to save. For example, if they set this parameter to 5, then T-Maze saves the top 5 distractors for each word in each sentence. Then, if the user finds an implausible distractor in their materials, they have the option of replacing it with the distractor with the second best score, as opposed to potentially needing to re-run the entire algorithm.

Validation Experiment

Procedure

In T-Maze's maiden voyage, we replicated the experiment Boyce et al. (2020) used to demonstrate A-Maze's efficacy. The sentence structures set up 3 types of syntactic ambiguity: 1) relative clause attachment ambiguity, 2) adverb attachment ambiguity, and 3) sentence (S) versus noun phrase (NP) coordination ambiguity. Table 1 contains examples of each type of ambiguity. Each sentence corresponds to a condition. Based on the results of Boyce et al. (2020), the (a) sentence types should be easier for native English speakers to process. We therefore generally expect the low attachment and comma conditions to have lower mean RTs.

Like Boyce et al. (2020), we matched distractors across the two sentences in the same item, so (1a) and (1b) in table 1 would have the exact same distractors. This eliminates a potential confound of different RTs across the critical region being in part due to differences in the distractors. Before the critical region, the sentences consist of the same words in the same context. The algorithm described in the previous section produces the same distractors for these words without any kind of adjustment. However, the disambiguating words must, of course, differ across the items to result in the different conditions. For example, (2a)'s disambiguating word (italicized) is "yesterday" while (1b)'s is "tomorrow." These words differ in both their length ("yesterday" has 9 characters while "tomorrow" has 8) and their frequencies. We therefore take the average of their lengths and frequencies to determine which potential distractors to collect, such that they match both disambiguating words equally. The preceding context of the sentences still match, so our evaluation procedure does not change. After the critical region, the words of the two sentences are the same again but now the preceding contexts differ. We therefore compute a potential distractor's scores for both contexts and average them. We choose the distractor with the best average score across both contexts.

We generated the distractors for our validation experiment with the `bert-base-uncased` model (Devlin, Chang, Lee, & Toutanova, 2018) in order to test the PLL scoring scheme. We argue that this model gives us an estimate of T-Maze's baseline performance because it has the fewest parameters of the models Salazar et al. (2019) tested on BLIMP. The general trend with Transformers that Salazar et al. (2019) also observed with PLLs is that the greater the number of parameters, the greater the performance. We evaluated 100 potential distractors for each word. We chose this parameter to match A-Maze, which evaluates 100 distractors if none of them meet the threshold (Boyce et al., 2020). Evaluating more potential distractors will result in chosen distractors that have lower PLL scores, which should correspond to distractors that are less acceptable in the given context. This parameter choice then also allows us to estimate T-Maze's baseline performance. We did not check the quality of our distractors after they were generated.

We chose to estimate the baseline performance as opposed to trying to achieve the best performance possible because we want to guarantee sufficient performance for researchers without many computational resources. If a researcher has the resources to use the best possible Transformer model and to evaluate more than 100 potential distractors, T-Maze is likely to produce better distractors than those used for this experiment. However, the best performing Transformer models are often not open-source. GPT-3 (Brown et al., 2020) is an example of such a model. Additionally, using a larger model to evaluate a greater number of distractors, will at the very least take more time, during which the researcher may not be able to do other intensive computational work. Therefore, an estimate of baseline performance also helps researchers with these resources determine how to best allocate them.

We hosted our experiment on PCIbex Farm (Zehr & Schwarz, 2018) and recruited 50 participants on Prolific

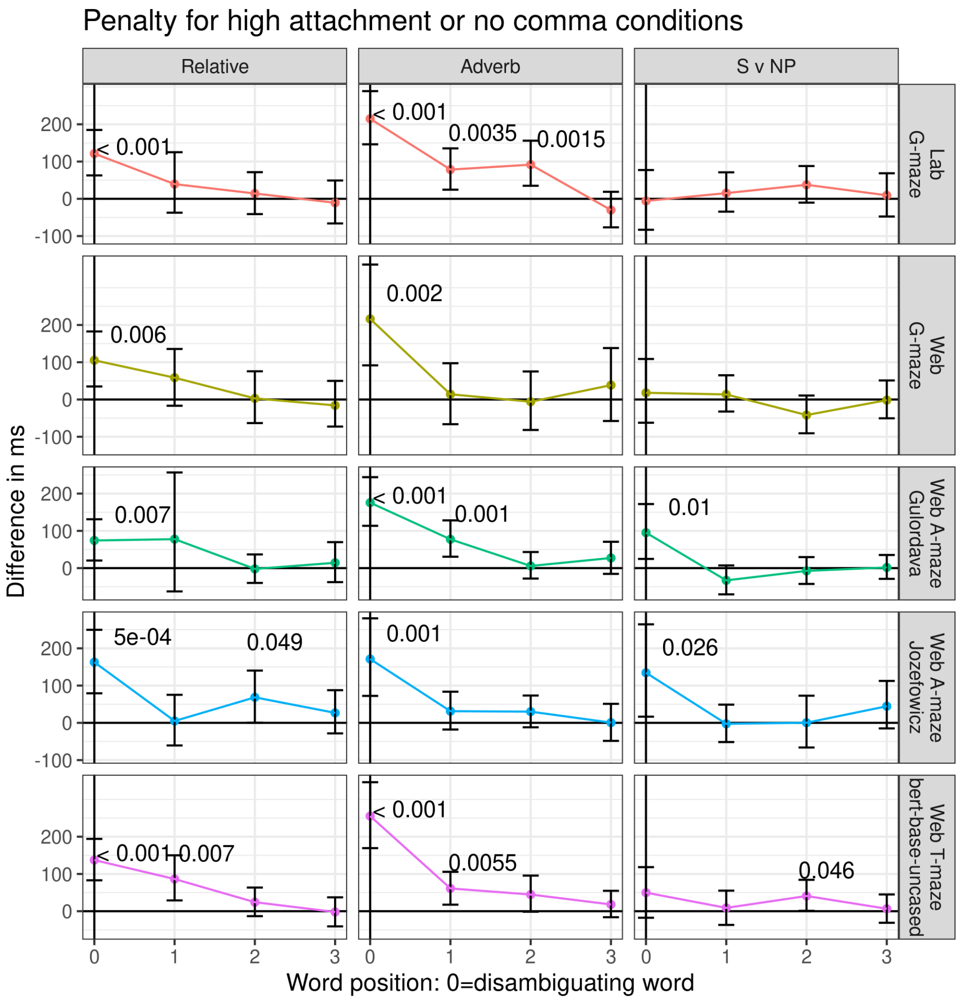

Figure 1: Estimated effect sizes with error bars indicating the 95% confidence intervals and p-value equivalents when $p < 0.05$. We include Boyce et al. (2020)'s data for comparison.

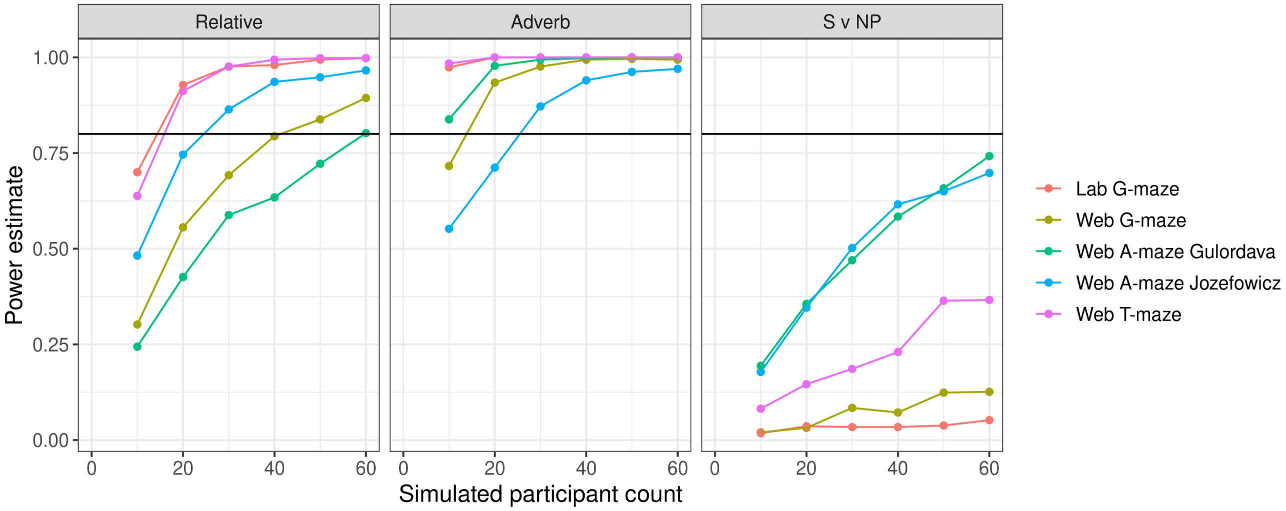

Figure 2: Estimated power for different numbers of participants based on observed data from different methods

(Palan & Schitter, 2018). However, due to some technical difficulties, we were only able to collect data from 49 participants. After working through 8 practice sentences, participants worked through 24 sentences of each ambiguity type mixed in with 24 fillers (96 total). Refer to the following PCIbex Farm demonstration for a complete picture of the experimental setup: `https://farm.pcibex.net/r/PFuPTr/`.

Results

If a participant makes a mistake, the trial ends at that word, and we cannot collect that participant's data for the rest of the sentence. We removed 2.3% of the data for being mistakes and 15% for being blank because of a mistake earlier in the sentence. This left us with 83%.

We repeated the same analysis conducted by Boyce et al. (2020) in order to more directly compare our results with theirs. We re-ran the same analysis on the A-Maze data collected by Boyce et al. (2020) and the data from the in-lab experiments conducted by Witzel et al. (2012). All this data is included in the A-Maze Github repository of Boyce et al. (2020) (`https://github.com/vboyce/Maze`). In figure 1, we compare T-Maze's estimated effect sizes of each type of attachment ambiguity with the methods Boyce et al. (2020) found to be the most effective. Lab G-Maze refers to the G-Maze experiment run in-lab by Witzel et al. (2012). Web G-Maze refers to the experiment run by Boyce et al. (2020) on Mechanical Turk with the same hand-made materials used in Lab G-Maze. The names of the two versions of A-Maze refer to the studies from which the RNN models come from: Gulordava, Bojanowski, Grave, Linzen, and Baroni (2018) and Jozefowicz, Vinyals, Schuster, Shazeer, and Wu (2016).

Note that to our knowledge, there is not yet a straightforward way to evaluate and compare distractor quality across multiple experiments. Distractors are good enough if 1) they can reveal known effects and 2) if an attentive participant finds them easy to discern from the actual words, while an inattentive participant is guessing at chance. Factors irrelevant to the distractor-generation method, like the quality of the underlying sentences, can obscure an evaluation of 1. A complete evaluation of 2 relies on researchers having access to the truth value of whether any given participant is paying attention at any given time. Because it is impossible to have access to this kind of information, we cannot be sure whether a participant chose a distractor because they were not paying attention, or because it was difficult to discern from the actual word. First, we discuss the T-Maze effect sizes and stimulated power across the 3 types of syntactic ambiguity in comparison to other methods like A-Maze to determine the likelihood of 1 being the case for the distractors that T-Maze generates.

T-Maze revealed large localized effects as well as lab- and web-based G-Maze and both versions of A-Maze. In figure 1, a localized effect is one at the 0th word position, where the ambiguity is resolved. Effects after this position could be considered spill-over effects, however they are not problematic as long as the greatest effect is seen at the disambiguating word. The power estimates displayed in figure 2 also demonstrates that T-Maze is as powerful as Lab G-Maze.

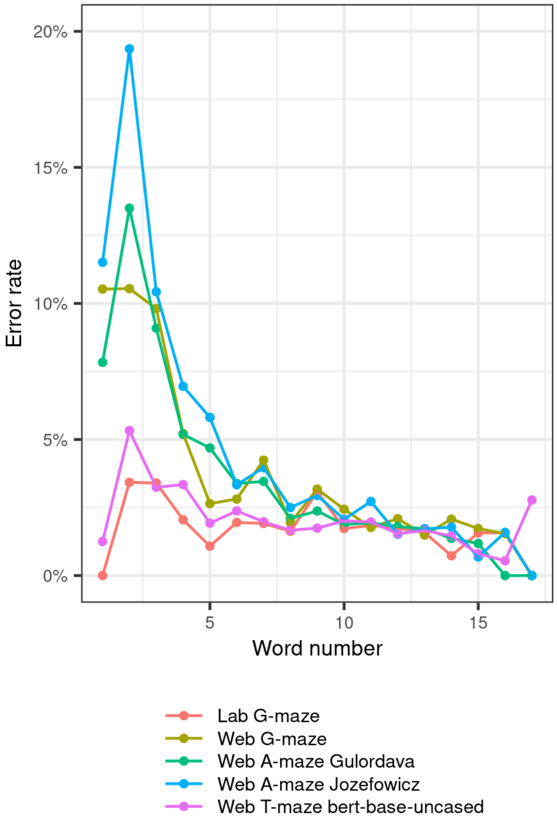

Figure 3: Participant error rate at each word position, where word 1 is the first word in the sentence (always paired with the distractor "x-x-x"). Lab G-Maze participants could not make an error at word 1 because they simply pushed a button to continue to word 2.

In both the relative clause and adverb attachment ambiguity conditions, T-Maze's results and estimated power were comparable to those of both A-Maze versions. However, both A-Mazes found an effect for the sentence (S) v noun phrase (NP) coordination condition, while T-Maze did not. Critically, Lab G-Maze also failed to find an effect, and the A-Maze effect sizes were much smaller than the effect sizes of any of the methods for the other conditions. This suggests that the inconsistency in finding an effect may have had to do with the original S v NP sentences as opposed to with the distractor-generation method. We nonetheless investigated whether the lack of an effect might have been due to distractors produced by T-Maze for the S v NP sentences appearing difficult to discern from the actual words.

How difficult a distractor is to discern from the actual word affects how much time a participant will take to select a word. Uncertainty about which word is the distractor will slow participants down, which can confound the delay of the dispreferred condition's greater processing difficulty. In addition to

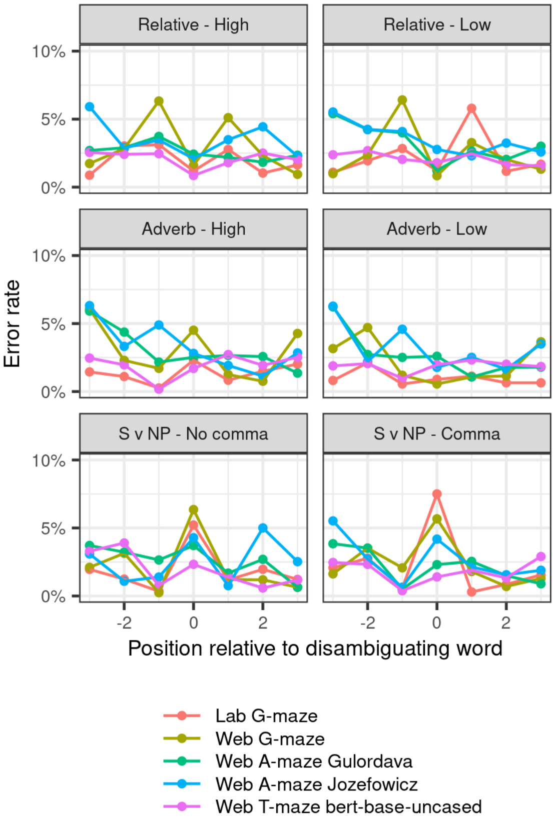

Figure 4: Error rates for each condition's critical region. Position 0 refers to the disambiguating word/critical region. Negative positions refer to words preceding the disambiguating word in the sentence, and positive positions to words following it. T-Maze tends to have some of the lowest error rates, especially at the critical regions.

taking more time to decide which word is the distractor, participants are also more likely to pick the distractor as opposed to the actual word. Therefore, a higher error rate may be an indicator of poorer quality distractors that might be masking the processing difficulty difference between the S v NP Comma and No Comma conditions.

However, the error rate is also a function of the participants' attentiveness. We cannot control for attentiveness, and we want to filter out data from when a participant is not paying attention. Therefore, while a low error rate indicates that our participants can easily tell the distractors apart from the actual words, a lower error rate is not necessarily always better. We have reason to believe that our participants were more attentive than those of Boyce et al. (2020) because Prolific has been found to produce higher quality data for online behavioral studies than Mechanical Turk (Peer, Rothschild, Gordon, Evernden, & Damer, 2021).

Then, to better gauge whether the distractor quality may have interfered with the detection of an S v NP coordination ambiguity effect, we looked to the error rate of each condition individually in figure 4.Consistent with figure 3, the Prolific participants who saw the T-Maze distractors had some of the lowest error rates across the different conditions. The S v NP Comma and No Comma conditions were no exception. In fact, at the critical region, the T-Maze participants had the lowest error rate of all the methods. This is evidence that T-Maze is not to blame for the lack of an S v NP effect.

Contributions

By automating distractor pairing via a sequential language model, A-Maze removed one of the greatest hurdles for researchers wanting to run G-Maze experiments–the time and effort required to think of good distractors for hundreds of words. T-Maze makes designing G-Maze experiments easier still, because it allows researchers to plug in transformers, the current NLP state of the art. These models, pretrained on several languages, are available online, making them accessible to researchers who do not have many computational resources at their disposal. We designed T-Maze itself to be just as accessible: we produced the materials for our experiment on a Google Colab notebook in about half an hour.

Through our validation experiment, we demonstrated that T-Maze is as effective as G-Maze run in a lab at localizing differences in processing difficulty due to syntactic attachment ambiguity. T-Maze's baseline performance is also on par with A-Maze's. T-Maze, however, is by nature easier to adjust to new languages. It also provides researchers with greater freedom: in addition to having more freely available models to plug in, a research can easily change the number of distractors evaluated via a parameter. We will make T-Maze as easy for other researchers to use as possible, as an open-source python package accompanied by thorough documentation. We hope that T-Maze enables and encourages more labs to collect the empirical evidence they need to develop and test theories of online language comprehension.

Code Repository

The T-Maze generation code and our experimental materials are available at `https://github.com/annikaheuser/TMaze`.

References

Bartek, B., Lewis, R. L., Vasishth, S., & Smith, M. R. (2011). In search of on-line locality effects in sentence comprehension. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *37*(5), 1178.

Boyce, V., Futrell, R., & Levy, R. P. (2020). Maze made easy: Better and easier measurement of incremental processing difficulty. *Journal of Memory and Language*, *111*, 104082. doi: https://doi.org/10.1016/j.jml.2019.104082

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., . . . others (2020). Language models are few-shot learners. *Advances in neural information processing systems*, *33*, 1877–1901.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Enochson, K., & Culbertson, J. (2015, 03). Collecting psycholinguistic response time data using amazon mechanical turk. *PLOS ONE*, *10*(3), 1-17. Retrieved from `https://doi.org/10.1371/journal.pone.0116946` doi: 10.1371/journal.pone.0116946

Forster, K. I., Guerrera, C., & Elliot, L. (2009). The maze task: Measuring forced incremental sentence processing time. *Behavior research methods*, *41*(1), 163–171.

Gibson, E., & Pearlmutter, N. J. (1998). Constraints on sentence comprehension. *Trends in cognitive sciences*, *2*(7), 262–268.

Gulordava, K., Bojanowski, P., Grave, E., Linzen, T., & Baroni, M. (2018). Colorless green recurrent networks dream hierarchically. *arXiv preprint arXiv:1803.11138*.

Jozefowicz, R., Vinyals, O., Schuster, M., Shazeer, N., & Wu, Y. (2016). Exploring the limits of language modeling. *arXiv preprint arXiv:1602.02410*.

Mitchell, D. C. (2004). On-line methods in language processing: Introduction and historical review. In *The on-line study of sentence comprehension* (pp. 15–32). Psychology Press.

Palan, S., & Schitter, C. (2018). Prolific. ac—a subject pool for online experiments. *Journal of Behavioral and Experimental Finance*, *17*, 22–27.

Paolacci, G., Chandler, J., & Ipeirotis, P. G. (2010). Running experiments on amazon mechanical turk. *Judgment and Decision making*, *5*(5), 411–419.

Peer, E., Rothschild, D., Gordon, A., Evernden, Z., & Damer, E. (2021). Data quality of platforms and panels for online behavioral research. *Behavior Research Methods*, 1–20.

Rayner, K. (1998). Eye movements in reading and information processing: 20 years of research. *Psychological bulletin*, *124*(3), 372.

Salazar, J., Liang, D., Nguyen, T. Q., & Kirchhoff, K. (2019). Masked language model scoring. *arXiv preprint arXiv:1910.14659*.

Tanenhaus, M. K., & Trueswell, J. C. (1995). Sentence comprehension.

Warstadt, A., Parrish, A., Liu, H., Mohananey, A., Peng, W., Wang, S.-F., & Bowman, S. R. (2020). Blimp: The benchmark of linguistic minimal pairs for english. *Transactions of the Association for Computational Linguistics*, *8*, 377–392.

Witzel, N., Witzel, J., & Forster, K. (2012). Comparisons of online reading paradigms: Eye tracking, moving-window, and maze. *Journal of psycholinguistic research*, *41*(2), 105–128.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., … Rush, A. (2020, October). Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: System demonstrations* (pp. 38–45). Online: Association for Computational Linguistics. Retrieved from `https://aclanthology.org/2020.emnlp-demos.6` doi: 10.18653/v1/2020.emnlp-demos.6

Zehr, J., & Schwarz, F. (2018). Penncontroller for internet based experiments (ibex). *DOI: https://doi.org/10.17605/OSF. IO/MD832*.