Maze Made Easy: Better and easier measurement of incremental processing difficulty

Veronica Boyce[1], Richard Futrell[2], and Roger P. Levy[1]

vboyce@mit.edu, rfutrell@uci.edu, rplevy@mit.edu
[1]Department of Brain and Cognitive Sciences, Massachusetts Institute of Technology
[2]Department of Language Science, University of California, Irvine

July 1, 2019

Abstract

Behavioral measures of incremental language comprehension difficulty form a crucial part of the empirical basis of psycholinguistics. The two most common methods for obtaining these measures have significant limitations: eye tracking studies are resource-intensive, and self-paced reading can yield noisy data with poor localization. These limitations are even more severe for web-based crowd-sourcing studies, where eye tracking is infeasible and self-paced reading is vulnerable to inattentive participants. Here we make a case for broader adoption of the Maze task, involving sequential forced choice between each successive word in a sentence and a contextually inappropriate distractor. We leverage natural language processing technology to automate the most researcher-laborious part of Maze – generating distractor materials – and show that the resulting A(uto)-Maze method has dramatically superior statistical power and localization for well-established syntactic ambiguity resolution phenomena. We make our code freely available online for widespread adoption of A-maze by the psycholinguistics community.

1 Introduction

One of the major questions in the cognitive science of language is how comprehension unfolds in real time. A key part of the empirical landscape is that processing difficulty is DIFFERENTIAL and LOCALIZED: some parts of a linguistic input are more effortful and time-consuming than others. In the field of sentence processing, researchers can gain insight into this differential and localized difficulty by measuring word-by-word patterns of reading behavior, which turn out to capture highly incremental linguistic processing, reflecting not only the bottom-up characteristics of the word currently being read, but also that word's relation to the context in which it appears (Frazier and Rayner, 1982; MacDonald, 1993). These word-by-word patterns, measured at the millisecond scale, enable the development and testing of detailed, computationally explicit theories of real-time language understanding (Grodner and Gibson, 2005; Staub, 2010; Bartek et al., 2011; Smith and Levy, 2013). Experimental methods that efficiently capture this incremental processing, are cheap and easy to deploy, and yield easy-to-analyze data are thus of considerable scientific value.

To date, the two most widely used methods of obtaining behavioral data on reading measures are eye tracking (Rayner, 1998) and self-paced reading (Mitchell, 1984). In eye tracking, a participant's eye movements are monitored with an infrared camera during reading of on-screen material. This method yields high-quality data but requires expensive equipment with a human operator and sometimes non-trivial data post-processing. Self-paced reading, in which a sentence starts off masked and an experimental participant presses a button to reveal each successive word and mask the previous word, with the time between button presses constituting the word's READING TIME (RT), is technically simpler. However, self-paced reading typically yields poorer temporal resolution, with processing difficulty effects often not showing up in RTs on the word of origin but instead "spilling over" some number of words downstream; it is also vulnerable to inattentive participants.

Within the past decade, dramatic new possibilities for data collection in experimental psychology have opened up with the advent of "crowd-sourcing" web services such as Mechanical Turk (Paolacci and Chandler, 2014) and Prolific (Peer et al., 2017), allowing large-scale recruitment of diverse populations with access to the World Wide Web. Experimental psycholinguistics today makes extensive use of crowd-sourcing for data collection, including the use of self-paced reading for measuring RTs (e.g., Enochson and Culbertson, 2015). Here we present a study using a less-widely-used method, the Maze task (Forster et al., 2009; Freedman and Forster, 1985), for crowd-sourced web experiments on incremental language processing. We find in this setting that the Maze task shows high sensitivity – far more than self-paced reading – for detecting processing difficulty differences evoked by structural ambiguity resolution. We further remove some critical barriers to the adoption of the Maze task by introducing an automatic method to eliminate a great deal of experimenter effort in designing task stimuli.

The remainder of the paper is structured as follows. In Section 2, we review methods for measuring incremental processing difficulty: self-paced reading, eyetracking, and Maze. In Section 3, we introduce our new variant of the Maze task, which we call A(uto)-maze, where distractor items are generated automatically using state-of-the-art natural language processing (NLP) technologies. In Section 4, we validate A-maze and previous Maze variants in a web-based format, replicating results from Witzel et al. (2012) over Amazon Mechanical Turk and using our A-maze system. Section 5 concludes.

2 Behavioral methods for measuring incremental processing difficulty

In the realm of human language understanding, one set of methods focus on measuring real-time processing effects, by tracking how long participants spend on each word as they read a sentence. These reading or reaction times (RTs) can be interpreted as indicative of how hard the words are to process; long RTs indicate some form of difficulty. Two methods dominate this area: eye tracking and self-paced reading.

2.1 Eye tracking

With eye tracking, participants freely read sentences on a screen while their eye-movements are recorded by an infrared camera. Eye movements are saccadic (consisting of sequences of fixations typically 200–300ms in duration connected by rapid ∼30ms saccades) and unconstrained, so several widely used dependent measures have been developed to analyze them, including whether a word is skipped, how long the eyes spend on the word the first time it is fixated, whether the first saccade out of a word is progressive or regressive, total looking time to a word, and how long until the participant moved on to the next word (Rayner, 1998). In general, greater processing difficulty is manifested in lower skip rates, longer looking times, and higher probability of a regressive saccade after fixating a word; it is well documented that both a word's fixed features, such as its length and frequency, and features of the word's relation with its context, such as its contextual predictability and whether it is grammatically or semantically anomalous given the context, affect these eye movement measures (Rayner et al., 2004), though different features can affect eye movements in different ways (e.g., Staub, 2011). One advantage of eye tracking is that unconstrained reading is a natural everyday activity for literate participants; however, this means participants are free to skim, jump ahead, or look back while reading (Witzel et al., 2012), which can offer challenging analytic and interpretive decisions for researchers (von der Malsburg and Angele, 2017).

Because of the equipment required for eye tracking, these experiments have to be done in laboratory settings under the supervision of researchers. This makes these experiments costly in time and money spent recruiting and running participants, and means that participant pools skew towards undergraduate psychology majors.

2.2 Self-paced reading

The other commonly used incremental processing method is self-paced reading (SPR). In moving-window SPR (the most common version), participants read a sentence one word at a time, pressing a button (e.g., the space bar on a computer keyboard) to mask the current word and unmask the next one. The time between

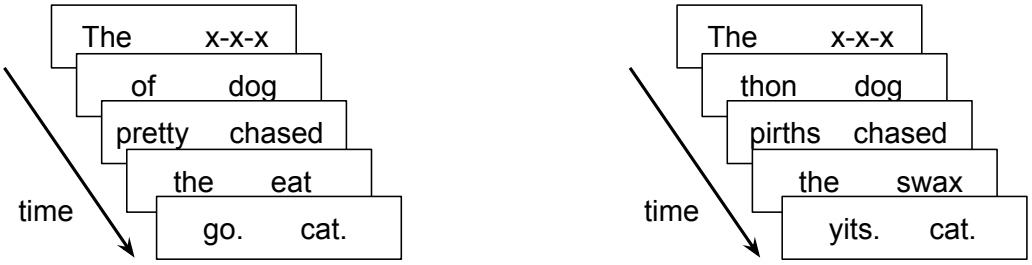
button presses (i.e. the time a word was visible) is used as the dependent measure. This method forces participants to read sentences sequentially, with no looking ahead or looking back; however, participants can continue processing a word as they look at later words. This can lead to "spillover" effects, where the difficulty induced by a given word slows RTs one or more words further downstream and may not manifest at all on the word in question (e.g., Mitchell, 1984; Koornneef and van Berkum, 2006; Smith and Levy, 2013). To compensate for this, SPR is often analysed using a multi-word spillover region, which works if the location of potential slow-down is known, but not if pinpointing the slow-down is the goal of the experiment. To encourage more careful reading, SPR (and eye tracking) can be paired with comprehension questions.

One of the advantages of SPR is that it can be run over the web with participants recruited from crowd-sourcing platforms, which leads to quick and cheap data collection. Crowdsourcing websites such as Amazon Mechanical Turk allow researchers to recruit and pay participants for doing small tasks, provided the task can be explained and administered through a web browser. SPR and other tasks that involve seeing stimuli and pressing buttons are easy to do in this environment, and the time between button presses can be measured precisely (Enochson and Culbertson, 2015). In addition, the participant pool from online platforms may be more representative of the general population than the participant pools available for in-person experiments at research universities (Casler et al., 2013), though these pools are still not completely representative of the societies from which they are drawn (Difallah et al., 2015). For crowd-sourced populations there are also questions as to the quality of data relative to in-lab experimental data. For some tasks, crowd-sourced data seem to be at least as high-quality as in-lab data (Casler et al., 2013). For self-paced reading, at least some studies have shown similar results in crowd-sourced populations with web-based methods and in-lab populations (Enochson and Culbertson, 2015). However, Enochson and Culbertson (2015) also found that web responses were on average 180ms faster than lab responses (and our unpublished data suggest similar results), perhaps due to participants' strong incentives to finish quickly. This raises the concern that crowdsourced participants might read less carefully than in-lab participants, leading to more superficial language understanding that might mask theoretically important comprehension processes.

2.3 Maze

A third incremental processing method that is used less often is the Maze task (Forster et al., 2009). As pictured in Figure 1, the Maze task has participants read a sentence word by word, but at each word position they are presented with a forced choice: between a correct word that serves as a legitimate continuation of the sentence and a distractor that does not. Participants must press a button corresponding to the correct word, and reaction time (RT) is used as the dependent measure. If the participant chooses the correct word, the trial continues with another Maze step involving a choice between the correct next word of the sentence and a distractor; if the participant chooses the wrong word, the trial is terminated and no further words in the sentence are shown. We are aware of two versions of the Maze task that have been tested: G(rammaticality)-maze, which uses real word distractors that are anomalous given the context, and L(exicality)-maze, which uses nonce word distractors. At least a dozen papers have been published using the Maze method (Qiao et al., 2012; O'Bryan et al., 2013; Kizach et al., 2013; Wang, 2015; Nyvad et al., 2015; Witzel and Witzel, 2016; Oliveira et al., 2017; Sikos et al., 2017; Li et al., 2017; Suzuki and Sunada, 2018), but this is tiny compared to the number of studies that use SPR and eye-tracking. While most uses of Maze have been to test sentence processing theories, the Maze task has also been used as a pedagogical tool for second language acquisition (Enkin, 2012). As far as we are aware, only Witzel et al. (2012) and Witzel and Forster (2014) have directly compared the sensitivity of L-maze and G-maze to that of SPR or eye-tracking. In a comparison of eye tracking, SPR, G-maze, and L-maze, Witzel et al. (2012) found that G-maze showed a localized effect that eye tracking did not in one of the three conditions, and that G-maze generally had larger and more localized effects than SPR. In another study, Witzel and Forster (2014) similarly found that G-maze has more localized effects than eye-tracking. Both studies found that G-maze had larger and more localized effects than L-maze.

However, G-maze has to date been much more laborious to construct materials for than L-maze: whereas L-maze simply requires that the distractor is a letter sequence that does not constitute a legitimate word in the vocabulary, a process that can easily be automated, G-maze requires that for each word in each experimental sentence, a distractor word be chosen from the vocabulary that cannot be integrated into the preceding context to continue the sentence, a process that to date has required manual work by the



(a) Sample G-maze

(b) Sample L-maze

Figure 1: In Maze tasks, participants see two words at a time and have to select the word that continues the sentence. They then see the next pair of words.

experimenter and that is potentially error-prone.

Like SPR and unlike eye tracking, the Maze task does not require special equipment; all it needs is a way of displaying stimuli and recording button-presses, so it should be amenable to running over the web. Given that Maze seems to be an effective method, we want to make it a more appealing option by making it easier to prepare materials and run on a large, crowd-sourced participant pool.

Here we introduce two innovations to the Maze paradigm and then validate them on the materials from Witzel et al. (2012). First, we set up Maze to run over the web, enabling it to be run on crowd-sourced participants. Second, we use contemporary machine-learning language models to automatically generate real word distractors, offering a lower-preparation-cost version of G-maze that we call Auto-maze (A-maze). We validate these methods by running A-maze along with G-maze, L-maze, and SPR on Mechanical Turk participants, using the materials of Witzel et al. (2012), a paper which compared in-lab SPR, eye tracking, and L- and G-maze on three established syntactic ambiguity resolution phenomena. The results of Witzel et al. (2012) indicated that some syntactic ambiguity resolution phenomena were picked up as effectively by L-maze as by SPR, and that G-maze was perhaps even more sensitive, although they did not conduct a direct comparison of the sensitivity of the methods.

To foreshadow our findings, we find that G-maze and A-maze run well over the web, and are more sensitive than SPR. Given that A-maze performs well and is easy to prepare, we argue that web-based A-maze should be added to the psycholinguist's toolkit for sentence processing research. We also make our code for generating distractors and running Maze online freely available at `github.com/vboyce/Maze`.

3 Automating Maze

3.1 Motivation

As described in Section 2.3, the Maze task is a good candidate for more widespread adoption in sentence processing research, and for being suitable for use on crowd-sourcing platforms. Among maze variants, G-maze shows signs of being more powerful than L-maze, but construction of materials is much more laborious for G-maze than for L-maze. Thus, it would be valuable to a) automate the creation of distractors for G-maze, and b) develop software for running the Maze task online, on crowd-sourced populations.

The key requirement for automating G-maze materials construction is to automate the selection of good distractor words – most crucially, words that are a *poor* fit to a given context. This is not a trivial task because there are many ways (both semantically and syntactically) that a sentence can legitimately continue. Here we take advantage of the impressive advances in NLP language models that are trained precisely to perform this task, putting a conditional probability distribution over next words given a preceding sentence context. These conditional probabilities are often quantified in terms of bits of SURPRISAL, where surprisal is the negative log of probability. (Thus, higher surprisal corresponds to lower conditional probability, and something with 1 bit more surprisal is half as likely to occur.) State-of-the-art language modeling architectures today are often recurrent neural network (RNN) models (Elman, 1990), typically using Long Short Term Memory (LSTM)

cells (Hochreiter and Schmidhuber, 1997), which have achieved impressive performance in learning structure from the statistics of sequences and in representing long distance dependencies (Jozefowicz et al., 2016). LSTM RNNs have been shown to learn some hallmark grammatical dependencies; Gulordava et al. (2018) showed that with careful parameter setting a model trained only on next-word prediction got long-distance agreement relations right most of the time, in the absence of semantic cues. While these models don't have any formal notion of "grammaticality", they have been shown to assign higher surprisal to ungrammatical forms compared to grammatical forms (Marvin and Linzen, 2018; Wilcox et al., 2018; Futrell et al., 2019).

We use these models to select words that are have high surprisal given the context. While there is no guarantee that these words will be ungrammatical, to the extent that a model has learned a distribution word sequences that correlates with human intuitive judgments, it should be the case that words the model finds unlikely will also be highly incongruous (and often ungrammatical) to human readers.

We also impose other constraints on our distractor words. For the Maze task to be effective, the distractor words should not only be identifiable as the wrong choice, but also not introduce too much variance into reaction times. To this end we match distractor words with the correct words for length (in letters) and overall frequency, which are two effects known to affect word recognition and reading times. This also prevents heuristic-based strategies that do not involve relating a word to its preceding context, such as 'choose the short word' or 'choose the overall more familiar word', from being effective.

3.2 Auto-generation Process

We illustrate our automated Maze materials construction process in Figure 2. It involves two main stages: a set-up stage and a distractor-selection stage.

3.2.1 Set-up

In the set-up stage, we create look-up tables mapping from words to frequencies and from ⟨length, frequency⟩ pairs to lists of potential distractor words.[1]

We use the Google Books Ngrams Corpus (Michel et al., 2011) as our source for word frequencies.[2] By using a large corpus, we ensure that we have frequency data for almost any word that might show up in psycholinguistic materials (without a frequency to look up, our algorithm doesn't work, so researchers would need to take special measures for experiments involving target sentences with words for which frequency statistics are not available).

Distractors should be easily identifiable as words, so participants aren't surprised by misspellings, proper nouns, or words they don't know (all of which occur in the Google Books corpus). We also include a requirement that distractors can legitimately be recapitalized to match the capitalization of the correct word they are paired with. To this end, we restrict distractors to words in the UNIX dictionary file that were only made up of lowercase letters. Additionally, we manually exclude a few short 'words' such as the letter 'm', which we consider to be insufficiently word-like.

From these frequencies, we built two look-up tables; one from words to frequency bins, and one from ⟨length, frequency-bin⟩ pairs to lists of valid distractors.[3]

3.2.2 Distractor selection

When the automation process is run on materials; it iterates through each item number (corresponding to a sentence, or minimal set of matched sentences), and selects a distractor for each word position. Distractors are selected to be matched to the real word(s) for length and approximate frequency, and to be low probability

[1] These look-up tables are made available so one can generate Maze materials without going through the set-up procedure. However, we also make all of our code available so that the set-up process can be replicated or modified.

[2] For most words, we use the overall unigram frequency; however, contractions were usually, but not always, parsed as multiple words, leading to inappropriately low unigram frequencies. For contractions, we manually approximated their frequencies using Google Ngrams Viewer (which shows their accurate bigram frequency). A list of contractions and the frequencies we assign to them is included with our code.

[3] Frequencies were binned by taking the floor of log_2 of the number of occurrences in the Google Book Ngrams corpus. We only considered words that occurred at least 2^{13} times, and all words that occurred more than 2^{25} were binned together. To account for sparsity of very short or very long words, words of length 3 or less were treated as having length 3 and words of length 15 or greater were treated as having length 15 for list-creation and look-up purposes.

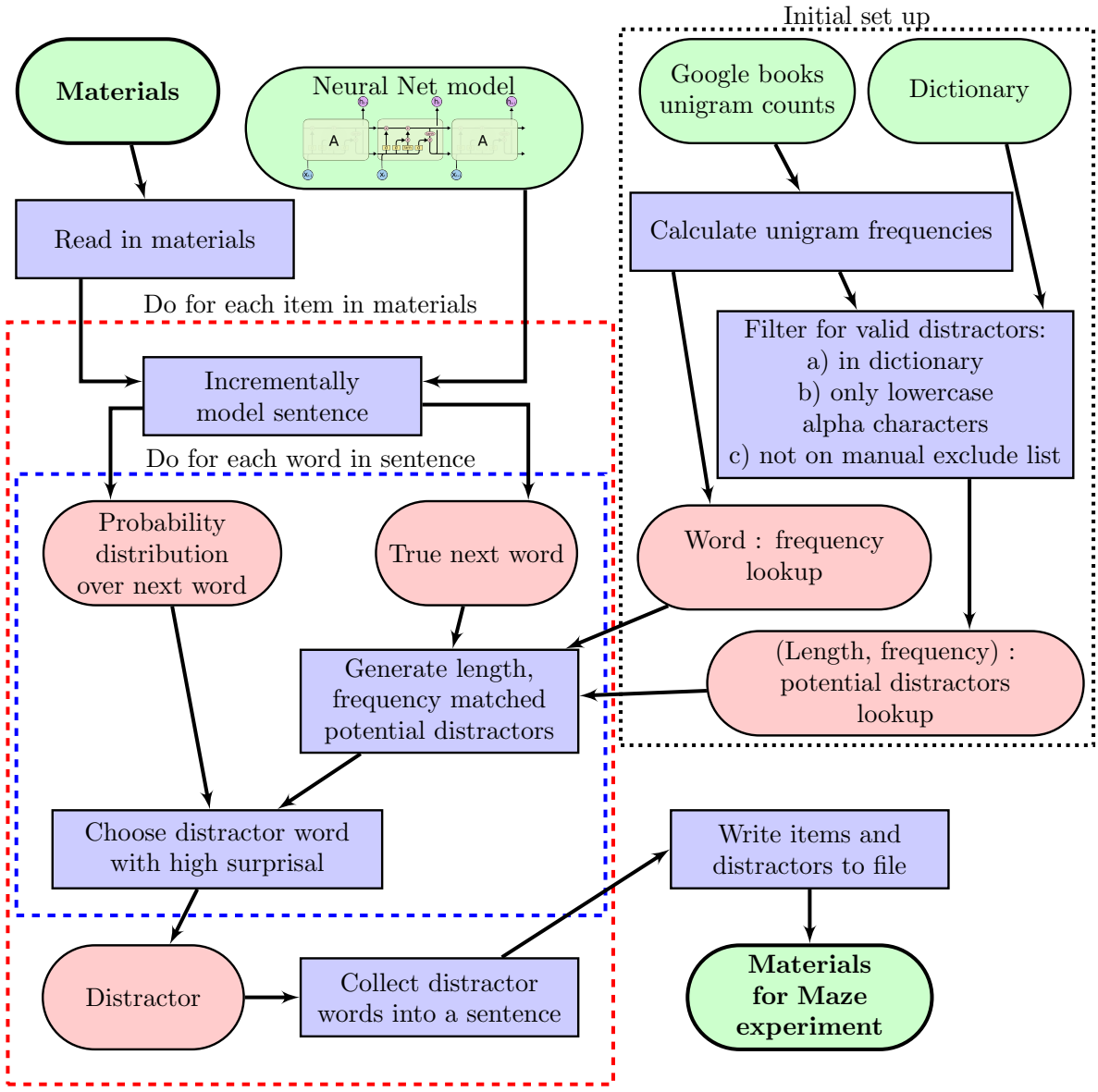


Figure 2: Schematic of how A-maze materials are generated. Image of LSTM from `colah.github.io/posts/2015-08-Understanding-LSTMs`

in context as judged by the language model. We set up the generation process to run with either of two pre-trained, freely available models, from Jozefowicz et al. (2016) and from Gulordava et al. (2018). Future implementations could use other existing state-of-the-art language models such as Transformer-XL (Dai et al., 2019) or Recurrent Neural Network Grammars (Dyer et al., 2016).

Rather than trying to globally optimize the choice of distractors according to some unified objective functioning, we adopt a procedure that runs quickly while still selecting sufficiently-low-probability distractors. To generate a distractor for a word w_i in a sentence, we run the language model on the sentence up through the immediately preceding word w_{i-1}, thus obtaining a probability distribution over possible next words w_i'. We then retrieve from our look-up tables the set of all the possible distractor words with the same length and frequency-bin as w_i and randomly order that set into a list. We then go through this list of potential distractors, checking their conditional probabilities, until we find one with a surprisal above a preset threshold (for the experiments presented here, we used a threshold of 21 bits of surprisal, corresponding to a conditional probability of roughly 4 in 10 million). Once a word with low enough probability is found, it is chosen as the distractor.[4] The model continues until an above-threshold word is found or 100 words have been checked.[5] If 100 words have been checked without any word meeting the threshold, the word among these with the lowest conditional probability is chosen as the distractor. The chosen distractor is then matched to the correct word on capitalization and end punctuation (period or comma). We then advance to the next word in the sentence and repeat this procedure to choose an appropriate distractor for that word.

In some cases it may be desirable (for both G- and L-maze) to use the same distractors across a set of minimally differing sentences (typically, this would be for the sentences instantiating different conditions of a given experimental item). For instance, Witzel et al. (2012) used critical items coming in two variants differing by one word, and gave the same word positions in each sentence the same distractors. We follow this pattern, generating one distractor word per word position per item number. Thus in the first pair of example sentences in Figure 3, 'herself' and 'himself' get the same distractor. When there are multiple sentences to match, we consider distractors matched to the average length and average log frequency of the correct words. When choosing a distractor, we take the first distractor that meets the theshold for all the contexts.[6] Chosen distractors are matched on capitalization and end punctuation individually to each correct word.[7]

3.3 Considerations when using A-maze

A-maze can be used on existing materials, such as those designed for self-paced reading. Some small adaptations may be needed; such as changing hyphenation of a compound word, or replacing a two word place name with a one word place name. However, we find that A-maze works even when some of the words in the materials are unknown to the models, because the model can make reasonable predictions based on the rest of the context.

Maze should be easy to run online; it merely involves showing stimuli on a screen and recording button press times, similar to SPR which is run over the web. One concern with web-based SPR is noisy data from inattentive participants, which researchers may attempt to weed out with comprehension questions and exclusion criteria. A-maze (and G-maze) are especially robust to participant pools where some participants don't pay enough attention all of the time. Participants who aren't paying attention (either in general, or during a period of the experiment) will have higher error rates. As soon as they make a mistake on a sentence, that sentence ends. Thus, participants who make mistakes before the region of interest on a sentence don't contribute RT data to the region of interest; and thus their potentially noisy data won't affect results. As we will see in Section 4.4, a substantial proportion of trials are filtered out in this way within the first few

[4] Potential distractors that the model treated as unknown (i.e. outside of the model's vocabulary) are not selected because we don't trust their conditional probabilities to be accurate.

[5] If the list of potential distractors runs out before one of these criteria is met, the list of distractors with the same length, but the next higher frequency-bin is used to supplement.

[6] If 100 words are checked without any word meeting the threshold, the word with the highest minimum surprisal across all sentences is chosen.

[7] One consequence of this: distractors might not be identical in capitalization or punctuation if target words forming a set across otherwise matched sentences differ in this respect. For instance, in the last pair of example sentences in Figure 3, the distractor paired with "coach," might be "chaos," (with comma), but the distractor paired with "coach" would be "chaos" (no comma).

words of the sentence when Maze is run on Mechanical Turk, and this is likely to be a major advantage of Maze over SPR. For this reason, however, we recommend that critical words in a Maze task be at least a few words into the sentence.

One concern with using automated distractors is that sometimes the algorithm might fail and generate a word that is acceptable in context. We have found that this does happen occasionally, particularly on word 2 of the sentence, when the model, with only one word of context, assigns low probability to many continuations, even some that can be felicitously integrated into the sentence. As the sentence continues, and context accumulates, the model's judgments about low probability words improve. This is not only a problem with the automated materials construction process of A-maze: even a highly trained and experienced human researcher constructing G-maze distractors can sometimes miss a potential parse and allow a grammatical distractor to slip through (see Table 2 for plausible distractors that emerged in our G-maze and A-maze experiments). Crucially, while distractor generation can take some computational time (depending on which model is used), it does not take much researcher time.

Both poor distractors and distracted participants may contribute to high error rates early in sentences, but we find that these error rates generally stabilize by word 5 (see Figure 5). As long as the critical regions of a sentence are more than five words into the sentence, these effects should not affect critical RTs.[8] They may reduce the number of data points available at the critical region, but this data loss can be estimated ahead of time, and with crowd-sourced experiments it is often easy to recruit a greater number of participants.

The code for creating A-maze distractors is freely available at `github.com/vboyce/Maze/tree/master/maze_automate`. As we update the method to produce better matched distractors for a wider set of experimental items, we will add improved versions of the process to this repository.

4 Validation Experiment

To compare the performance of these crowd-sourceable experimental methods and to evaluate the performance of our A-maze implementation, we conducted 5 experiments: SPR, L-maze, G-maze, Jozefowicz A-maze (using the language model of (Jozefowicz et al., 2016) for word conditional probability estimation), and Gulordava A-maze (using the language model of (Gulordava et al., 2018)). We use the materials of Witzel et al. (2012), which further allows us to compare our results with their in-lab results. We pre-registered this study in two parts: one for SPR, L-maze and G-maze, and another for the A-mazes. Pre-registrations are available at `aspredicted.org/blind.php?x=iq2rd9` and `aspredicted.org/blind.php?x=m9n5bc`. The SPR, L-maze and G-maze data was collected on 25 July 2018, and the A-maze data was collected on 9 May 2019. We make our materials, data, and analysis code available at `github.com/vboyce/Maze/tree/master/experiment`.

4.1 Methods

4.1.1 Materials

We requested and received materials from Witzel et al. (2012), and followed their design closely. These experimental materials examined three different attachment preferences. In each case, the context sets up a syntactic attachment ambiguity in which one attachment possibility has generally been found to be preferred in incremental processing by native English speakers; we expect that the critical disambiguating word in the sentence that will be harder to process when it disambiguates to the previously dispreferred attachment than when it disambiguates to the preferred attachment (see Figure 3 for sample stimuli).

The first ambiguity involves attachment of relative clauses into preceding complex noun phrases that involve a prepositional phrase postmodifier; in English it is typically the case that "low" attachment, to the most recent noun, is preferred (Cuetos and Mitchell, 1988). These are disambiguated by gendered reflexive pronouns within the relative clause which match the gender of only one of the nouns. The second ambiguity involves of attachment of temporal adverbs into nested preceding verb phrases; again, "low" attachment into the most recent verb phrase is generally preferred. These are disambiguated by the temporal adverb (which might be two words, i.e. 'next week'), which matches the tense of only one of the clauses. The

[8] If critical words need to be early in the sentence, one could potentially filter the data and only consider data from sentences that were (correctly) completed.

Relative Clause– Low attachment:
The son of the lady who politely introduced herself was popular at the party.
Relative Clause – High attachment:
The son of the lady who politely introduced himself was popular at the party.
Adverb clause – Low attachment:
James will fix the car he drove yesterday, but he will need some help.
Adverb Clause – High attachment:
James will fix the car he drove tomorrow, but he will need some help.
Sentence v Noun Phrase conjunction (S v NP) – With comma:
The swimmer disappointed her coach, and her mother tried to console her.
Sentence v Noun Phrase conjunction (S v NP) – No comma:
The swimmer disappointed her coach and her mother tried to console her.

Figure 3: Sample Stimuli with disambiguating words underlined

last ambiguity involves the ambiguity of an "and NP" sequence immediately following transitive clause as involving either Sentence or Noun Phrase (S v NP) coordination. When the preceding transitive clause is ended with a comma, Sentence coordination is typically preferred; when it is ended without a comma, Noun Phrase coordination is typically preferred (Frazier and Clifton, 1997). This is disambiguated by the second verb, which disambiguates to a sentence conjunction. Thus, based on previous studies and on the results of Witzel et al. (2012), we expect faster RTs at the critical disambiguating word in the low-attachment and comma condition, because participants would be less likely to favor a parse of the sentence inconsistent with this word.

For SPR, Witzel et al. (2012) used yes/no comprehension questions for half of the items. We wrote similar comprehension questions for the other half of the items, and gave a comprehension question after every item. For L-maze and G-maze, we used the same distractor words and the same positioning (was the correct word on the left or right?) as Witzel et al. (2012). For both A-maze tasks, we used the same correct materials, but generated our own distractors, using the process described in Section 3.2. We ran our procedure twice, once for each of the two models. We took the distractors as is, without any checking or quality control. For both A-mazes the right/left positioning of correct words and distractors was randomized, except that the first word of each sentence was always presented on the left, against a distractor of 'x-x-x'.

4.1.2 Participants

We recruited 50 participants in each of the five experiments. Participants were recruited from Amazon Mechanical Turk and paid $3.00 for each task. Participants clicked the link, which opened our study running on a webpage; at the end of the experiment they were given a code which they could enter on Mechanical Turk to receive payment. We used UniqueTurker ID (`uniqueturker.myleott.com`) to ensure that individuals did not particulate in multiple experiments.

4.1.3 Procedure

We used the Ibex web-based psycholinguistic experiment software platform (`github.com/addrummond/ibex`) for our experiments. For SPR, we used the SPR module already provided in Ibex. For the Maze tasks we implemented a new module based on the SPR module. In our Maze implementation, each target–distractor word pair appears on-screen simultaneously, one on the right and one on the left. The participants uses the 'e' and 'i' keys to select among the two words. If the participant correctly selects the target, the experiment advances to the next word pair. If they incorrectly select the distractor, an error message ("Incorrect! Press any key to continue.") is shown and with the next key press the experiment continues to the next sentence. After correctly completing a sentence, a participant sees the message "Correct! Press any key to continue." and with the next key press the experiment continues to the next sentence. As a slight gamification, we added a running counter of words correct at the top of the screen, which does not reset between sentences (but does reset between experimental blocks; see below). The Maze module records time in between presses (using the same button-press timing code as the SPR module), as well as whether the selection was correct.

When the experiment finishes, all results are transmitted to the server and recorded, for later researcher download.

At the start of the experiment, participants were told how their data would be used and asked to indicate their informed consent. They then saw instructions, followed by 8 practice items. They then saw 24 sentences of each type (12 in each of the two levels) mixed in with 24 filler items.[9] These items were arranged in 8 blocks of 12 items each, with a brief pause between blocks when participants were told how many blocks were left. This is the same design as used in Witzel et al. (2012). For SPR, each sentence was followed by a yes/no comprehension question (and feedback was given on the correctness of the response); for Maze, no comprehension questions were used.

At the end of the experiment, participants were asked for feedback on the experiment, asked for demographic information and debriefed about the goals of the experiment. They were then given a code which they could enter into Amazon Mechanical Turk to receive payment. The entire experiment took on average 15 minutes, plus a couple minutes for instructions and optional demographic questions.

4.2 Data Analysis

Although this study was described as being for native English speaking US citizens, anyone could complete the experiment and get paid. In the demographic section, we included three yes/no questions: were they US citizens, were they currently living in the US, and were they native English speakers. Only data from participants who answered yes to all three of these questions was included in the analysis. After this exclusion, we had 44 participants contributing data for L-maze, 44 for G-maze, 43 for SPR, 46 for Gulordava A-maze, and 42 for Jozefowicz A-maze.

For SPR, we additionally exclude data from participants who correctly answered less than 80% of comprehension questions, leaving us with data from 32 participants. For the Maze tasks, we only include RTs where the correct word was chosen. Because sentences terminate when a mistake is made, we don't have data for the rest of a sentence after a mistake. Accounting for this, we have RTs for 75% of words for L-maze, 64% for G-maze, 64% for Gulordava A-maze, and 55% for Jozefowicz A-maze. This leaves us with roughly comparable amounts of data across all Maze tasks and SPR. In many RT-based sentence processing studies, very long RTs are often identified as "outliers" and excluded or replaced to improve statistical power; we instead use log(RT) as our dependent measure, which reduces these concerns as RTs are right-skewed and very roughly log-normal distributed (Luce et al., 1986; Van Zandt, 2000; Baayen and Milin, 2010). We excluded 2 words from L-maze and 1 word from Jozefowicz A-maze for having recorded RTs of 0; indicative of a software error in RT recording.

For all tasks, the key measure is the difference in RT between the two conditions at the critical word (where disambiguation occurs) and following region (see Figure 3 for examples of disambiguating words). We measure the difference in RT at each word position (relative to critical/disambiguating word) from -5 to +5 (five words before, the critical word, five words after). We follow Witzel et al. (2012) in averaging the RTs for the two-word critical regions (e.g. 'next week') and analysing them as one word. We used the mixed effects model

```
log(RT) ~ condition + (condition | subject) + (condition | item)
```

for each word position, type of item, and task combination. We report estimated effect sizes and two-sided p-value equivalents. We do not correct for multiple comparisons, as our goal is to compare the strengths of effects found by different experimental methods, rather than make claims based on significance of any particular result.[10] Analysis was done in R, using BRMS (R Development Core Team, 2009; Bürkner, 2018).

Results for the 0-3 word region are shown in Figure 4, a table of all the estimated effect sizes and p-value equivalents is included in Table 1.

To allow for direct comparison between our results and the in-lab results of Witzel et al. (2012), we obtained the data from Witzel et al. (2012) and re-analysed it identically to how we analysed our own data; these re-analysed data are referred to as Lab SPR, Lab G-maze, and Lab L-maze.

[9] Due to a typo in the grouping label, only half of the Adverb clause sentences (12, 6 in each level) were shown to G-maze participants. This reduction in data would be expected to lead to weaker results, compared to if all items had been shown.

[10] By "p-value equivalent" we mean the following: if the largest symmetric interval on the posterior distribution for the fixed effect of 'condition' that does not include zero contains probability mass q, then we report $(1 - q)$ as a "p-value equivalent", following common practice in Markov-chain Monte Carlo fitting of mixed effects models (e.g. Baayen et al., 2008).

Word Position	Lab SPR	Lab L-maze	Lab G-maze	Web SPR	Web L-maze	Web G-maze	Web A-maze Gulordava	Web A-maze Jozefowicz
Relative Clause								
-5	-4 (0.65)	16 (0.17)	3 (0.89)	-10 (0.23)	-7 (0.67)	-3 (0.9)	10 (0.53)	-11 (0.56)
-4	4 (0.7)	-10 (0.58)	-4 (0.85)	2 (0.8)	22 (0.19)	7 (0.71)	-10 (0.63)	-8 (0.57)
-3	6 (0.64)	-1 (0.95)	37 (0.099)	-10 (0.26)	13 (0.49)	-8 (0.72)	-14 (0.39)	-20 (0.48)
-2	2 (0.87)	2 (0.88)	-31 (0.33)	-10 (0.29)	17 (0.33)	-1 (0.99)	16 (0.42)	-2 (0.92)
-1	6 (0.7)	-11 (0.63)	-33 (0.3)	-7 (0.48)	14 (0.46)	-44 (0.22)	35 (0.21)	-14 (0.68)
0	15 (0.39)	35 (0.051)	**121 (0)**	-10 (0.31)	18 (0.32)	**105 (0.0025)**	**73 (0.0095)**	**163 (0.001)**
1	23 (0.15)	26 (0.27)	39 (0.33)	17 (0.17)	**47 (0.047)**	58 (0.14)	82 (0.23)	5 (0.88)
2	24 (0.063)	21 (0.43)	14 (0.61)	10 (0.3)	23 (0.22)	2 (0.96)	-2 (0.9)	**68 (0.045)**
3	-2 (0.85)	27 (0.13)	-10 (0.71)	4 (0.68)	1 (0.95)	-15 (0.61)	14 (0.58)	26 (0.34)
4	-11 (0.29)	-8 (0.72)	-51 (0.059)	1 (0.89)	-4 (0.8)	-51 (0.19)	17 (0.45)	17 (0.49)
5	-3 (0.77)	19 (0.32)	-9 (0.76)	14 (0.23)	56 (0.052)	-22 (0.62)	-16 (0.49)	20 (0.56)
Adverb Clause								
-5	0 (0.98)	-3 (0.84)	-12 (0.57)	5 (0.53)	-1 (0.99)	-31 (0.5)	-9 (0.66)	2 (0.94)
-4	15 (0.19)	9 (0.51)	-12 (0.51)	4 (0.57)	7 (0.74)	-22 (0.46)	-15 (0.4)	11 (0.61)
-3	-1 (0.95)	18 (0.28)	12 (0.68)	0 (0.97)	8 (0.59)	-64 (0.15)	9 (0.7)	14 (0.61)
-2	**18 (0.048)**	-8 (0.7)	33 (0.27)	-14 (0.092)	-24 (0.24)	0 (0.98)	32 (0.2)	16 (0.59)
-1	8 (0.45)	-10 (0.57)	-17 (0.42)	-8 (0.34)	-15 (0.36)	-3 (0.94)	18 (0.41)	-26 (0.33)
0	**56 (0.017)**	**48 (0.0025)**	**216 (0)**	9 (0.33)	**44 (0.014)**	**213 (0.005)**	**175 (0)**	**170 (0.001)**
1	**25 (0.032)**	13 (0.4)	**78 (0.0065)**	**27 (0.002)**	-11 (0.5)	13 (0.72)	**77 (0.001)**	32 (0.22)
2	15 (0.083)	9 (0.62)	**93 (0.003)**	**27 (0.0045)**	7 (0.62)	-5 (0.89)	6 (0.76)	30 (0.15)
3	9 (0.48)	-8 (0.72)	-30 (0.22)	14 (0.12)	-7 (0.73)	39 (0.41)	27 (0.23)	1 (0.95)
4	13 (0.093)	3 (0.88)	23 (0.35)	**15 (0.02)**	-41 (0.054)	41 (0.18)	0 (1)	12 (0.66)
5	8 (0.32)	16 (0.29)	20 (0.4)	10 (0.19)	-2 (0.91)	30 (0.37)	-15 (0.42)	-41 (0.11)
S v NP								
-5	5 (0.59)	9 (0.5)	-6 (0.8)	6 (0.39)	12 (0.49)	-31 (0.29)	-1 (0.96)	-10 (0.73)
-4	**-69 (0.0045)**	5 (0.76)	2 (0.93)	-5 (0.57)	-28 (0.17)	2 (0.94)	-25 (0.46)	-55 (0.11)
-3	-9 (0.37)	**-32 (0.024)**	-17 (0.42)	-14 (0.064)	-25 (0.078)	-14 (0.6)	-6 (0.75)	4 (0.87)
-2	11 (0.15)	-28 (0.078)	-46 (0.054)	2 (0.82)	-23 (0.18)	-24 (0.37)	-12 (0.42)	5 (0.86)
-1	7 (0.48)	-13 (0.47)	-6 (0.82)	2 (0.75)	-33 (0.071)	-38 (0.09)	-10 (0.54)	-32 (0.3)
0	17 (0.17)	-5 (0.81)	-7 (0.86)	2 (0.82)	-1 (0.98)	19 (0.65)	**96 (0.013)**	**134 (0.024)**
1	12 (0.28)	-6 (0.82)	15 (0.58)	0 (0.98)	0 (0.99)	13 (0.6)	-32 (0.11)	-2 (0.92)
2	3 (0.73)	-6 (0.71)	38 (0.099)	5 (0.55)	7 (0.7)	-42 (0.12)	-8 (0.7)	1 (0.98)
3	6 (0.57)	-3 (0.88)	9 (0.74)	2 (0.81)	-6 (0.69)	-2 (0.92)	1 (0.95)	45 (0.14)
4	-3 (0.7)	-7 (0.79)	-27 (0.37)	-1 (0.87)	-3 (0.88)	-25 (0.42)	4 (0.83)	-5 (0.92)
5	-13 (0.42)	30 (0.34)	-13 (0.7)	6 (0.71)	-12 (0.78)	-3 (0.95)	29 (0.28)	-28 (0.56)

Table 1: Mean difference in RT between the dispreferred conditions (high attachment or no comma) and the preferred conditions. P-value equivalents are in parentheses. Bolding indicates $p < .05$.

4.3 Results

Figure 4 summarizes the estimated effect size for each type of attachment ambiguity and each experimental method, at the critical disambiguating words and each of the next three words. As is immediately evident, G-maze and both A-mazes generally yield large, immediate effects strongly localized to the disambiguating word (with smaller effects sometimes also spilling over one to two words further downstream), compared to SPR and L-maze, which do not.

Looking first at relative clause attachment disambiguation, we see significant effects of 105 ms for Web G-maze ($p = .0025$), 73 ms for Gulordava A-maze ($p = .0095$), and 163 ms for Jozefowicz A-maze ($p = .001$). These are all qualitatively similar to the 121 ms effect on Lab G-maze. We see a numerical trend towards the effect to also appear on one to two words downstream, but this reaches significance only for word +2 and only for Jozefowicz A-maze. Neither Web L-maze nor Web SPR find an effect on the critical word, although L-maze does have an effect of 46 ms on the immediately following word ($p = .047$).

For adverb attachment disambiguation, we see larger effects all around, consistent with the findings of Witzel et al. (2012). Web G-maze, Gulordava A-maze, and Jozefowicz A-maze again have large effects of 213 ms, 175 ms, and 170 ms respectively (p's ≤ 0.005) on the critical word. Gulordava A-maze also has a 77 ms spillover on the next word ($p = .001$). For comparison, Lab G-maze has a 216 ms effect on the critical word, followed by 78 and 93 ms spillover effects on the next two words, respectively. Web L-maze finds a localized effect of 44 ms (p = .014) on the critical word; this is similar to the 48 ms effect from Lab L-maze. Web SPR shows no significant effect on the critical word, but finds spillover effects of 27 ms on the next two words (p $\leq$ .005); Lab SPR had effects of 56 ms on the critical word, and 25 ms on the next word (p $<$ .05).

For disambiguation of S v NP coordination ambiguity, both A-mazes find effects on the critical word; Gulordava A-maze finds a 96 ms effect ($p = .013$) and Jozefowicz finds a 134 ms effect ($p = 0.024$). This effect does not show up with the other tasks, but did show up in the eye tracking data from Witzel et al. (2012) (not shown here).

Overall, these results indicate that G-maze, Gulordava A-maze, and Jozefowicz A-maze are roughly equivalently good methods that can find strong, localized effects where SPR and L-maze cannot. They find comparable effect sizes to Lab G-maze. We take this as evidence that web-based A-maze may be superior to web-based SPR for at least some crowd-sourced psycholinguistics experiments: A-maze detected effects for all three phenomena we tested whereas SPR detected only one, and for A-maze the effect was always largest immediately at the critical disambiguating region, whereas SPR detected its one effect only in spillover.

4.4 Error rate

To better understand how and when data are lost to participant mistakes, we conducted a post hoc analysis of error rates by word position for G-maze and A-maze. Word positions earlier in sentences tend to have higher error rates than later words (Figure 5). This is likely due in part to mixed participant diligence. Participants with higher error rates will contribute disproportionately to the error rates at early words. Once a participant makes a mistake on a sentence, they no longer contribute to error rates for later words. It could also be due to worse (i.e. more plausible) distractors early in sentences.

We also directly checked how participant attentiveness differed between the web-based experiments and the in-lab experiments. We operationalized participant attentiveness by the number of sentences they completed (i.e. made no mistakes on), and compared the in-lab G-maze results and the web-based G-maze and A-maze results. As we see in Figure 6a, the web-based experiments had some participants who were not very attentive, while the in-lab experiment did not. However, if we weight not by participant, but by completed sentence (a proxy for reaching late-in-sentence critical words), the distributions of quality are more similar: most of the completed sentences are coming from fairly attentive participants (Figure 6b). Thus, this task seems to give us a way of selectively getting data from attentive participants, given a mixed participant pool, without having to create exclusion criteria.

We next looked at what sentence/correct word/distractor combinations had high error rates, both on G-maze and on A-mazes. These sources of high error rates drive data loss (when participants choose incorrectly, they don't see the rest of the sentence), and are likely to indicate places where it was ambiguous which word was the correct choice.

We found a few instances in the hand-constructed G-maze materials where the distractor word was grammatical and plausible (see Table 2). While these grammatical distractors were rare, they illustrate the

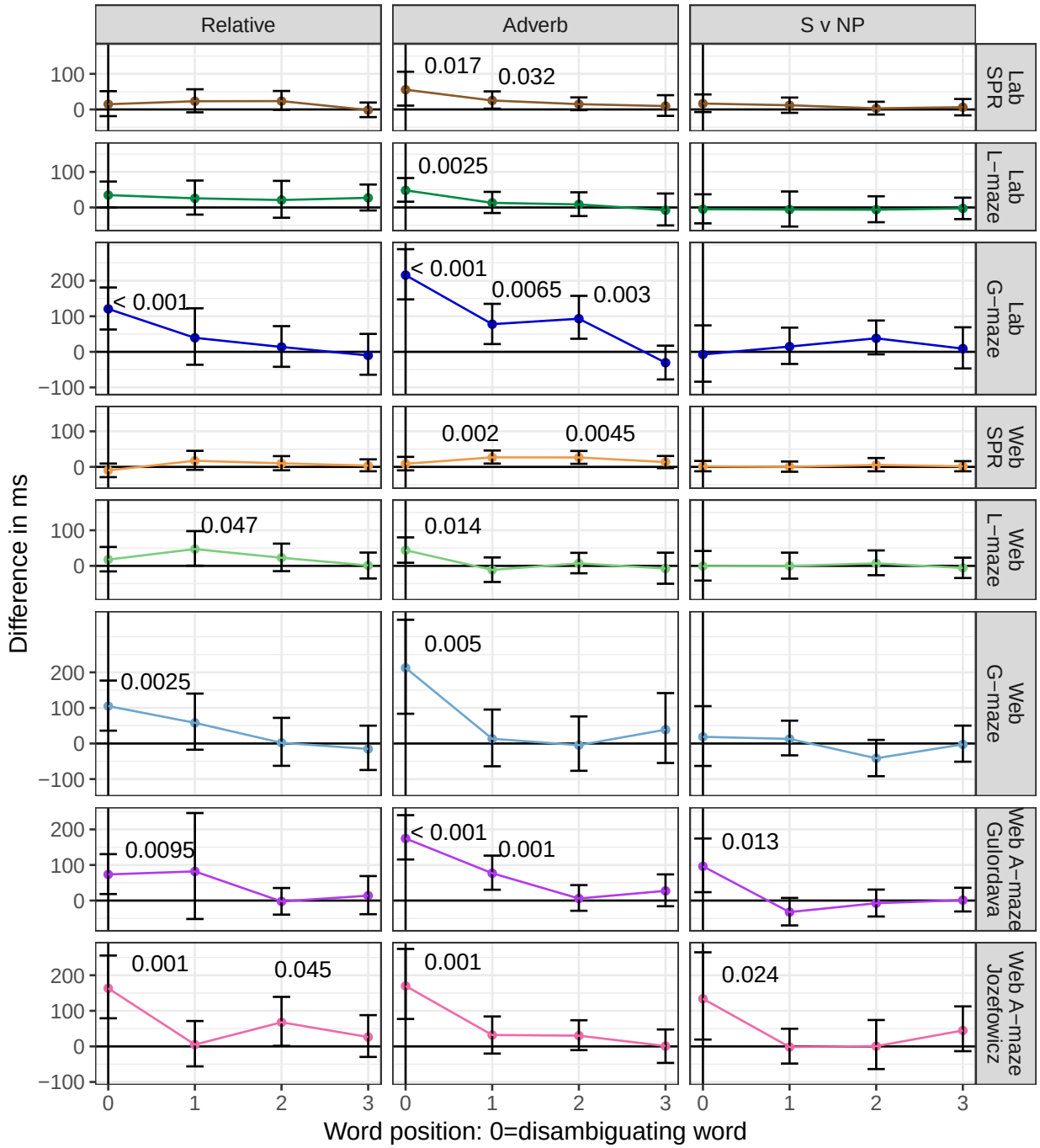


Figure 4: Mean difference in RT between the dispreferred conditions (high attachment or no comma) and the preferred conditions. Error bars indicate 95% confidence interval. P-value equivalents are shown when $p < .05$.

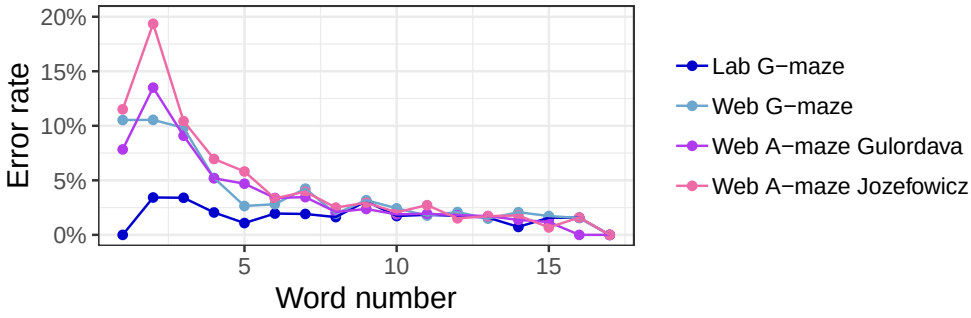
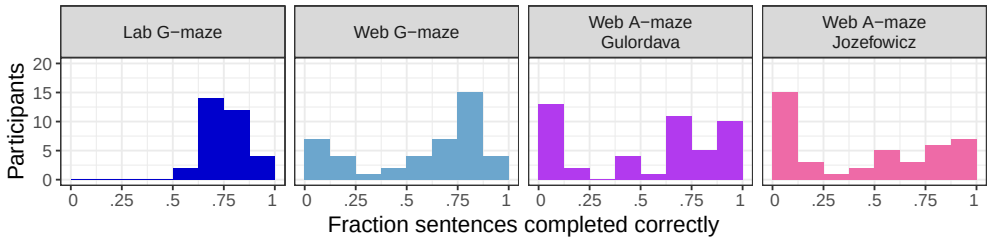
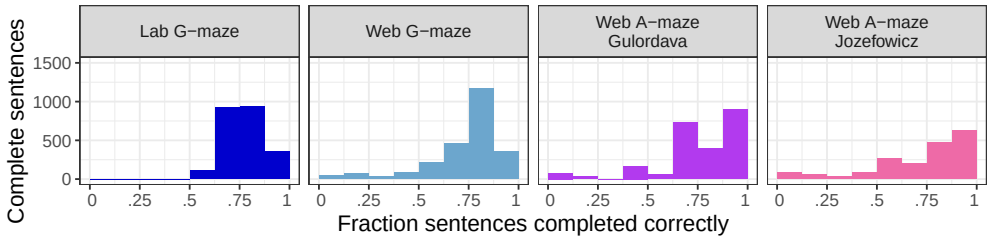


Figure 5: Error rate by word position. Word 1 is the first word of the sentence (always with an x-x-x distractor). In Lab G-maze, participants pushed a button to continue at word 1, but could not make an error.



(a) Distribution of participants by proportion of sentences they completed correctly.



(b) Distribution of completed sentences by proportion of sentences the participant completed correctly.

Figure 6: Distributions of participant and completed sentence quality. While some crowd-sourced participants do not complete many sentences; most of the completed sentences come from diligent participants.

Prefix	Correct	Distractor	Error Rate
G-maze			
Sarah and her mother had	steak,	mental,	35% (web), 57% (lab)
Margo will open bakeries in Chicago and	New	carve	34% (web), 46% (lab)
Jane	prepared	first	28% (web), 21% (lab)
A-maze Gulordava			
The	niece	cooks	44%
The swimmer	disappointed	propositions	30%
The	semester	steroids	29%
The daughter of the actor who hated herself/himself for failing	always	taught	28%
A-maze Jozefowicz			
Mark will answer the	email	exams	48%
The	husband	authors	46%
Jim	listened	survived	43%
The	uncle	roads	42%
The	knight	saints	40%

Table 2: Examples of plausible distractors and associated higher error rates

difficulty of constructing distractors that don't fit under any parse. Some other moderately high error-rates in G-maze seemed to come from distractors that, while ungrammatical, were very similar to plausible words, such as untensed forms of verbs that were good semantic fits in the context.

Both our A-mazes also occasionally generated grammatical, plausible distractors, especially for the second word of the sentence (see Table 2 for examples). This is perhaps unsurprising given how little context there is at the second word of the sentence, and given our A-maze constraint that the distractor is length-matched to the target (truly ungrammatical continuations at the second word of the sentence will often require a word in a closed-class part of speech, such as *The* ***of***, and these words are few and typically short). This leads to substantial data loss at word 2 (Figure 5). Future deployment of the Maze task might address this by using x-x-x distractors for more than just the first word of a sentence, introducing real-world distractors only after there's enough context for sharper constraint on sentence continuations. One might also construct a hybrid between Maze and centered SPR, where the first few words are presented by themselves (no distractor) as SPR and the rest of the sentence (including any critical regions) is done with Maze. It may also be possible to adjust the A-maze algorithm's thresholds for accepting distractors to reduce the chance of getting grammatical distractors, even with very little sentence context.

We also examined error rates at and around the critical disambiguating words. One puzzle in our results is that the two A-maze tasks find a reasonably large effect on the critical word in the S v NP condition, while G-maze, either in lab or over the web, doesn't not. This seems to suggest that for G-maze, the correct word in each condition was about as easy to select and integrate into the sentence; which could potentially be due to distractors that were better fits (more distracting) in the comma condition. If this were the case, we would expect to see inflated error rates. In Figure 7, we can see a spike in error rates in both S v NP conditions (especially for G-maze), but not in other conditions. This suggests that distractors at the S v NP critical word may have been hard to rule out.

More generally, it will likely be desirable for researchers to spot-check A-maze distractors at key parts of the sentence, including early on in the sentence and at and immediately after the critical word, carefully replacing any automatically generated distractors that are unsatisfactory. Additionally, an important post-hoc check on the quality of experimental materials is that error rates are low at critical words in the sentence. Researchers must also keep in mind that hard-to-reject distractors will generally yield longer RTs, and ensure that differences in RTs are not due to differences in the quality of distractors across conditions.

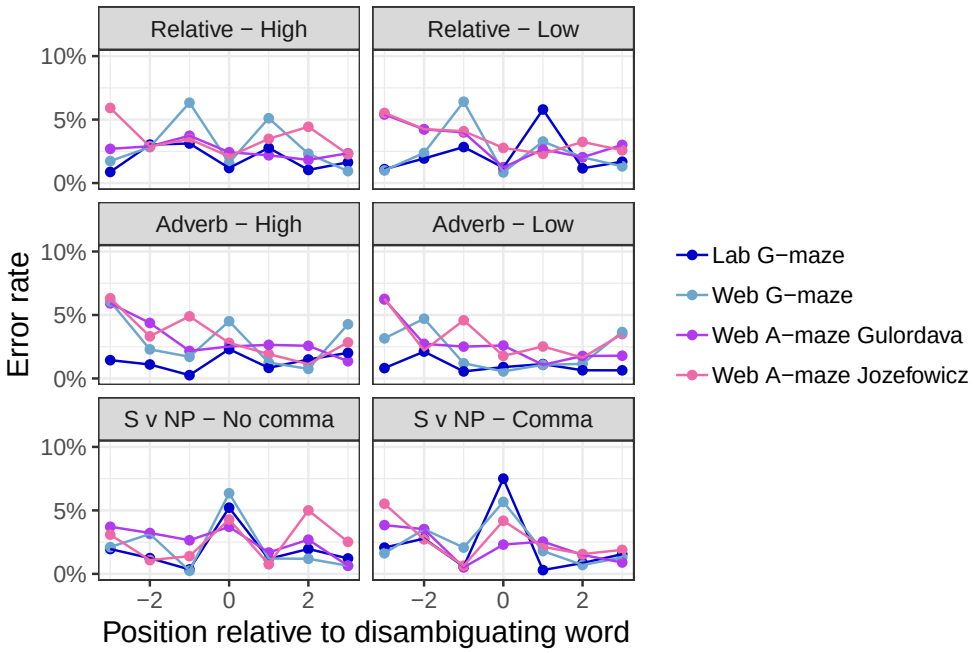


Figure 7: Error rates at the critical/disambiguating word by condition. Error rates are generally stable in this region, but the S v NP conditions have higher error at the critical word.

4.5 Power analysis

We can start to quantify the sensitivity of each experimental method studied here by performing power analysis for prospective future replications of these studies based on the data reported here. We assume reuse of the same materials (including the same distractors for the Maze study) and use Monte Carlo simulation to estimate the probability of obtaining an effect significant at the p-value equivalent 0.05 level as a function of the number of participants recruited. Simulated participant counts ranged from 10–60, and we ran 500 Monte Carlo replicates for each manipulation/method combination. For each replicate, we simulated new participants, but kept the items to same. To model data lost to errors (including earlier in the sentence), we assumed that participant data loss rates were normally distributed with the experimentally determined mean and variance, and sampled data loss rates for each participant. We randomly eliminated lines of data with probability equal to the simulated participants data loss rate. Using the same brms model as described in Section 4.2), each replication, we sampled a set of parameter values from the posterior and simulated data using these parameters. Then, we modelled the simulated data using the same model (run in lme4 (Bates et al., 2015) for speed), We report the proportion of replicates for which the effect size reaches statistical significance (operationalized as $t > 2$) as the statistical power level. Because SPR is usually analysed with a spillover region, here we calculate its power on the summed 0-3 word region; power in the Maze task are simulated just on word 0.

Figure 8 shows the results of our power simulations. Consistent with the results seen in Figure 4, we find that A-maze and G-maze are the most powerful methods, requiring fewer participants to have a high probability of finding a significant effect for these well-established syntactic attachment disambiguation phenomena. While different methods are better on different tasks, we find that A-maze tends to be higher powered than SPR, even when SPR is summed over a spillover region.

5 Discussion

This paper reports two methodological innovations for sentence-processing experiments and a test of these innovations. First, we created a web-based implementation of the Maze paradigm that can be used for crowd-sourced populations. Second, we developed and implemented a procedure for A(utomatically) generating

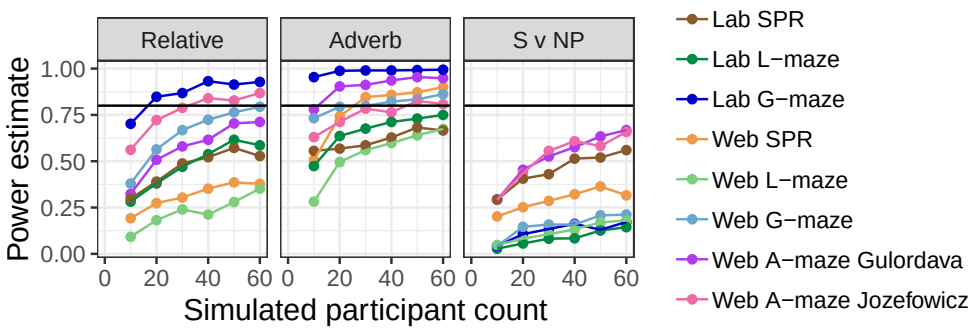


Figure 8: Estimated power for different numbers of participants. Power for SPR was calculated on the summed 0-3 word region, to account for spillover effects.

distractor words for G(rammaticality)-mazes. We find that the Maze task, but not self-paced reading, works as well on Mechanical Turk as in-lab. We further find (consistent with the results of Witzel et al. (2012)) that crowd-sourced G-maze is more powerful than crowd-sourced L(exicality)-maze for the syntactic attachment disambiguations studied here. Finally, we find that both our A-mazes, where distractors are generated entirely automatically, is just as powerful as G-maze with hand-constructed materials, and for one of the three phenomena studied was even more powerful.

Although these are encouraging initial results, it of course remains to be seen how well crowd-sourced Maze generalizes to other phenomena of psycholinguistic interest. As we can see from comparing our results and those of Witzel et al. (2012), different methods are more or less sensitive to different constructions. Web-based A-maze is a promising tool, and future work using it on a wider variety of materials could both pinpoint results, and also teach us about what effects it can or cannot detect. Our work also identified opportunities for further methodological refinement, including different surprisal thresholds and criteria for distractor matching (both to target words and across sentences). In particular we identified a problem with A-maze distractor generation word 2 which may be addressable by revised distractor criteria for that position (see Section 4.4).

While the implementation of A-maze we present is specific to English (and specific to the two language models we use), the same principles could be used to automate distractor generation for any language with large enough corpora for training good language models. This could potentially get around difficulties creating G-maze for languages with more flexible word order than English, where hand-constructing materials may be even more difficult. In addition, this set-up could be adapted to use future models, as better and better language models emerge from NLP research.

While our main interest is in developing a method for easier incremental processing data, our results also tell us something about the capacities of these NLP language models. For one, their predictions of high surprisal seems to align reasonably well with human plausibility judgments. However, their predictions seems much better a couple words into a sentence than at the beginning.

We encourage researchers to consider using A-maze as an alternative to SPR for crowd-sourced experiments (and potentially even for in-lab experiments). With automated distractor generation, A-maze is no more work than SPR to set up. Researchers familiar with the widely used Ibex software for SPR experiments should find it particularly easy to transition to web-based Maze tasks; the results are in nearly identical format.

In sum, our work helps unlock the potential of the Maze task for psycholinguistics research by removing three hurdles to its adoption: (1) we show that it can be run reliably in a crowdsourced format; (2) we provide a procedure for automatically generating distractors; and (3) we show that our automatic distractor-generation procedure leads to successful and powerful experimental tests for established sentence-processing phenomena. We make our A-maze generation code, as well as the Ibex code for the web-based Maze task, freely available online at `github.com/vboyce/Maze`.

Author Contributions

All authors conceptualized the experiment and methodology. VEB and RF designed the software, VEB did data collection, and VEB and RPL did data analysis. RPL provided funding and supervision. VEB drafted the manuscript, and all authors revised it.

Acknowledgments

This work benefited from presentation and feedback at the 2019 CUNY Sentence Processing Conference and from members of the MIT Computational Psycholinguistics Laboratory. We are grateful to Jeffrey Witzel for sharing experimental materials and data from Witzel et al. (2012) and for answering our questions. We gratefully acknowledge support to RPL from the National Science Foundation (BCS-1456081 and BCS-1551866), the MIT-IBM Watson AI Laboratory, and the MIT-SenseTime Artificial Intelligence Alliance.

References

Baayen, R. H., Davidson, D. J., and Bates, D. M. (2008). Mixed-effects modeling with crossed random effects for subjects and items. *Journal of Memory and Language*, 59(4):390–412.

Baayen, R. H. and Milin, P. (2010). Analyzing reaction times. *International Journal of Psychological Research*, 3(2):12–28.

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

Bates, D., Mächler, M., Bolker, B., and Walker, S. (2015). Fitting linear mixed-effects models using lme4. *Journal of Statistical Software*, 67(1):1–48.

Bürkner, P.-C. (2018). Advanced Bayesian Multilevel Modeling with the R package brms. *The R Journal*, 10(1):395–411.

Casler, K., Bickel, L., and Hackett, E. (2013). Separate but equal? a comparison of participants and data gathered via amazon's mturk, social media, and face-to-face behavioral testing. *Computers in Human Behavior*, 29(6):2156–2160.

Cuetos, F. and Mitchell, D. C. (1988). Cross-linguistic differences in parsing: Restrictions on the use of the Late Closure strategy in Spanish. *Cognition*, 30(1):73–105.

Dai, Z., Yang, Z., Yang, Y., Cohen, W. W., Carbonell, J., Le, Q. V., and Salakhutdinov, R. (2019). Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860*.

Difallah, D. E., Catasta, M., Demartini, G., Ipeirotis, P. G., and Cudré-Mauroux, P. (2015). The dynamics of micro-task crowdsourcing: The case of Amazon MTurk. In *Proceedings of the 24th International Ionference on the World Wide Web*, pages 238–247. International World Wide Web Conferences Steering Committee.

Dyer, C., Kuncoro, A., Ballesteros, M., and Smith, N. A. (2016). Recurrent Neural Network Grammars. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 199–209.

Elman, J. (1990). Finding structure in time. *Cognitive Science*, 14:179–211.

Enkin, E. (2012). The maze task: Training methods for second language learning.

Enochson, K. and Culbertson, J. (2015). Collecting Psycholinguistic Response Time Data Using Amazon Mechanical Turk. *PLoS ONE*, 10(3).

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

Frazier, L. and Clifton, C. (1997). Construal: Overview, motivation, and some new evidence. *Journal of Psycholinguistic Research*, 26(3):277–295.

Frazier, L. and Rayner, K. (1982). Making and correcting errors during sentence comprehension: Eye movements in the analysis of structurally ambiguous sentences. *Cognitive Psychology*, 14:178–210.

Freedman, S. E. and Forster, K. I. (1985). The psychological status of overgenerated sentences. *Cognition*, 19(2):101–131.

Futrell, R., Wilcox, E., Morita, T., Qian, P., Ballesteros, M., and Levy, R. (2019). Neural language models as psycholinguistic subjects: Representations of syntactic state. In *Proceedings of the 18th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Grodner, D. and Gibson, E. (2005). Some consequences of the serial nature of linguistic input. *Cognitive Science*, 29(2):261–290.

Gulordava, K., Bojanowski, P., Grave, E., Linzen, T., and Baroni, M. (2018). Colorless Green Recurrent Networks Dream Hierarchically. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1195–1205.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

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

Kizach, J., Nyvad, A. M., and Christensen, K. R. (2013). Structure before meaning: Sentence processing, plausibility, and subcategorization. *Plos One*, 8(10):e76326.

Koornneef, A. W. and van Berkum, J. J. (2006). On the use of verb-based implicit causality in sentence comprehension : Evidence from self-paced reading and eye tracking. *Journal of Memory and Language*, 54(4):445–465.

Li, R., Zhang, Z., and Ni, C. (2017). The impact of world knowledge on the processing of mandarin possessive reflexive zijide. *Journal of psycholinguistic research*, 46(3):597–615.

Luce, R. D. et al. (1986). *Response times: Their role in inferring elementary mental organization*. Number 8. Oxford University Press on Demand.

MacDonald, M. C. (1993). The interaction of lexical and syntactic ambiguity. *Journal of Memory and Language*, 32:692–715.

Marvin, R. and Linzen, T. (2018). Targeted Syntactic Evaluation of Language Models. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1192–1202.

Michel, J.-B., Shen, Y. K., Aiden, A. P., Veres, A., Gray, M. K., Pickett, J. P., Hoiberg, D., Clancy, D., Norvig, P., Orwant, J., et al. (2011). Quantitative analysis of culture using millions of digitized books. *Science*, 331(6014):176–182.

Mitchell, D. C. (1984). An evaluation of subject-paced reading tasks and other methods for investigating immediate processes in reading. In Kieras, D. and Just, M. A., editors, *New methods in reading comprehension*. Hillsdale, NJ: Earlbaum.

Nyvad, A. M., Kizach, J., and Christensen, K. R. (2015). (non-) arguments in long-distance extractions. *Journal of psycholinguistic research*, 44(5):519–531.

O'Bryan, E., Folli, R., Harley, H., and Bever, T. G. (2013). Evidence for the use of verb telicity in sentence comprehension.

Oliveira, C. S. F. d., Souza, R. A. d., and Oliveira, F. L. P. d. (2017). Bilingualism effects on l1 representation and processing of argument structure.

Paolacci, G. and Chandler, J. (2014). Inside the turk: Understanding mechanical turk as a participant pool. *Current Directions in Psychological Science*, 23(3):184–188.

Peer, E., Brandimarte, L., Samat, S., and Acquisti, A. (2017). Beyond the turk: Alternative platforms for crowdsourcing behavioral research. *Journal of Experimental Social Psychology*, 70:153–163.

Qiao, X., Shen, L., and Forster, K. (2012). Relative clause processing in mandarin: Evidence from the maze task. *Language and Cognitive Processes*, 27(4):611–630.

R Development Core Team (2009). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria.

Rayner, K. (1998). Eye movements in reading and information processing: 20 years of research. *Psychological Bulletin*, 124(3):372–422.

Rayner, K., Ashby, J., Pollatsek, A., and Reichle, E. D. (2004). The effects of frequency and predictability on eye fixations in reading: Implications for the E-Z Reader model. *Journal of Experimental Psychology: Human Perception & Performance*, 30(4):720–732.

Sikos, L., Greenberg, C., Drenhaus, H., and Crocker, M. W. (2017). Information density of encodings: The role of syntactic variation in comprehension. In *CogSci*.

Smith, N. J. and Levy, R. (2013). The effect of word predictability on reading time is logarithmic. *Cognition*, 128(3):302–319.

Staub, A. (2010). Eye movements and processing difficulty in object relative clauses. *Cognition*, 116:71–86.

Staub, A. (2011). Word recognition and syntactic attachment in reading: Evidence for a staged architecture. *Journal of Experimental Psychology: General*, 140(3):407.

Suzuki, Y. and Sunada, M. (2018). Automatization in second language sentence processing: Relationship between elicited imitation and maze tasks. *Bilingualism: Language and Cognition*, 21(1):32–46.

Van Zandt, T. (2000). How to fit a response time distribution. 7(3):424–465.

von der Malsburg, T. and Angele, B. (2017). False positives and other statistical errors in standard analyses of eye movements in reading. *Journal of Memory and Language*, 94:119–133.

Wang, X. (2015). Language control in bilingual language comprehension: evidence from the maze task. *Frontiers in psychology*, 6:1179.

Wilcox, E., Levy, R., Morita, T., and Futrell, R. (2018). What do RNN Language Models Learn about Filler-Gap Dependencies? In *Proceedings of the 2018 EMNLP Workshop Blackbox NLP: Analysing and Interpreting Neural Networks for NLP*, pages 211–221.

Witzel, J. and Forster, K. (2014). Lexical co-occurrence and ambiguity resolution. *Language, Cognition and Neuroscience*, 29(2):158–185.

Witzel, J. and Witzel, N. (2016). Incremental sentence processing in japanese: A maze investigation into scrambled and control sentences. *Journal of psycholinguistic research*, 45(3):475–505.

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